Flexible Margin-Based Classification Techniques

Seo Young Park

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Statistics and Operations Research.

Chapel Hill
2010

Approved by

Advisor: Dr. Yufeng Liu

Reader: Dr. Douglas G. Kelly

Reader: Dr. J. S. Marron

Reader: Dr. Wei Sun

Reader: Dr. Hao Helen Zhang

ABSTRACT

SEO YOUNG PARK: Flexible Margin-Based Classification Techniques
(Under the direction of Dr. Yufeng Liu)

Classification is a very useful statistical tool for information extraction. Among numerous classification methods, margin-based classification techniques have attracted a lot of attention. It can be typically expressed as a general minimization problem in the form of $loss + penalty$, where the loss function controls goodness of fit of the training data and the penalty term enforces smoothness of the model. Since the loss function decides how functional margins affect the resulting margin-based classifier, one can modify the existing loss functions to obtain classifiers with desirable properties.

In this research, we design several new margin-based classifiers, via modifying loss functions of two well-known classifiers, Penalized Logistic Regression (PLR) and the Support Vector Machine (SVM). In particular, we propose three new binary classification techniques, Robust Penalized Logistic Regression (RPLR), Bounded Constraint Machine (BCM), and the Balancing Support Vector Machine (BSVM). For multicategory case, we propose the multicagegory Composite Least Squares (CLS) classifier, a new multicategory classifier based on the squared loss function. We study properties of the new methods and provide efficient computational algorithms. Simulated and microarray gene expression data analysis examples are used to demonstrate competitive performance of the proposed methods.

ACKNOWLEDGEMENTS

I owe my deepest gratitude to my advisor, Professor Yufeng Liu, whose guidance, encouragement, and support enabled me to enjoy my research and complete my dissertation work successfully. He helped me explore the subject with his keen insight and immense knowledge, and provided many opportunities for various kinds of collaborative research. Beyond and above his obligations as a thesis advisor, he has been an integral if not essential advocate to any and all of my potential future pursuits. I could not imagine having a better advisor for my Ph.D. study.

I wish to express my sincere appreciation to committee members, Douglas G. Kelly, J. Steve Marron, Wei Sun, Hao Helen Zhang for their valuable comments and suggestions on this dissertation.

Finally, I would like to thank my husband Sungkyu Jung for everything he has done for me as a family, a friend and a fellow statistician. This thesis would not have been possible without his love, patience, and understanding.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background on Classification

Classification, as an example of supervised learning, is a procedure that builds a model based on a training dataset to predict the class memberships for new examples with only covariates available. It can be understood as a special form of regression with the response variable being categorical. If the response variable is binary, that is, there are only two classes, it is known as binary classification. If there are more than two classes, we have multicategory classification.

For simplicity, we first focus on binary classification, and multicategory classification will be discussed in Chapter 4. In binary classification, we want to build a classifier based on a training sample $\{(\boldsymbol{x}_i, y_i)\}_{i=1,\cdots,n}$, where $\boldsymbol{x}_i \in \mathbf{R}^d$ is a d-dimensional vector of predictors, and $y_i \in \{+1, -1\}$ is its class label. Typically it is assumed that the training data are distributed according to an unknown probability distribution $P(\boldsymbol{x}, y)$. Binary classification is to find a decision rule $\phi(\cdot)$ and predict the class membership as $\hat{y} = \phi(\boldsymbol{x})$ for any future observation $\boldsymbol{x}$. One important goal is to minimize the misclassification rate $P(Y \neq \phi(\boldsymbol{X}))$.

Our focus in this thesis is on margin-based classifiers. In that case, we want to find a decision function $f(\boldsymbol{x})$ and its associated classifier $\phi(\boldsymbol{x}) = \text{sign}[f(\boldsymbol{x})]$ which minimizes the misclassification rate. That is, once the classification function f is obtained, we use $\text{sign}(f(\boldsymbol{x}))$ to estimate the label of $\boldsymbol{x}$, i.e. $\hat{y} = +1$ if $f(\boldsymbol{x}) \geqslant 0$, and $\hat{y} = -1$ otherwise. Thus, the quantity $yf(\boldsymbol{x})$, which is called *functional margin*, is positive when the estimated class membership agrees with the true class membership, and negative when the observation $\boldsymbol{x}$ is misclassified. Moreover, we can think of the absolute value of $yf(\boldsymbol{x})$ as our 'confidence' in class label prediction, considering

the value of $f(\boldsymbol{x})$ close to zero indicates that x is near the decision boundary. Thus, high value of $yf(\boldsymbol{x})$ implies the classification for $\boldsymbol{x}$ is correct with much confidence, and as the value of $yf(\boldsymbol{x})$ goes to negative infinity, it means the classification was wrong with high confidence, which is not desirable. Hence, we can say that functional margin $yf(\boldsymbol{x})$ shows 'correctness' of the classification, and we generally want values of functional margin to be high.

To make use of the functional margin, one can think of finding the decision function $f(\boldsymbol{x})$ by minimizing the sum of values of a certain loss function in $yf(\boldsymbol{x})$. That is, minimizing $\sum_{i=1}^{n} L(y_i f(\boldsymbol{x}_i))$, where $L(u)$ is a loss function, can be a criterion to find a decision function $f(\boldsymbol{x})$. One of the natural loss functions is the $0-1$ loss function, $L(yf(\boldsymbol{x})) = I(yf(\boldsymbol{x}) \leqslant 0)$, which is hard to implement computationally. Hence, it is often to use convex surrogate loss functions in practice. However, this formulation often provides poor classification rules of $f(\boldsymbol{x})$, because of potential overfitting. A common solution to this is to add a constraint on the parameters to stabilize or to *shrink* the estimates. Then margin-based classifiers can be summarized using the regularization framework in the following form

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{n} L(y_i f(\boldsymbol{x}_i)) + \lambda J(f), \tag{1.1}$$

where $\mathcal{F}$ is the decision function class of interest, and $L(u)$ is the loss function which is a function of the margin $yf(\boldsymbol{x})$, $J(f)$ is the penalty term that controls the smoothness of the model, and λ is a tuning parameter which balances the tradeoff between those two. In some practice, one may also use $\min_{f \in \mathcal{F}} C \sum_{i=1}^{n} L(y_i f(\boldsymbol{x}_i)) + J(f)$ instead, but it is equivalent to (1.1) since λ plays the same role as $1/C$. The loss function controls goodness of fit of the model, and the penalization term helps avoid overfitting so that good generalization can be obtained.

In the literature, there exist a number of margin-based classifiers. Using different loss functions, we can formulate different classifiers such as the Support Vector Machine (SVM) (Vapnik, 1998; Cristianini and Shawe-Taylor, 2000), the Penalized Logistic Regression (PLR) (Wahba, 1999; Lin et al., 2000), Distance-Weighted Discrimination (DWD) (Marron et al., 2007) and so on. Due to the definition of the functional margin, many well-known margin-based methods use nonincreasing loss functions on $yf(x)$ which encourages large functional margin.

The loss function in (1.1) plays an important role for the corresponding classifier, and we can

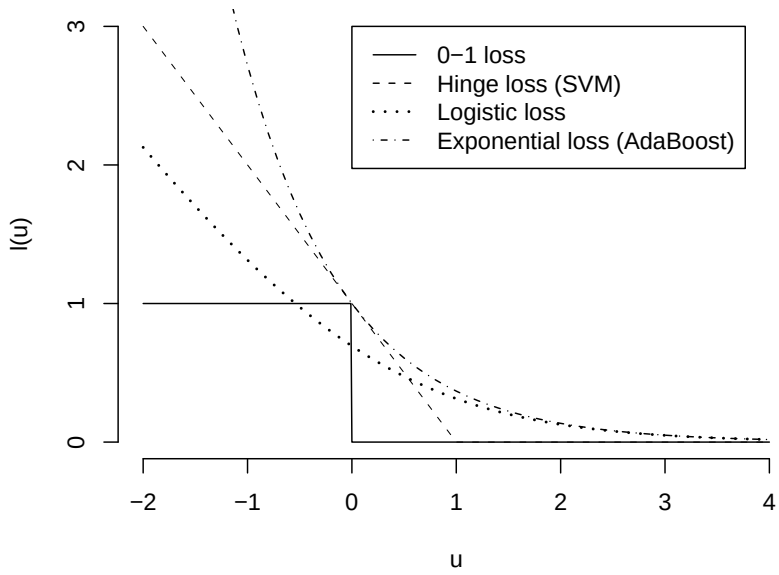

Figure 1.1: Plot of different loss functions.

modify the loss function to obtain different classifiers with desirable properties. One important contribution of this research is to study various modifications of the loss function to derive several classifiers with different properties.

Next we briefly overview several commonly used margin-based classifiers including the PLR, the SVM, and Boosting. Each of them can be understood as a special form of (1.1) with a different loss function $L(u)$. The loss functions of these classifiers are plotted in Figure 1.1 for graphical comparison.

1.2 Several Existing Methods

1.2.1 Penalized Logistic Regression

In the standard logistic regression model for binary classification, one assumes the logit, the log odds ratio, can be modeled as a linear function in covariates. Specifically, the model can be written as follows:

$$\log \frac{P(Y=+1|\boldsymbol{X})}{P(Y=-1|\boldsymbol{X})} = \boldsymbol{w}^T\boldsymbol{X} + b, \tag{1.2}$$

where $\boldsymbol{X}$ and Y denote the vector of explanatory variables and the class label, respectively. The coefficients of logistic regression $(\boldsymbol{w}, b)$ can be estimated by the method of Maximum Likelihood (ML) (McCullagh and Nelder, 1989). Once the ML estimators for (1.2) are obtained, the sign of $f(\boldsymbol{x})$, where $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$ can be used as the class membership estimates. This is because the model (1.2) implies that $P(Y = +1|\boldsymbol{X} = \boldsymbol{x}) > 0.5$ if $f(\boldsymbol{x}) > 0$, and $P(Y = +1|\boldsymbol{X} = \boldsymbol{x}) \leqslant 0.5$ otherwise.

The linear logistic regression can be generalized to the PLR by adding a constraint on the parameters. In particular, le Cessie and van Houwelingen (1992) proposed PLR, which maximizes the log-likelihood subject to a constraint on the L_2 norm of the coefficients. Wahba (1999) showed the linear PLR is equivalent to finding b and $\boldsymbol{w}$ which solves (1.1) where $\mathcal{F} = \{f : f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b\}$, $L(u) = l(u) = \log(1 + e^{-u})$, $J(f) = \frac{1}{2}\|\boldsymbol{w}\|_2^2$, and $\lambda > 0$ is a tuning parameter.

For a nonlinear problem, theory of reproducing kernel Hilbert spaces can be applied and then the kernel PLR has $\mathcal{F} = \{f : f(\boldsymbol{x}) = r(\boldsymbol{x}) + b, r(\boldsymbol{x}) \in \mathcal{H}_K\}$ and $J(f) = \|r\|_{\mathcal{H}_K}$, where $r(\boldsymbol{x}) = \sum_{i=1}^n v_i K(\boldsymbol{x}_i, \boldsymbol{x})$ and K is the kernel function (Wahba, 1999). Properties of the reproducing kernel and the representative theorem imply that $\|r\|_{\mathcal{H}_K}^2 = \boldsymbol{v}^T\boldsymbol{K}\boldsymbol{v}$ where $\boldsymbol{v} = (v_1, \ldots, v_n)^T$ and $\boldsymbol{K}$ is an $n \times n$ positive definite matrix with its $i_1 i_2$-th element $K(\boldsymbol{x}_{i_1}, \boldsymbol{x}_{i_2})$ (Kimeldorf and Wahba, 1971).

1.2.2 Support Vector Machine

The Support Vector Machine (SVM) can be viewed as a member of the regularization framework (1.1). It employs the hinge loss function $L(yf(\boldsymbol{x})) = [1 - yf(\boldsymbol{x})]_+$. (See Figure 1.1.) The value of $L(yf(\boldsymbol{x}))$ increases as $yf(\boldsymbol{x})$ becomes smaller and it stays at zero when $yf(\boldsymbol{x}) \geqslant 1$. That is, the SVM puts positive loss on the misclassified data points but 0 loss on the correctly classified observations once $yf(\boldsymbol{x})$ becomes greater than 1. Hence the data points with $yf(\boldsymbol{x}) \geqslant 1$ have no influence on the SVM solution. To explain further, we rewrite the SVM optimization in the following primal problem with the penalty term $J(f) = \frac{1}{2}\|\boldsymbol{w}\|^2$ for the standard SVM,

$$\min_{(b,\boldsymbol{w})} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^n [1 - yf(\boldsymbol{x})]_+. \tag{1.3}$$

To handle the hinge loss, we introduce n nonnegative slack variables, $\xi_i, i = 1, \cdots, n$. Then (1.3) is equivalent to

$$\begin{aligned} &\min_{(b,\boldsymbol{w})} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i \\ &\text{subject to} \quad \xi_i \geqslant 1 - y_i f(\boldsymbol{x}_i); \xi_i \geqslant 0, \forall i = 1, \cdots, n. \end{aligned}$$

We can transform this problem into its corresponding dual problem with the Lagrange multipliers γ_i and α_i, $i = 1, \cdots, n$, for contraints. The Lagrange primal function is

$$\mathfrak{L}(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\alpha_i[1 - y_i f(\boldsymbol{x}_i) - \xi_i] - \sum_{i=1}^{n}\gamma_i\xi_i \tag{1.4}$$

where $C = 1/\lambda$, and $\alpha_i \geqslant 0$ and $\gamma_i \geqslant 0$ are the Lagrange multipliers. Setting derivatives to zero, we have

$$\begin{aligned} \frac{\partial \mathfrak{L}}{\partial \boldsymbol{w}} &= \boldsymbol{w} - \sum_{i=1}^{n} y_i\alpha_i\boldsymbol{x}_i = \boldsymbol{0} && (1.5)\\ \frac{\partial \mathfrak{L}}{\partial b} &= -\sum_{i=1}^{n} y_i\alpha_i = 0 && (1.6)\\ \frac{\partial \mathfrak{L}}{\partial \xi_i} &= C - \alpha_i - \gamma_i = 0, && (1.7) \end{aligned}$$

with Karush-Kuhn-Tucker (KKT) conditions of the convex optimization theory

$$\alpha_i(1 - y_i f(\boldsymbol{x}_i) - \xi_i) = 0 \tag{1.8}$$

$$\gamma_i\xi_i = 0. \tag{1.9}$$

Substituting (1.5)-(1.9) into (1.4) gives the dual problem of the SVM

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle - \sum_{i=1}^{n}\alpha_i \\ &\text{subject to} \quad \sum_{i=1}^{n} y_i\alpha_i = 0; 0 \leqslant \alpha_i \leqslant C, \forall i = 1, \cdots, n. \end{aligned} \tag{1.10}$$

Using the α_i obtained from (1.10), $\boldsymbol{w}$ can be calculated as $\sum_{i=1}^{n}\alpha_i y_i \boldsymbol{x}_i$, and b can be obtained

by (1.7). Thus the decision boundary becomes $f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i y_i \langle \boldsymbol{x}_i, \boldsymbol{x} \rangle + b$. Because of (1.8), we can see that $\alpha_i > 0$ implies $y_i f(\boldsymbol{x}_i) \leqslant 1$ and actually that is the only case that $(\boldsymbol{x}_i, y_i)$ affects the solution. On the other hand, when $\alpha_i = 0$, the observation $(\boldsymbol{x}_i, y_i)$ has no impact on the solution. We call $\boldsymbol{x}_i$ with $\alpha_i > 0$ a Support Vector (SV), which is the observation misclassified or correctly classified but with less confidence, satisfying $y_i f(\boldsymbol{x}_i) \leqslant 1$.

1.2.3 Boosting

Boosting has been a very important machine learning method in the past 20 years. The original boosting algorithm, AdaBoost (Freund and Schapire, 1997), is an iterative procedure that combines many weak classifiers updating weights of training observations. In particular, initially a weak classifier is trained on the training data with all equal weights. Then, for each iteration, the weights of the misclassified observations are increased and the weak classifier is recalculated based on the newly weighted data. Then a score is assigned to the classifier based on the misclassification rate. After repeating this procedure for sufficiently many times, the final classifier is defined as weighted sum of all the classifiers from the iterations with the scores as weights.

Friedman et al. (2000) showed that the AdaBoost is approximating to fitting additive model using the exponential loss function. Thus, we can view the AdaBoost as a special member of regularization problem in (1.1) with loss $L(yf(\boldsymbol{x})) = \exp(-yf(\boldsymbol{x}))$. (See Figure 1.1.)

1.3 Outline

In the following chapters, we propose several new margin-based classifiers with various loss functions.

- In Chapter 2, we introduce the Robust Penalized Logistic Regression (RPLR) and study its properties. Moreover, we derive a computational algorithm as well as methods for class probability estimation and tuning parameter selection. Numerical demonstration includes simulated examples and the application on Lung Cancer Dataset.

- Chapter 3 proposes the Bounded Constraint Machine (BCM), and the Balancing Support Vector Machine (BSVM) as a bridge between the BCM and the standard SVM. We show

their properties, asymptotic behaviors, and the entire solution path for efficient computation. Numerical results include the simulated example and the Lung Cancer Data.

- In Chapter 4, we discuss multicategory classifiers and propose the multicategory Composite Least Squares (CLS) classifier. In addition, its properties, procedure for class probability estimation, and a computational algorithm are derived. Numerical results are included.

Proposed future work of each part and the proofs of our theorems are included at the end of each chapter.

Chapter 2

Truncation for robustness

2.1 Introduction

The PLR is a commonly used classification method in practice. It is a generalization of the standard logistic regression with a penalty term on the coefficients. Similar to the SVM, the PLR can be fit in the regularization framework with $loss + penalty$ (Wahba, 1999; Lin et al., 2000). The loss function controls goodness of fit of the model, and the penalization term helps avoid overfitting so that good generalization can be obtained.

For the standard SVM, its hinge loss function is unbounded, as a result, the SVM classifier can be sensitive to outliers (Shen et al., 2003; Liu and Shen, 2006). Wu and Liu (2007) proposed the Robust SVM (RSVM) as a modification of the original SVM by truncating the hinge loss function. They showed that through the operation of truncation, the impact of outliers can be reduced, consequently, the resulting classifier may be more robust.

Comparing to the SVM, the PLR uses the logistic loss which is also unbounded. Therefore, similar to the SVM, the PLR can be sensitive to extreme outliers as well. In this chapter, we propose the Robust Penalized Logistic Regression (RPLR), which uses truncated logistic loss function. Because truncation reduces the impact of misclassified outliers, the RPLR is more robust and accurate than the standard PLR. Comparisons of the proposed RPLR with the existing robust logistic regression methods are discussed as well.

One important aspect of classification is class probability estimation. A good estimated class probability can not only give the class prediction, it should also reflect the strength of classification. Therefore, class probability estimation is desirable in many applications. In

the PLR, one can use the estimated classification function, i.e. the estimated logit function, to derive the corresponding probability estimate. When we replace the logisitic loss by its truncated version, properties of the corresponding classification function may not preserve all class probability information any more. To solve this problem, we propose three different schemes for class probability estimation. Properties and performance of these three schemes are explored as well.

Although the original logistic loss function is convex, its truncated version becomes non-convex. Consequently, the corresponding minimization problem involves difficult non-convex optimization. To implement the RPLR, we decompose the non-convex truncated logistic loss function into the difference of two convex functions. Then, using this decomposition, we apply the difference convex (d.c.) algorithm to obtain the solution of the RPLR through iterative convex minimization.

The tuning parameter plays an important role in the RPLR implementation. To select a good tuning parameter, we develop the Estimated Generalized Approximate Cross Validation (EGACV) procedure and compare its performance with the cross validation method.

In the following sections, we describe the new proposed method in more details with theoretical justification and numerical examples. Section 2.2 reviews the PLR and gives a maximum likelihood interpretation. In Section 2.3 we review some related robust logistic regression methods in the literature. In Section 2.4 we describe the RPLR and explore its theoretical properties. The methods for class probability estimation are also introduced. Section 2.5 develops the d.c. algorithm to solve the nonconvex minimization problem for the RPLR. In Section 2.6 we explore various ways to select the tuning parameter. Numerical results are presented in Section 2.7 and Section 2.8 provides some discussions. The proofs of theorems are included in Section 2.9.

2.2 Penalized Logistic Regression

As mentioned in Section 1.2.1, the PLR solves (1.1) with the logistic loss function $l(u) = \log(1 + e^{-u})$. Here, we briefly review the PLR and its likelihood interpretation.

Notice that the loss function $l(u) = \log(1 + e^{-u})$ is a smooth decreasing function as shown in the middle panel of Figure 2.1 and in particular, its value grows rapidly as u goes to negative

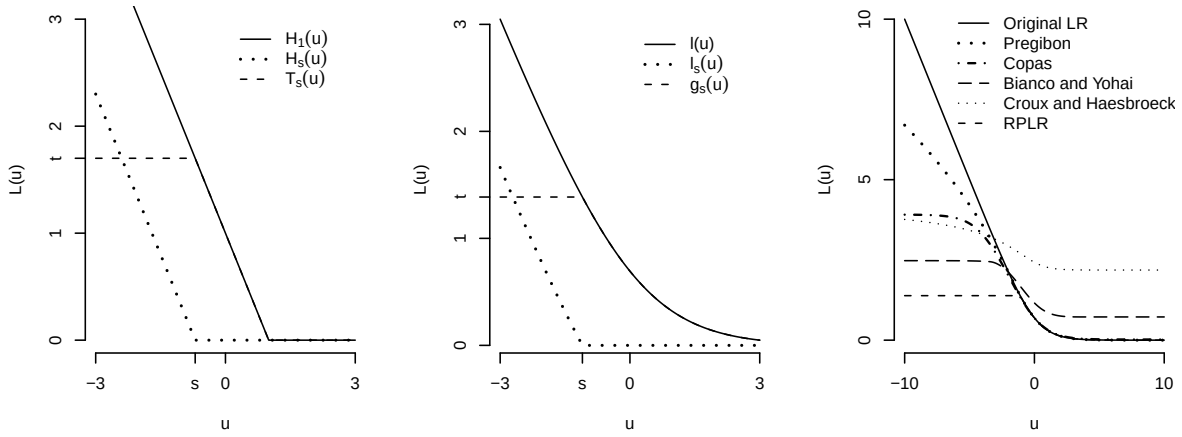

Figure 2.1: Left: Plot of the functions $H_1(u)$, $H_s(u)$, and $T_s(u)$ with $H_s(u) = [H_1(u) - H_1(s)]_+$ and $T_s(u) = H_1(u) - H_s(u)$; Middle: Plot of the functions $l(u)$, $l_s(u)$, and $g_s(u)$ with $l_s(u) = [l(u) - l(s)]_+$ and $g_s(u) = l(u) - l_s(u)$; Right: Plot of the loss functions of the original logistic regression, Pregibon's resistant fitting model, Copas' misclassification model, and the RPLR.

infinity. This causes high impact of outliers with very small (negative) value of $y_i f(\boldsymbol{x}_i)$. As a result, the coefficient estimates of the PLR can be affected by outliers far from their own classes. To further illustrate the effect of outliers on the PLR, we randomly generate 2-dimensional separable data and apply the PLR to obtain a classification boundary. As shown in the left panel of Figure 2.2, the PLR works very well without outliers. However, if we randomly select one of the observations and move it away from its own class, then the classification boundary of the PLR is pulled towards to that outlier, as shown in the right panel of Figure 2.2. As a result, the corresponding misclassification rate will become higher. In contrast, our new proposed method is much more robust to the outlier so that its classification boundary is more accurate.

The effect of outliers on the PLR can also be interpreted using maximum likelihood. The likelihood function of unpenalized logistic regression can be written as

$$\mathcal{L}(b, \boldsymbol{w}) = \prod_{i=1}^{n} P(\boldsymbol{x}_i)^{\frac{1+y_i}{2}} (1 - P(\boldsymbol{x}_i))^{\frac{1-y_i}{2}}, \tag{2.1}$$

where $P(\boldsymbol{x}) = P(Y = +1 | \boldsymbol{X} = \boldsymbol{x})$. Then, we can plug in the logit function (1.2) into (2.1), and the corresponding maximizer of $\mathcal{L}(b, \boldsymbol{w})$ is the solution of the logistic regression. Note that the i-th term of the product in the likelihood is $P(\boldsymbol{x}_i)$ when $y_i = +1$, and $1 - P(\boldsymbol{x}_i)$, otherwise.

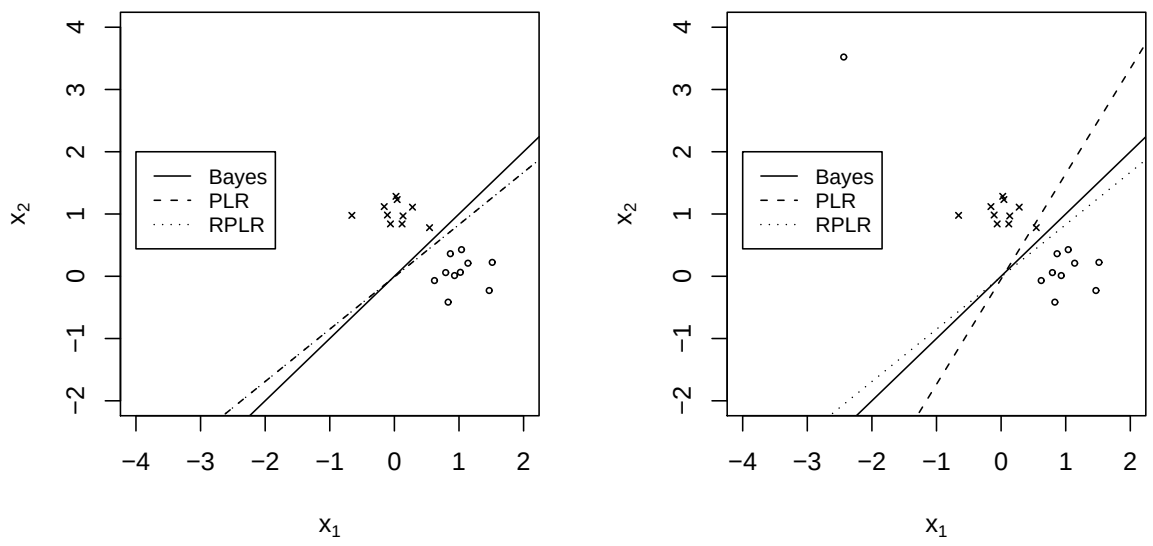

Figure 2.2: Illustration plot of the effect of outliers with an outlier far away from its own class. The RPLR boundary is much robust than that of the original PLR.

Therefore to maximize the likelihood, one needs to find $(\boldsymbol{w}, b)$ to make $P(\boldsymbol{x}_i)$ big when $y_i = 1$ and small when $y_i = -1$. However, this could be sensitive to outliers. To illustrate this further, assume there is one data point x_i with $y_i = +1$ which locates far from the other data points of class $+1$ but closer to data of class -1 as illustrated in the right panel of Figure 2.2. Using the solution $(\boldsymbol{w}, b)$ without the outlier, the corresponding $P(\boldsymbol{x}_i)$ for the outlier will be very small because $\boldsymbol{x}_i$ is closer to the data of class -1. Consequently, the ML method would select $(\boldsymbol{w}, b)$ which will make $P(\boldsymbol{x}_i)$ larger to obtain bigger likelihood at the expense of other entries' classification accuracy. This results in the boundary moving towards to the outlier. In the next section, we discuss some literature on robust logistic regression.

2.3 Literature on Robust Logistic Regression

There is a large literature on the robustness issue of the Logistic Regression. Most of the existing methods attempt to achieve robustness by downweighting observations which are far from the majority of the data, i.e. outliers (Copas, 1988; Carroll and Pederson, 1993; Pregibon, 1982; Bianco and Yohai, 1996; Bondell, 2005; Stefanski et al., 1986; Künsch et al., 1989; Krasker and

Welsch, 1982; Morgenthaler, 1992). Stefanski et al. (1986) and Künsch et al. (1989) modified original score function of the logistic regression to obtain bounded sensitivity, which is a concept introduced by Krasker and Welsch (1982). Morgenthaler (1992) used L_1-norm instead of L_2-norm in the likelihood, resulting in a weighted score function of the original score function. Cantoni and Ronchetti (2001) focused on robustness of inference rather than the model.

Pregibon (1982) suggested resistant fitting methods which taper the standard likelihood to reduce the influence of extreme observations. In particular, he proposed to estimate $(\boldsymbol{w}, b)$ by solving

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{n} h(\boldsymbol{x}_i) \rho \left(\frac{d_i}{h(\boldsymbol{x}_i)} \right), \tag{2.2}$$

where $\rho(u)$ is a tapering function, $h(\boldsymbol{x})$ is a factor which controls leverage of each observation, and d_i is negative log likelihood, that is, $d_i = -\left[\frac{1+Y_i}{2} \log P(\boldsymbol{x}_i) + \frac{1-Y_i}{2} \log(1 - P(\boldsymbol{x}_i)) \right]$. Note that this reduces to standard maximum likelihood estimation of the logistic regression when $h(\boldsymbol{x}) \equiv 1$ and $\rho(u) = u$. The particular tapering function Pregibon (1982) proposed to use is the Huber's loss function

$$\rho(u) = \begin{cases} u & \text{if } u \leqslant H, \\ 2(uH)^{1/2} - H & \text{otherwise,} \end{cases} \tag{2.3}$$

where H is a prespecified constant. In order to compare with our new method, we provide a new view of the method by Pregibon (1982) in the loss function framework. In particular, with ρ in (2.3) and $h(\boldsymbol{x}) \equiv 1$, we can reduce (2.2) to

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{n} l^{\text{Pregibon}}(y_i f(\boldsymbol{x}_i)),$$

where

$$l^{\text{Pregibon}}(u) = \rho(l(u)) = \begin{cases} \log\left(1 + e^{-u}\right) & \text{if } u \geqslant -\log\left(e^{H} - 1\right) \\ 2(H \log\left(1 + e^{-u}\right))^{1/2} - H & \text{otherwise.} \end{cases} \tag{2.4}$$

The estimate in (2.4) was shown to have approximately 95% asymptotic relative efficiency when $H = 1.345^2$. The loss function in (2.4) with $H = 1.345^2$ is plotted in the right panel of

Figure 2.1 for comparison. As shown in the plot, $l^{\text{Pregibon}}(u)$ grows as u goes to negative infinity, but less rapidly than the loss function of the original logistic regression $l(u)$. Consequently, the resulting coefficient estimates become less sensitive to extreme observations. However, the value of $l^{\text{Pregibon}}(u)$ remains to be unbounded, hence the impact of outliers can still be large.

Bianco and Yohai (1996) proposed a consistent and more robust version of Pregibon's estimator, by adding a bias correction term. More specifically, they suggested to solve

$$\min_{f\in\mathcal{F}} \sum_{i=1}^{n} \rho(d_i) + C_i, \tag{2.5}$$

with the d_i previously defined and the bias correction term C_i, where $C_i = G(P(\boldsymbol{x}_i)) + G(1 - P(\boldsymbol{x}_i)) - G(1)$, $G(t) = \int_0^t \rho'(-\log u)du$, and

$$\rho(t) = \begin{cases} t - \frac{t^2}{2c} & \text{if } t \leqslant c \\ \frac{c}{2} & \text{otherwise,} \end{cases} \tag{2.6}$$

where c is a constant. Croux and Haesbroeck (2003) pointed out that the minimizer of (2.5) with $\rho(t)$ in (2.6) does not exist quite often, in particular, the minimizer tends to be infinity. To overcome this problem, they suggested to use

$$\rho(t) = \begin{cases} t\mathrm{e}^{-\sqrt{d}} & \text{if } t \leqslant d \\ -2\mathrm{e}^{-\sqrt{t}}(1+\sqrt{t}) + \mathrm{e}^{-\sqrt{d}}(2(1+\sqrt{d})+d) & \text{otherwise,} \end{cases} \tag{2.7}$$

and

$$G(t) = \begin{cases} t\mathrm{e}^{-\sqrt{-\log t}} + \mathrm{e}^{1/4}\sqrt{\pi}\Phi(\sqrt{2}(\frac{1}{2}+\sqrt{-\log t})) - \mathrm{e}^{1/4}\sqrt{\pi} & \text{if } t \leqslant d \\ \mathrm{e}^{-\sqrt{d}}t - \mathrm{e}^{-1/4}\sqrt{\pi} + \mathrm{e}^{1/4}\sqrt{\pi}\Phi(\sqrt{2}(\frac{1}{2}+\sqrt{d})) & \text{otherwise,} \end{cases} \tag{2.8}$$

where d is a constant and Φ is the normal cumulative distribution function. To view the method by Croux and Haesbroeck (2003) in the loss function framework, we show that the problem (2.5) with $\rho(t)$ in (2.7) is equivalent to solving

$$\min_{f\in\mathcal{F}} \sum_{i=1}^{n} l^{\text{CH}}(y_i f(\boldsymbol{x}_i)), \tag{2.9}$$

where

$$
\begin{aligned}
l^{\mathrm{CH}}(u) = \quad & I_{\{u \geqslant -\log(\mathrm{e}^d-1)\}} \left[\log(1+\mathrm{e}^{-u})\mathrm{e}^{-\sqrt{d}} + \mathrm{e}^{-\sqrt{d}}\tfrac{1}{1+\mathrm{e}^{-u}} - \mathrm{e}^{-1/4}\sqrt{\pi} + \mathrm{e}^{1/4}\sqrt{\pi}\Phi(\sqrt{2}(\tfrac{1}{2}+\sqrt{d}))\right] \\
& + I_{\{u < -\log(\mathrm{e}^d-1)\}} \left[-2\mathrm{e}^{-\sqrt{\log(1+\mathrm{e}^{-u})}}(1+\sqrt{\log(1+\mathrm{e}^{-u})}) + \mathrm{e}^{-\sqrt{d}}(2(1+\sqrt{d})+d)\right. \\
& \left. \tfrac{1}{1+\mathrm{e}^{-u}}\mathrm{e}^{-\sqrt{\log(1+\mathrm{e}^{-u})}} + \mathrm{e}^{1/4}\sqrt{\pi}\Phi(\sqrt{2}(\tfrac{1}{2}+\sqrt{\log(1+\mathrm{e}^{-u})})) - \mathrm{e}^{-1/4}\sqrt{\pi}\right] \\
& + I_{\{u \geqslant \log(\mathrm{e}^d-1)\}} \left[\tfrac{1}{1+\mathrm{e}^{u}}\mathrm{e}^{-\sqrt{\log(1+\mathrm{e}^{u})}} + \mathrm{e}^{1/4}\sqrt{\pi}\Phi(\sqrt{2}(\tfrac{1}{2}+\sqrt{\log(1+\mathrm{e}^{u})})) - \mathrm{e}^{-1/4}\sqrt{\pi}\right] \\
& + I_{\{u < \log(\mathrm{e}^d-1)\}} \left[\mathrm{e}^{-\sqrt{d}}\tfrac{1}{1+\mathrm{e}^{u}} - \mathrm{e}^{-1/4}\sqrt{\pi} + \mathrm{e}^{1/4}\sqrt{\pi}\Phi(\sqrt{2}(\tfrac{1}{2}+\sqrt{d}))\right].
\end{aligned}
\tag{2.10}
$$

The loss function (2.10) is plotted in the right panel of Figure 2.1.

Another attempt to achieve robustness was made by Copas (1988), who modeled contamination of class labels in the training data. Specifically, it is assumed that the class label $y \in \{1, -1\}$ was transposed with a small probability γ. As a result, the response y can be 1 with probability $P^*(\boldsymbol{x})$, where

$$
P^*(\boldsymbol{x}) = (1-\gamma)P(\boldsymbol{x}) + \gamma(1-P(\boldsymbol{x})). \tag{2.11}
$$

Using (1.2) and (2.11), the log-likelihood with $P^*(\boldsymbol{x})$ becomes

$$
\begin{aligned}
& \sum_{i=1}^{n}\left[\frac{1+Y_i}{2}\log P^*(\boldsymbol{x}_i) + \frac{1-Y_i}{2}\log(1-P^*(\boldsymbol{x}_i))\right] \\
& = \sum_{i=1}^{n}\left[\frac{1+Y_i}{2}\log\frac{1+\gamma(e^{-f(\boldsymbol{x}_i)}-1)}{1+e^{-f(\boldsymbol{x}_i)}} + \frac{1-Y_i}{2}\log\frac{1+\gamma(e^{f(\boldsymbol{x}_i)}-1)}{1+e^{f(\boldsymbol{x}_i)}}\right] \\
& = \sum_{i=1}^{n}\left[I_{(Y_i=1)}\log\frac{1+\gamma(e^{-Y_i f(\boldsymbol{x}_i)}-1)}{1+e^{-Y_i f(\boldsymbol{x}_i)}} + I_{(Y_i=-1)}\log\frac{1+\gamma(e^{-Y_i f(\boldsymbol{x}_i)}-1)}{1+e^{-Y_i f(\boldsymbol{x}_i)}}\right] \\
& = \sum_{i=1}^{n}\log\frac{1+\gamma(e^{-Y_i f(\boldsymbol{x}_i)}-1)}{1+e^{-Y_i f(\boldsymbol{x}_i)}}.
\end{aligned}
\tag{2.12}
$$

To view this in the loss framework, we get the equivalent problem of log likelihood maximization in (2.12) as follows

$$
\min_{f\in\mathcal{F}}\sum_{i=1}^{n} l^{\mathrm{Copas}}(y_i f(\boldsymbol{x}_i)), \tag{2.13}
$$

where $l^{\mathrm{Copas}}(u) = \log\frac{1+e^{-u}}{1+\gamma(e^{-u}-1)}$, which is plotted with $\gamma = 0.02$ in the right panel of Figure 2.1. With any γ smaller than 0.5, $l^{\mathrm{Copas}}(u)$ is decreasing in u, and bounded by $-\log\gamma$. Though it reduces the impact of outliers, it heavily depends on the misclassification rate γ, which is unknown and needs to be tuned. In the next section, we propose a new classifier which effectively

reduces the influence of outliers by truncating the logistic loss function.

2.4 Robust Penalized Logistic Regression

2.4.1 Truncated Loss for Robustness

Although most of the previous methods done on robust logistic regression takes the likelihood point of view, it can be transformed into the loss function framework as shown in the previous section. In this thesis, we take a different approach to achieve robustness for the logistic regression. In particular, we develop a new classifier by truncating the loss function directly rather than modifying the log likelihood function.

Due to the unboundedness of the logistic loss function, it assigns large loss values for points far from their own classes. Consequently, the resulting classifiers will be affected by those outliers. To reduce the effect of outliers, we propose a novel Robust version of the PLR (RPLR), which truncates the loss function of the PLR. Specifically, we propose to use the truncated logistic loss function $g_s(u) = \min(l(u), l(s))$ instead of $l(u)$. Here $s \leqslant 0$ represents the location of truncation. As illustrated in the middle panel of Figure 2.1, $g_s(yf(\boldsymbol{x}))$ increases as $yf(\boldsymbol{x})$ decreases, but once $yf(\boldsymbol{x})$ becomes less than s, $g_s(yf(\boldsymbol{x}))$ becomes a constant. This implies that g_s becomes bigger as an observation gets further away from the classification boundary up to an upperbound. For outliers located further away from the boundary satisfying $yf(\boldsymbol{x}) \leqslant s$, the loss stays at a constant $l(s)$ so that the outliers cannot further influence the classification boundary. This is in contrast to the untruncated version whose impact grows to infinity. Also, it differs from the other existing methods we covered in the previous section in the sense that the effect of extreme observations stays the same once $yf(\boldsymbol{x})$ becomes less than s, while that of others keeps increasing. Note that s determines the level of truncation. When $s = -\infty$, no truncation occurs, thus the loss is the same as the original logistic loss. As s gets closer to 0, we have more truncation on the loss which may reduce the effect of outliers further. Therefore, $g_s(u)$ contains a group of loss functions indexed by s.

Similar idea of truncation has been applied to the SVM to derive the RSVM in Wu and Liu (2007). They truncated the original hinge loss of the SVM $H_1(u)$ at s, resulting in the truncated hinge loss function $T_s(u) = \min(H_1(u), H_1(s)) = H_1(u) - H_s(u)$, where $H_s(u) =$

$[H_1(u) - H_1(s)]_+$. As shown in the left panel of the Figure 2.1, the truncated hinge loss function $T_s(yf(\boldsymbol{x}))$ stays the same once $yf(\boldsymbol{x})$ becomes less than s, similarly to $g_s(yf(\boldsymbol{x}))$. But once $yf(\boldsymbol{x})$ becomes greater than 1, the loss function of the RSVM $T_s(yf(\boldsymbol{x}))$ becomes 0. In contrast, the loss function of the RPLR $g_s(yf(\boldsymbol{x}))$ remains small but positive. That is, the RSVM does not use the information about data points with $yf(\boldsymbol{x}) > 1$, while the RPLR uses all the data points to build a classification boundary. This can be beneficial for the RPLR as reflected in the simulation results in Section 2.7 .

From the likelihood point of view, minimizing $\sum_{i=1}^{n} g_s(y_i f(\boldsymbol{x}_i))$ is equivalent to maximizing

$$\prod_{i=1}^{n} Q^+(\boldsymbol{x}_i)^{\frac{1+y_i}{2}} (1 - Q^-(\boldsymbol{x}_i))^{\frac{1-y_i}{2}}, \tag{2.14}$$

where $Q^+(\boldsymbol{x}) = \max(P(\boldsymbol{x}), \frac{1}{1+e^{-s}})$ and $Q^-(\boldsymbol{x}) = \min(P(\boldsymbol{x}), \frac{1}{1+e^{s}})$. Interestingly, (2.14) has a similar form as that of the logistic regression in (2.1). The difference is that the i-th factor is $Q^+(\boldsymbol{x}_i)$ or $1 - Q^-(\boldsymbol{x}_i)$, instead of $P(\boldsymbol{x}_i)$ and $1 - P(\boldsymbol{x}_i)$, depending on y_i. Hence, maximizing (2.14) is equivalent to finding $(\boldsymbol{w}, b)$ which gives big $Q^+(\boldsymbol{x})$ when $y = +1$ and small $Q^-(\boldsymbol{x})$ when $y = -1$. By definition, $Q^+(\boldsymbol{x})$ can not get extremely small because it is lower bounded by $(1 + e^{-s})^{-1}$. Similarly, $Q^-(\boldsymbol{x})$ can not get extremely big. Therefore outliers may not influence (2.14) as much comparing to (2.1). As a result, the maximizer of (2.14) can be less sensitive to outliers. For the toy example illustrated in Figure 2.2, the classification boundary of the original PLR deteriorates dramatically when there exists an extreme outlier in the dataset. In contrast, the RPLR boundary is very stable whether there is an outlier or not.

2.4.2 Fisher Consistency

In this section, we study Fisher consistency of robust logistic regression and its weighted version. Fisher consistency, also known as classification-calibration (Bartlett et al., 2006), requires that the population minimizer of a binary loss function has the same sign as $P(\boldsymbol{x}) - 1/2$ (Lin, 2004). Wu and Liu (2007) established the conditions of a truncated loss for Fisher consistency. In particular, the binary truncated logistic loss function $g_s(u) = \min(l(u), l(s))$ is Fisher-consistent for any $s \leqslant 0$. For the multicategory case with k classes, $g_s(u)$ is Fisher consistent for $s \in [-\log(2^{k/(k-1)} - 1), 0]$, which reduces to $s \in [-\log 3, 0]$ when $k = 2$. In this paper,

we consider three different truncation locations $s = 0$, $-\log 3$, and $-\log 7$ for the RPLR. The corresponding values of the logistic loss are $l(0)$, $2l(0)$, and $3l(0)$, respectively. Our numerical results suggest that $s = -\log 3$ with $l(s) = 2l(0)$ gives the best performance. This matches the Fisher consistency results for multicategory classification.

So far, we have focused on the standard case, i.e., treating different types of misclassification equally. Sometimes, it can be natural to impose different costs for different types of misclassification. For example, it can be more severe to misclassify an observation of class $+1$ to class -1 than that of class -1 to $+1$. Then it is sensible to put a bigger cost for the first kind of misclassification than the second type. Lin et al. (2002) discussed the weighted SVM to deal with nonstandard situations such as different misclassification costs for different classes. Recently, Wang et al. (2007) applied weighted learning to large margin classifiers for probability estimation. In addition to Fisher consistency of non-weighted robust logistic regression, we investigate similar properties of the weighted robust logistic regression.

Let $(1-\pi, \pi)$ with $0 < \pi < 1$ be the weights for class $+1$ and class -1 respectively, then the weighted version of the RPLR becomes

$$\min_{f \in \mathcal{F}} (1-\pi) \sum_{y_i=1} g_s(y_i f(\boldsymbol{x}_i)) + \pi \sum_{y_i=-1} g_s(y_i f(\boldsymbol{x}_i)) + \frac{\lambda}{2} J(f), \tag{2.15}$$

where $\lambda > 0$ balances the goodness of fit, measured by the loss function, and the smoothness of f. If $\lambda = 0$, the objective function in (2.15) reduces to the unpenalized robust logistic regression. Note that the expectation of the weighted loss part in (2.15) is $E[h_\pi(Y) g_s(Y f(\boldsymbol{X}))]$, where $h_\pi(1) = 1-\pi$ and $h_\pi(-1) = \pi$.

To understand the RPLR further, we need to explore the property of weighted robust logistic regression. The following theorem discusses the theoretical minimizer of the truncated logistic loss.

Theorem 1. *The minimizer f_π^* of $E[h_\pi(Y) g_s(Y f(\boldsymbol{X}))]$ has the same sign as $P(\boldsymbol{x}) - \pi$.*

Theorem 1 indicates that the sign of f_π^* is the same as $\text{sign}(P(\boldsymbol{x}) - \pi)$. Thus, $\text{sign}(f_\pi^*)$ provides a natural estimate of $\text{sign}(P(\boldsymbol{x}) - \pi)$. In particular, if $f_\pi^* > 0$, then $P(\boldsymbol{x}) > \pi$, otherwise $P(\boldsymbol{x}) \leqslant \pi$. This offers a natural procedure for class probability estimation. In particular, one can estimate f_π^* for many different π's $\in (0, 1)$ to obtain further information about $P(\cdot)$. Thus,

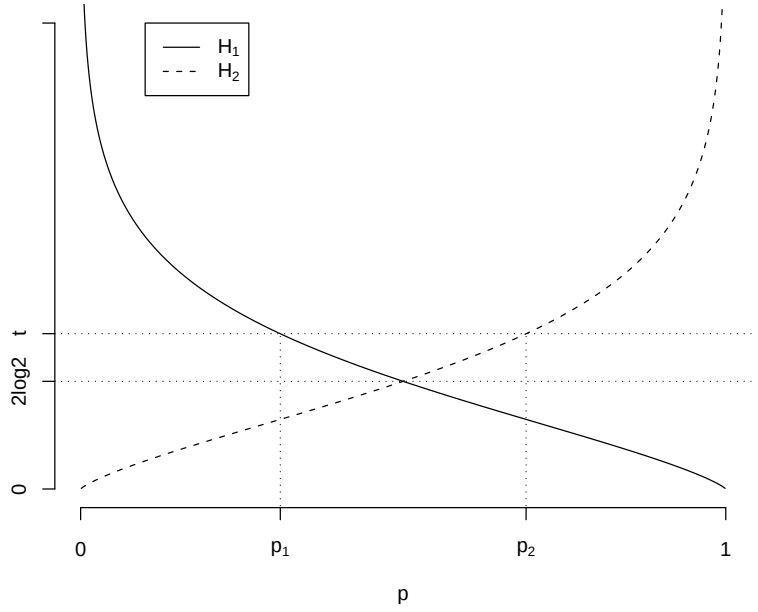

Figure 2.3: Plot of H_1 and H_2 for Theorem 2 in Section 2.4.3. The condition $t > H_1(\pi, p)$ and $t > H_2(\pi, p)$ hold only when $p \in [p_1, p_2]$.

it can be used for class probability estimation, as discussed further in Section 2.4.3.

2.4.3 Probability Estimation

Lin (2002) showed that under certain conditions the solution $\hat{f}_\pi$ of (2.15) approaches $f_\pi^* = \operatorname{argmin} E[h_\pi(Y)g_s(Yf(\boldsymbol{X}))]$. Therefore, we can use the property of f_π^* to design estimators of class probabilities $\hat{P}(\boldsymbol{x})$. In the simplest scenario where $\pi = 1/2$ and $s = -\infty$, we use the regular logistic loss and (2.15) reduces to the ordinary PLR. In that case, it is well known that the minimizer of $E[l(Yf(X))]$ is $f = \log[p(X)/(1-p(X))]$. Then a natural estimator of $P(\boldsymbol{x})$ is $e^{\hat{f}}/(1+e^{\hat{f}})$.

When we use the truncated loss function, the minimizer of $E[h_\pi(Y)g_s(Yf(X))]$ does not always maintain enough information to obtain class probability estimation. The following theorem establishes the minimizer of $E[h_\pi(Y)g_s(Yf(X))]$.

Theorem 2. *Define* $H_1(\pi, P(\boldsymbol{x})) = \log[1 + 1/\tau(P(\boldsymbol{x}), \pi)] + [1/\tau(P(\boldsymbol{x}), \pi)] \log[1 + \tau(P(\boldsymbol{x}), \pi)]$, $H_2(\pi, P(\boldsymbol{x})) = \tau(P(\boldsymbol{x}), \pi) \log[1 + 1/\tau(P(\boldsymbol{x}), \pi)] + \log[1 + \tau(P(\boldsymbol{x}), \pi)]$, *and* $\tau(P(\boldsymbol{x}), \pi) = \frac{(1-\pi)P(\boldsymbol{x})}{\pi(1-P(\boldsymbol{x}))}$.

Then, for $t = g_s(s)$,

$$f_\pi^* = \begin{cases} \log \tau(P(\boldsymbol{x}), \pi) & \text{if } t > H_1(\pi, P(\boldsymbol{x})) \text{ and } t > H_2(\pi, P(\boldsymbol{x})) \\ -\infty & \text{if } t < H_1(\pi, P(\boldsymbol{x})) \text{ and } H_1(\pi, P(\boldsymbol{x})) > H_2(\pi, P(\boldsymbol{x})) \\ \infty & \text{if } t < H_2(\pi, P(\boldsymbol{x})) \text{ and } H_1(\pi, P(\boldsymbol{x})) < H_2(\pi, P(\boldsymbol{x})) \\ -\infty, \infty & \text{if } t < H_1(\pi, P(\boldsymbol{x})) = H_2(\pi, P(\boldsymbol{x})). \end{cases}$$

Theorem 2 implies that we can use f_π^* to express class probability only when $f_\pi^* = \log \tau(P(\boldsymbol{x}), \pi) = \log \frac{(1-\pi)P(\boldsymbol{x})}{\pi(1-P(\boldsymbol{x}))}$. Otherwise we cannot reconstruct $P(\boldsymbol{x})$ using f_π^*. To further illustrate the relationship between f_π^* and $P(\boldsymbol{x})$, we consider H_1 and H_2 in the case that $\pi = 1/2$, which is plotted in Figure 2.3. When $P(\boldsymbol{x}) \in [p_1, p_2]$ with $t = H_1(\pi, p_1)$ and $t = H_2(\pi, p_2)$, then $f_\pi^* = \log \frac{(1-\pi)P(\boldsymbol{x})}{\pi(1-P(\boldsymbol{x}))}$. However, when $P(\boldsymbol{x}) \notin [p_1, p_2]$, f_π^* is either ∞ or $-\infty$, which does not have enough information to recover $P(\boldsymbol{x})$. For this reason, we need to explore other schemes to estimate $P(\boldsymbol{x})$.

To estimate the class probability, we propose the following three schemes.

Scheme 1 Since the RPLR works only for estimation of $P(\boldsymbol{x}) \in [p_1, p_2]$, we can consider utilizing it for those p, and using the ordinary PLR for $P(\boldsymbol{x}) \notin [p_1, p_2]$. Notice that this scheme is valid only for $t > 2\log 2$, because if $t \leqslant 2\log 2$, $p_1 = p_2$ and t is smaller than H_1 and H_2 for any $P(\boldsymbol{x})$ as shown in Figure 2.3. Thus by Theorem 2, the RPLR does not work for estimation of any $P(\boldsymbol{x})$ when $t \leqslant 2\log 2$.

This scheme is a valid approach in the sense that estimation of $P(\boldsymbol{x}) \in [p_1, p_2]$ is more critical than that of $P(\boldsymbol{x}) \notin [p_1, p_2]$. Usually the data points with very small $P(\boldsymbol{x})$ or very big $P(\boldsymbol{x})$ are easier to classify and we are more certain about the class membership of those points. However, class membership prediction for data points with $P(\boldsymbol{x})$ near $1/2$ is not only difficult, but also highly affected by outliers. Thus estimation of the class probability becomes more important for those points. Therefore, we use the RPLR for estimation of $P(\boldsymbol{x}) \in [p_1, p_2]$, and use the ordinary PLR for $P(\boldsymbol{x}) \notin [p_1, p_2]$.

Scheme 2 The second scheme is motivated by the idea that we can shift p_1 and p_2 by changing π. Because H_1 and H_2 in Theorem 2 depend on π, different π's bring different estimable region $[p_1, p_2]$. Hence, we can cover most of $P(\boldsymbol{x}) \in [0, 1]$ using many different π's. Note

that this method is applicable only when $t > 2\log 2$, and here we illustrate the case with $t = 3\log 2$. More specifically, we use seven different π's such as $\pi_1 = 1/2$, $\pi_2 = 1/5$, $\pi_3 = 4/5$, $\pi_4 = 1/20$, $\pi_5 = 19/20$, $\pi_6 = 1/91$, $\pi_7 = 90/91$, which give different estimable regions for $P(\boldsymbol{x})$, $[0.310, 0.690]$, $[0.105, 0.358]$, $[0.642, 0.899]$, $[0.024, 0.101]$, $[0.895, 0.976]$, $[0.005, 0.024]$, $[0.976, 0.995]$. Using $\hat{f}_j$ which denotes the solution from the RPLR with π_j, we can construct the estimator $\hat{P}^j(\boldsymbol{x}) = e^{\hat{f}_j}/(1+e^{\hat{f}_j})$; $j = 1, \cdots, 7$, to estimate $P(\boldsymbol{x})$ in the corresponding region.

There are some drawbacks of the second scheme. First, there are overlaps between the estimable regions. Moreover, the RPLR with different π's can give contradictory inference about $P(\boldsymbol{x})$. To solve this, for given $\hat{P}^j(\boldsymbol{x})$, we consider $\hat{P}^1(\boldsymbol{x})$ first. If $\hat{P}^1(\boldsymbol{x}) \in [0.310, 0.690]$, then take $\hat{P}^1(\boldsymbol{x})$ as $\hat{P}(\boldsymbol{x})$. Otherwise, we consider $\hat{P}^2(\boldsymbol{x})$ or $\hat{P}^3(\boldsymbol{x})$ depending on whether $\hat{P}^1(\boldsymbol{x})$ is less than 0.310 or greater than 0.690. Then take $\hat{P}^2(\boldsymbol{x})$ or $\hat{P}^3(\boldsymbol{x})$ as $\hat{P}(\boldsymbol{x})$ if it falls in the estimable region, otherwise, take $\hat{P}^4(\boldsymbol{x})$ or $\hat{P}^5(\boldsymbol{x})$ in the same manner as $\hat{P}(\boldsymbol{x})$ or use $\hat{P}^6(\boldsymbol{x})$ or $\hat{P}^7(\boldsymbol{x})$ likewise. If the RPLR with $\hat{P}^j(\boldsymbol{x})$ gives contradictory inference about $P(\boldsymbol{x})$ or none of them gives the estimate of $P(\boldsymbol{x})$ in the estimable region, then we use the PLR to estimate $P(\boldsymbol{x})$.

Scheme 3 Wang et al. (2007) suggested to estimate the class probability for large margin classifiers via bracketing the probability using multiple weighted classifiers. We consider to apply the same idea to the RPLR. First, we make equally spaced partitions of $[0, 1]$, that is, $0 = \pi_0 < \pi_1 < \cdots < \pi_m < \pi_{m+1} = 1$ such that $\pi_{j+1} - \pi_j$ is constant for any $i = 0, \cdots, m$. Then we can obtain $\hat{f}_j$, $j = 1, \cdots, m$ from the RPLR with $\pi_j, j = 1, \cdots, m$. By Theorem 1, $\hat{f}_j$ estimates whether the class probability is greater than π or not. Therefore, if we make the partition fine enough, then we can achieve probability estimation with the desired level of accuracy. To be more specific, we define $\pi^* = \text{argmax}_{\pi_j}\{\hat{f}_j > 0\}$ and $\pi_* = \text{argmin}_{\pi_j}\{\hat{f}_j < 0\}$, then $\hat{p}$ is obtained by $\frac{1}{2}(\pi^* + \pi_*)$.

This method is not restricted by the truncation location, that is, we can use this method for any $t > \log 2$, corresponding to $s \leqslant 0$. The larger m we use, the finer estimate we can get. However, larger m's require higher computational costs.

2.5 Computational Algorithms

Since the loss function g_s is not convex, the RPLR requires non-convex minimization. Note that g_s can be written as the difference of two convex functions as $g_s(u) = l(u) - l_s(u)$ as shown in the middle panel of Figure 2.1. With this decomposition, we can solve the non-convex minimization via the d.c. algorithm (An and Tao, 1997; Horst and Thoai, 1999; Liu et al., 2005). The d.c. algorithm solves the problem by sequential convex minimization. For each iteration, l_s is replaced by its linear approximation using the current solution. Then the problem becomes convex minimization. We iterate this until the objective function converges. In this section, we discuss the d.c. algorithm for the RPLR.

In the literature, Fan and Li (2001) introduced Local Quadratic Approximation (LQA) to solve penalized likelihood optimization problems. Hunter and Li (2005) studied convergence of LQA as an instance of minorize-maximize or majorize-minimize (MM) algorithm. Considering a linear approximation of l_s as the affine minorization, the d.c. algorithm for RPLR is also a special case of the MM algorithm. Since the objective function in (2.15) is positive, our d.c. algorithm converges to an ϵ-local minimizer in finite iterations (An and Tao, 1997; Liu et al., 2005).

In linear learning with $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$, (2.15) can be reduced to

$$\min_{b,\boldsymbol{w}} \sum_{i=1}^{n} h_\pi(y_i) g_s(y_i f(\boldsymbol{x}_i)) + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2 . \tag{2.16}$$

Using the fact that $g_s(u) = l(u) - l_s(u)$ with $l(u) = \log(1 + e^{-u})$ and $l_s(u) = [\log(1 + e^{-u}) - \log(1 + e^{-s})]_+$, (2.16) can be written as

$$\min_{\Theta} Q(\Theta) = \min_{\Theta} Q_{vex}(\Theta) + Q_{cav}(\Theta), \tag{2.17}$$

where $\Theta = (b, \boldsymbol{w})$, $Q_{vex}(\Theta)^s = \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2 + \sum_{i=1}^n h(y_i) l(y_i f(\boldsymbol{x}_i))$ and $Q_{cav}(\Theta)^s = -\sum_{i=1}^n h(y_i) l_s(y_i f(\boldsymbol{x}_i))$. Then, at the $(m+1)$-th iteration, the d.c. algorithm minimizes

$$\begin{aligned} &Q_{vex}(\Theta_m)^s + \langle \tfrac{\partial}{\partial \boldsymbol{w}} Q^s_{cav}(\Theta_m), \boldsymbol{w}\rangle + b\tfrac{\partial}{\partial b} Q^s_{cav}(\Theta_m) \\ &= \tfrac{\lambda}{2}\|\boldsymbol{w}\|_2^2 + \textstyle\sum_{i=1}^n h(y_i)\log(1 + e^{-y_i f(\boldsymbol{x}_i)}) + \sum_{i=1}^n h(y_i)\beta_i \frac{e^{-y_i f_m(\boldsymbol{x}_i)}}{1+e^{-y_i f_m(\boldsymbol{x}_i)}}(\boldsymbol{w}^T\boldsymbol{x}_i + b), \end{aligned} \tag{2.18}$$

where $f_m(\boldsymbol{x}) = \boldsymbol{w}_m^T\boldsymbol{x}+b_m$ and $\beta_i = 1$ if $y_i = 1$ and $f(\boldsymbol{x}_i) < s$, -1 if $y_i = -1$ and $f(\boldsymbol{x}_i) > -s$, and 0 otherwise. Problem (2.18) can then be solved using nonlinear convex minimization techniques.

The algorithm can be extended to nonlinear learning directly. Specifically, for kernel learning, (2.15) becomes

$$\min_{b,\boldsymbol{v}} \sum_{i=1}^{n} h_\pi(y_i) g_s(y_i f(\boldsymbol{x}_i)) + \frac{\lambda}{2} \|f\|^2_{\mathcal{H}_K} \tag{2.19}$$

where $f(\boldsymbol{x}) = \sum_{i=1}^n v_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b$ and $\boldsymbol{v} = (v_1, \cdots, v_n)$. Notice that $\sum_{i=1}^n v_i K(\boldsymbol{x}_i, \boldsymbol{x}) \in H_K$ and $\|f\|^2_{\mathcal{H}_K} = \langle \boldsymbol{v}, K\boldsymbol{v}\rangle$. Using $\Theta = (b, \boldsymbol{v})$ in (2.17) leads to a similar algorithm for the nonlinear kernel learning case.

2.6 Tuning Parameter Selection

The tuning parameter λ in (2.16) and (2.19) plays an important role for the RPLR. In this section, we explore various ways to tune λ. We use penalty term which measures smoothness of the model to avoid overfitting the data, and the tuning parameter λ decides how smooth our model will be. Thus, the choice of λ has a big impact on the resulting model.

There are numerous ways proposed to tune λ in the penalized likelihood literature and we employ some of those here for the RPLR. Some well known ones include the cross validation, AIC, and BIC. Among them, cross validation is probably one of the most commonly used method. Since cross validation requires intensive computation, Generalized Approximate Cross Validation (GACV) can be a good approximation. In this section, we explore how to generalize these existing methods such as AIC, BIC, and GACV to the RPLR problem.

The term AIC and BIC are defined as $\mathit{deviance} + \mathit{cp} \times \mathit{df}$, with $\mathit{cp} = 2$ and $\mathit{cp} = \log n$, respectively. *deviance* measures goodness of fit of the model, and *df* measures amount of overfitting. More specifically, $\mathit{deviance} = -2\log \mathit{likelihood}$, hence better fitting on the training data gives the smaller deviance. By minimizing the sum of *deviance* and $\mathit{cp} \times \mathit{df}$, we can balance the tradeoff between goodness of fit and generalization.

For a linear smoother in the form of $\hat{\boldsymbol{y}} = S\boldsymbol{y}$, a popular definition of *df* is $\mathrm{tr}(S)$ (Hastie and Tibshirani, 1990). However, this definition is not applicable for the RPLR problem directly since the RPLR is not such a linear smoother. Park and Hastie (2007) generalized definition of *df* to *change in deviance* of null data, i.e. *deviance* of null model $-$ *deviance* of current model.

The idea is that the difference of *deviance* between null model and current model would be due to overfitting if we use pure noise as data. Hence we can use *change in deviance* to measure the amount of overfitting. In Park and Hastie (2007), they simulated many samples of null data to estimate df. Their approach can be used for the RPLR problem as well in the same manner. However, this method can be computationally expensive in practice.

Xiang and Wahba (1996) proposed GACV, which estimates comparative Kullback-Leibler distance between the true linear predictor $f(\boldsymbol{x})$ and the estimated one for a particular λ. It starts with a leaving-out-one version, then uses Taylor expansion to get an estimate. This idea can be generalized here to get GACV of the RPLR. The details are as follows.

Let $f_\lambda(\boldsymbol{x})$ be the solution of the RPLR for a particular value of λ. The Kullback-Leibler distance $KL(f, f_\lambda)$ is

$$KL(f, f_\lambda) = \frac{1}{n}\sum_{i=1}^{n} E \log \frac{\tilde{\mathcal{L}}(y_i, f(\boldsymbol{x}_i))}{\tilde{\mathcal{L}}(y_i, f_\lambda(\boldsymbol{x}_i))},$$

where $\tilde{\mathcal{L}}(y_i, f(\boldsymbol{x}_i)) = P(\boldsymbol{x}_i)^{\frac{1+y_i}{2}}(1 - P(\boldsymbol{x}_i))^{\frac{1-y_i}{2}}$ for the PLR and $\tilde{\mathcal{L}}(y_i, f(\boldsymbol{x}_i)) = Q^+(\boldsymbol{x}_i)^{\frac{1+y_i}{2}}(1 - Q^-(\boldsymbol{x}_i))^{\frac{1-y_i}{2}}$ for the RPLR. Since the true $f(\boldsymbol{x})$ is unknown and does not depend on λ, we define the Comparative KL loss,

$$CKL(\lambda) = KL(f, f_\lambda) - \frac{1}{n}\sum_{i=1}^{n} E \log \tilde{\mathcal{L}}(y_i, f(\boldsymbol{x}_i))$$

to compare models with different λ. It can be shown that $CKL(\lambda) = \frac{1}{n}\sum_{i=1}^{n} E[-z_i f_\lambda(\boldsymbol{x}_i) + \log(1 + e^{f_\lambda(\boldsymbol{x}_i)})]$ for the PLR, and $CKL(\lambda) = \frac{1}{n}\sum_{i=1}^{n} E[\min\{t, -z_i f_\lambda(\boldsymbol{x}_i) + \log(1 + e^{f_\lambda(\boldsymbol{x}_i)})\}]$ for the RPLR, with $z_i = \frac{1}{2}(1 + y_i)$. Then the remaining issue is how to estimate the CKL.

First, let $f_\lambda^{(-i)}(\cdot)$ is the solution of the RPLR with the i-th data point omitted. Adopting the leaving-out-cone cross validation function $CV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}[-z_i f_\lambda^{(-i)}(\boldsymbol{x}_i) + \log(1 + e^{f_\lambda(\boldsymbol{x}_i)})]$ for data from general exponential family in Xiang and Wahba (1996), we define $CV(\lambda)$ for the RPLR,

$$CV(\lambda) = \frac{1}{n}\sum_{i=1}^{n} \min\left\{t, -z_i f_\lambda^{(-i)}(\boldsymbol{x}_i) + \log(1 + e^{f_\lambda(\boldsymbol{x}_i)})\right\}. \tag{2.20}$$

Since it is computationally expensive to calculate $f_\lambda^{(-i)}(\boldsymbol{x}_i)$, we approximate $CV(\lambda)$ using formulae introduced in Xiang and Wahba (1996) and Liu (1995). Specifically, from (2.20), we

have

$$\begin{aligned} CV(\lambda) &= \tfrac{1}{n}\sum_{i=1}^{n} \min\left\{t, -z_i f_\lambda(\boldsymbol{x}_i) + \log(1+e^{f_\lambda(\boldsymbol{x}_i)}) + z_i(f_\lambda(\boldsymbol{x}_i) - f_\lambda^{(-i)}(\boldsymbol{x}_i))\right\} \\ &= \tfrac{1}{n}\sum_{i=1}^{n} \min\{t, a_i + b_i\}, \end{aligned} \tag{2.21}$$

where $a_i = -z_i f_\lambda(\boldsymbol{x}_i) + \log(1+e^{f_\lambda(\boldsymbol{x}_i)})$ and $b_i = z_i(f_\lambda(\boldsymbol{x}_i) - f_\lambda^{(-i)}(\boldsymbol{x}_i))$. Define

$$d_i = \begin{cases} 1 & \text{if } t > \max(a_i+b_i, a_i) \\ 0 & \text{if } t < \min(a_i+b_i, a_i) \\ \frac{t-(a_i+b_i)}{-b_i} & \text{if } a_i + b_i < t < a_i \\ \frac{t-a_i}{b_i} & \text{if } a_i < t < a_i + b_i. \end{cases} \tag{2.22}$$

Note that $0 < d_i < 1$. Now (2.21) becomes

$$\begin{aligned} CV(\lambda) &= \tfrac{1}{n}\sum_{i=1}^{n} \left[\min\left\{t, -z_i f_\lambda(\boldsymbol{x}_i) + \log(1+e^{f_\lambda(\boldsymbol{x}_i)})\right\} + d_i z_i (f_\lambda(\boldsymbol{x}_i) - f_\lambda^{(-i)}(\boldsymbol{x}_i))\right] \\ &= \tfrac{1}{n}\sum_{i=1}^{n} \min\left\{t, -z_i f_\lambda(\boldsymbol{x}_i) + \log(1+e^{f_\lambda(\boldsymbol{x}_i)})\right\} + \tfrac{1}{n}\sum_{i=1}^{n} d_i z_i \frac{f_\lambda(\boldsymbol{x}_i) - f_\lambda^{(-i)}(\boldsymbol{x}_i)}{z_i - P_\lambda^{(-i)}(\boldsymbol{x}_i)} \frac{z_i - P_\lambda(\boldsymbol{x}_i)}{1 - \frac{P_\lambda(\boldsymbol{x}_i) - P_\lambda^{(-i)}(\boldsymbol{x}_i)}{z_i - P_\lambda^{(-i)}(\boldsymbol{x}_i)}} \end{aligned} \tag{2.23}$$

where $P_\lambda(\boldsymbol{x}_i) = 1/(1+e^{-f_\lambda(\boldsymbol{x}_i)})$ and $P_\lambda^{(-i)}(\boldsymbol{x}_i) = 1/(1+e^{-f_\lambda^{(-i)}(\boldsymbol{x}_i)})$. Let $b(f_\lambda(\boldsymbol{x}_i)) = \log(1 + e^{f_\lambda(\boldsymbol{x}_i)})$. Since $b'(f_\lambda(\boldsymbol{x}_i)) = P_\lambda(\boldsymbol{x}_i)$ and $b''(f_\lambda(\boldsymbol{x}_i)) = P_\lambda(\boldsymbol{x}_i)(1 - P_\lambda(\boldsymbol{x}_i))$,

$$\frac{P_\lambda(\boldsymbol{x}_i) - P_\lambda^{(-i)}(\boldsymbol{x}_i)}{z_i - P_\lambda^{(-i)}(\boldsymbol{x}_i)} = \frac{b'(f_\lambda(\boldsymbol{x}_i)) - b'(f_\lambda^{(-i)}(\boldsymbol{x}_i))}{z_i - P_\lambda^{(-i)}(\boldsymbol{x}_i)} \approx b''(f_\lambda(\boldsymbol{x}_i)) \frac{f_\lambda(\boldsymbol{x}_i) - f_\lambda^{(-i)}(\boldsymbol{x}_i)}{z_i - P_\lambda^{(-i)}(\boldsymbol{x}_i)}, \tag{2.24}$$

and (2.23) becomes

$$CV(\lambda) = \frac{1}{n}\sum_{i=1}^{n} \min\left\{t, -z_i f_\lambda(\boldsymbol{x}_i) + \log(1+e^{f_\lambda(\boldsymbol{x}_i)})\right\} + \frac{1}{n}\sum_{i=1}^{n} d_i \frac{z_i(z_i - P_\lambda(\boldsymbol{x}_i))}{\frac{z_i - P_\lambda^{(-i)}(\boldsymbol{x}_i)}{f_\lambda(\boldsymbol{x}_i) - f_\lambda^{(-i)}(\boldsymbol{x}_i)} - P_\lambda(\boldsymbol{x}_i)(1 - P_\lambda(\boldsymbol{x}_i))}. \tag{2.25}$$

Now what is left is the calculation of $\frac{z_i - P_\lambda^{(-i)}(\boldsymbol{x}_i)}{f_\lambda(\boldsymbol{x}_i) - f_\lambda^{(-i)}(\boldsymbol{x}_i)}$. We modify the leaving-out-one lemma of Xiang and Wahba (1996), which is a generalized version of the leaving-out-one lemma of Craven and Wahba (1979).

Lemma 1. *(Leaving-out-one lemma) Let* $-\tilde{l}(z_i, f(\boldsymbol{x}_i)) = \min\{t, -z_i f(\boldsymbol{x}_i) + \log(1+e^{f(\boldsymbol{x}_i)})\}$ *and* $I_\lambda(f, \boldsymbol{z}) = -\sum_{i=1}^{n} \tilde{l}(z_i, f(\boldsymbol{x}_i)) + n\lambda J(f)$. *Suppose* $f^*(i, z^*, \cdot)$ *is the minimizer in* $\mathcal{F}$ *of* $I_\lambda(f, \boldsymbol{z}^*)$,

where $\boldsymbol{z}^* = (z_1, \cdots, z_{i-1}, z^*, z_{i+1}, \cdots, z_n)$. *Then,*

$$f^*(i, P_\lambda^{(-i)}(\boldsymbol{x}_i), \cdot) = f_\lambda^{(-i)}(\cdot),$$

where $f_\lambda^{(-i)}(\cdot)$ *is the minimizer of* $-\sum_{j\neq i}\tilde{l}(z_j, f(\boldsymbol{x}_j))+n\lambda J(f)$, *and* $P_\lambda^{(-i)}(\boldsymbol{x}) = 1/(1+e^{-f_\lambda^{(-i)}(\boldsymbol{x})})$.

Now let $\boldsymbol{f}_\lambda = (f_\lambda(\boldsymbol{x}_1), \cdots, f_\lambda(\boldsymbol{x}_n))^T$, $\boldsymbol{f}_\lambda^{(-i)} = (f_\lambda^{(-i)}(\boldsymbol{x}_1), \cdots, f_\lambda^{(-i)}(\boldsymbol{x}_n))^T$, $\boldsymbol{z} = (z_1, \cdots, z_n)^T$, and $\boldsymbol{z}^{(-i)} = (z_1, \cdots, z_{i-1}, P_\lambda^{(-i)}(\boldsymbol{x}_i), z_{i+1}, \cdots, z_n)^T$. By the definition of f_λ, $(f_\lambda, \boldsymbol{z})$ is a local minimizer of $I_\lambda(f, \boldsymbol{z}^*)$. Also, $(f_\lambda^{(-i)}, \boldsymbol{z}^{(-i)})$ is a local minimizer of $I_\lambda(f, \boldsymbol{z}^*)$ by Lemma 1. Therefore, $\frac{\partial I_\lambda(f, \boldsymbol{z}^*)}{\partial f}(f_\lambda, \boldsymbol{z}) = 0$ and $\frac{\partial I_\lambda(f, \boldsymbol{z}^*)}{\partial f}(f_\lambda^{(-i)}, \boldsymbol{z}^{(-i)}) = 0$. Writing $J(f) = \boldsymbol{f}^T\Sigma\boldsymbol{f}$ gives $I_\lambda = \min\{t, -z_i f(\boldsymbol{x}_i) + \log(1 + e^{f(\boldsymbol{x}_i)})\} + n\lambda\boldsymbol{f}^T\Sigma\boldsymbol{f}$(See Section 3.1. of Xiang and Wahba (1996) for computation of Σ.). Since I_λ is not differentiable, we approximate it with a differentiable function

$$I_\lambda^* = \sum_{i=1}^{n} g^*(f_i, z_i, \boldsymbol{x}_i) + n\lambda\boldsymbol{f}^T\Sigma\boldsymbol{f}, \tag{2.26}$$

with

$$g^*(f, z, \boldsymbol{x}) = \begin{cases} t & \text{if } yf < -\log(e^t - 1) - \epsilon \\ g^{**}(f, z, \boldsymbol{x}) & \text{if } -\log(e^t - 1) - \epsilon \leqslant yf \leqslant -\log(e^t - 1) + \delta \\ -zf + \log(1 + e^f) & \text{if } yf > -\log(e^t - 1) + \delta(\epsilon) \end{cases} \tag{2.27}$$

where g^{**} is a quadratic function of f which makes g^* differentiable in f. Note that $I_\lambda^* \longrightarrow I_\lambda$ as $\epsilon \to 0$. Let σ_{ij} be the ij-th element of Σ. Then,

$$\frac{\partial I_\lambda^*}{\partial f(\boldsymbol{x}_i)} \xrightarrow{\epsilon\to 0} \begin{cases} -z_i + 1/(1 + e^{-f(\boldsymbol{x}_i)}) + n\lambda\sum_j \sigma_{ij} f(\boldsymbol{x}_i) & \text{if } z_i f(\boldsymbol{x}_i) \geqslant -\log(e^t - 1) \\ n\lambda\sum_j \sigma_{ij} f(\boldsymbol{x}_i) & \text{otherwise,} \end{cases} \tag{2.28}$$

and

$$\frac{\partial^2 I_\lambda^*}{\partial f(\boldsymbol{x}_i)\partial f(\boldsymbol{x}_j)} \xrightarrow{\epsilon\to 0} \begin{cases} n\lambda\sum_j \sigma_{ii} + I_{\{z_i f(\boldsymbol{x}_i) \geqslant -\log(e^t-1)\}} \frac{e^{f(\boldsymbol{x}_i)}}{(1+e^{f(\boldsymbol{x}_i)})^2} & \text{if } i = j \\ n\lambda\sum_j \sigma_{ij} & \text{if } i \neq j. \end{cases} \tag{2.29}$$

Therefore, defining $W(\boldsymbol{f}) = \text{diag}(I_{\{z_1 f(\boldsymbol{x}_1) \geqslant -\log(e^t-1)\}} \frac{e^{f(\boldsymbol{x}_1)}}{(1+e^{f(\boldsymbol{x}_1)})^2}, \cdots, I_{\{z_n f(\boldsymbol{x}_n) \geqslant -\log(e^t-1)\}} \frac{e^{f(\boldsymbol{x}_n)}}{(1+e^{f(\boldsymbol{x}_n)})^2})$,

we have $\frac{\partial^2 I_\lambda^*}{\partial \boldsymbol{f} \partial \boldsymbol{f}^T} \xrightarrow{\epsilon \to 0} W + n\lambda\Sigma$, and $\frac{\partial^2 I_\lambda^*}{\partial \boldsymbol{z} \partial \boldsymbol{f}^T} \xrightarrow{\epsilon \to 0} -I$. Using Taylor expansion,

$$\begin{aligned} 0 &= \tfrac{\partial I_\lambda^*}{\partial \boldsymbol{f}}(\boldsymbol{f}_\lambda^{(-i)}, \boldsymbol{z}^{(-i)}) \\ &= \tfrac{\partial I_\lambda^*}{\partial \boldsymbol{f}}(\boldsymbol{f}_\lambda, \boldsymbol{z}) + \tfrac{\partial^2 I_\lambda^*}{\partial \boldsymbol{f} \partial \boldsymbol{f}^T}(\boldsymbol{f}_\lambda^{**}, \boldsymbol{z}^{**})(\boldsymbol{f}_\lambda^{(-i)} - \boldsymbol{f}_\lambda) + \tfrac{\partial^2 I_\lambda^*}{\partial \boldsymbol{z} \partial \boldsymbol{f}^T}(\boldsymbol{f}_\lambda^{**}, \boldsymbol{z}^{**})(\boldsymbol{z} - \boldsymbol{z}^{(-i)}) \\ &\xrightarrow{\epsilon \to 0} 0 + \{W(\boldsymbol{f}^{**}) + n\lambda\Sigma\}(\boldsymbol{f}_\lambda^{(-i)} - \boldsymbol{f}_\lambda) - (\boldsymbol{z} - \boldsymbol{z}^{(-i)}), \end{aligned} \tag{2.30}$$

where $(\boldsymbol{f}_\lambda^{**}, \boldsymbol{z}^{**})$ is a point somewhere between $(\boldsymbol{f}_\lambda, \boldsymbol{z})$ and $(\boldsymbol{f}_\lambda^{(-i)}, \boldsymbol{z}^{(-i)})$. Approximating $W(\boldsymbol{f}_\lambda^{**})$ by $W(\boldsymbol{f}_\lambda)$ and letting $\epsilon \to 0$ gives $\boldsymbol{f}_\lambda - \boldsymbol{f}_\lambda^{(-i)} = \{W(\boldsymbol{f}_\lambda^{**}) + n\lambda\Sigma\}^{-1}(\boldsymbol{z} - \boldsymbol{z}^{(-i)})$, i.e.

$$\begin{pmatrix} f_\lambda(\boldsymbol{x}_1) - f_\lambda^{(-i)}(\boldsymbol{x}_1) \\ \vdots \\ f_\lambda(\boldsymbol{x}_i) - f_\lambda^{(-i)}(\boldsymbol{x}_i) \\ \vdots \\ f_\lambda(\boldsymbol{x}_n) - f_\lambda^{(-i)}(\boldsymbol{x}_n) \end{pmatrix} \simeq \{W(\boldsymbol{f}_\lambda^{**}) + n\lambda\Sigma\}^{-1} \begin{pmatrix} 0 \\ \vdots \\ z_i - P_\lambda^{(-i)}(\boldsymbol{x}_i) \\ \vdots \\ 0 \end{pmatrix}. \tag{2.31}$$

Let $H = \{W(\boldsymbol{f}_\lambda) + n\lambda\Sigma\}^{-1}$ and h_{ii} be the i-th diagonal entry of H. Then (2.31) implies

$$\frac{f_\lambda(\boldsymbol{x}_i) - f_\lambda^{(-i)}(\boldsymbol{x}_i)}{z_i - P_\lambda^{(-i)}(\boldsymbol{x}_i)} \simeq h_{ii} \tag{2.32}$$

Using (2.32), (2.25) becomes

$$\frac{1}{n}\sum_{i=1}^{n} \min\left\{t, -z_i f_\lambda(\boldsymbol{x}_i) + \log(1 + e^{f_\lambda(\boldsymbol{x}_i)})\right\} + \frac{1}{n}\sum_{i=1}^{n} d_i \frac{h_{ii} z_i (z_i - P_\lambda(\boldsymbol{x}_i))}{1 - h_{ii} P_\lambda(\boldsymbol{x}_i)(1 - P_\lambda(\boldsymbol{x}_i))}. \tag{2.33}$$

Replacing h_{ii} by $\mathrm{tr}(H)/n$ and replacing $h_{ii} P_\lambda(\boldsymbol{x}_i)(1 - P_\lambda(\boldsymbol{x}_i))$ by $\mathrm{tr}(W^{*1/2} H W^{*1/2})/\mathrm{n}$ with $W^* = \mathrm{diag}(\frac{e^{f(\boldsymbol{x}_1)}}{(1+e^{f(\boldsymbol{x}_1)})^2}, \cdots, \frac{e^{f(\boldsymbol{x}_n)}}{(1+e^{f(\boldsymbol{x}_n)})^2})$, we define

$$GACV(\lambda) = \frac{1}{n}\sum_{i=1}^{n} \min\left\{t, -z_i f_\lambda(\boldsymbol{x}_i) + \log(1 + e^{f_\lambda(\boldsymbol{x}_i)})\right\} + \frac{\mathrm{tr}(H)}{n}\sum_{i=1}^{n} d_i \frac{h_{ii} z_i (z_i - P_\lambda(\boldsymbol{x}_i))}{n - \mathrm{tr}(W^{*1/2} H W^{*1/2})}, \tag{2.34}$$

where $H = \{W(\boldsymbol{f}_\lambda) + n\lambda\Sigma\}^{-1}$ with Σ such that $\boldsymbol{f}^T \Sigma \boldsymbol{f}$, h_{ii} is the i-th diagonal entry of H,

$P_\lambda(\boldsymbol{x}) = 1/(1+e^{-f_\lambda(\boldsymbol{x})})$, and

$$d_i = \begin{cases} 1 & \text{if } t > \max(a_i + b_i, a_i) \\ 0 & \text{if } t < \min(a_i + b_i, a_i) \\ \frac{t-(a_i+b_i)}{-b_i} & \text{if } a_i + b_i < t < a_i \\ \frac{t-a_i}{b_i} & \text{if } a_i < t < a_i + b_i. \end{cases}$$

with $a_i = -z_i f_\lambda(\boldsymbol{x}_i) + \log(1+e^{f_\lambda(\boldsymbol{x}_i)})$ and $b_i = z_i(f_\lambda(\boldsymbol{x}_i) - f_\lambda^{(-i)}(\boldsymbol{x}_i))$ where $f_\lambda^{(-i)}(\cdot)$ is the solution of the RPLR with the i-th data point omitted. Using the fact that $0 < d_i < 1$, we can bound $GACV(\lambda)$. We use the average of the upper and lower bound of $GACV$, that is, we define Estimated GACV (EGACV)

$$EGACV(\lambda) = \frac{1}{n}\sum_{i=1}^{n} \min\left\{t, -z_i f_\lambda(\boldsymbol{x}_i) + \log(1+e^{f_\lambda(\boldsymbol{x}_i)})\right\} + \frac{\mathrm{tr}(H)}{2n}\sum_{i=1}^{n} \frac{h_{ii} z_i (z_i - P_\lambda(\boldsymbol{x}_i))}{n - \mathrm{tr}(W^{*1/2} H W^{*1/2})}.$$

We use simulated data to illustrate the performance of $EGACV(\lambda)$. The training set consists of 50 data points sampled from the uniform distribution over a unit disk $\{(x_1, x_2) : x_1^2 + x_2^2 \leqslant 1\}$ and labeled as $y = 1$ if $x_1 \geqslant x_2$, $y = -1$ otherwise. The testing set has 10^5 data points which are sampled and labeled in the same manner as the training set. Using these datasets, we build a model using the RPLR with $t = 2\log 2$ based on the training set and calculate $CKL(\lambda)$ of the testing set for each λ such that $\log_{10}\lambda \in \{-3.0, -2.9, \cdots, 2.0\}$. Then we calculate $EGACV(\lambda)$ using the training set only and plot it with $CKL(\lambda)$ to see how close they are. We repeat this 100 times with a different training set each time and take average of $EGACV(\lambda)$ and $CKL(\lambda)$ and plot them. The left panel of Figure 2.4 illustrates typical curves of $EGACV(\lambda)$ and $CKL(\lambda)$ from one example, and the average curves of the 100 repetitions are plotted in the right panel. The solid line shows $CKL(\lambda)$, the dashed line shows $EGACV(\lambda)$, and the dotted lines show the upper and lower bounds of $GACV(\lambda)$. As shown in the Figure 2.4, $EGACV(\lambda)$ reflects the variation of $CKL(\lambda)$ quite well, thus $EGACV(\lambda)$ can be a useful tool for tuning λ.

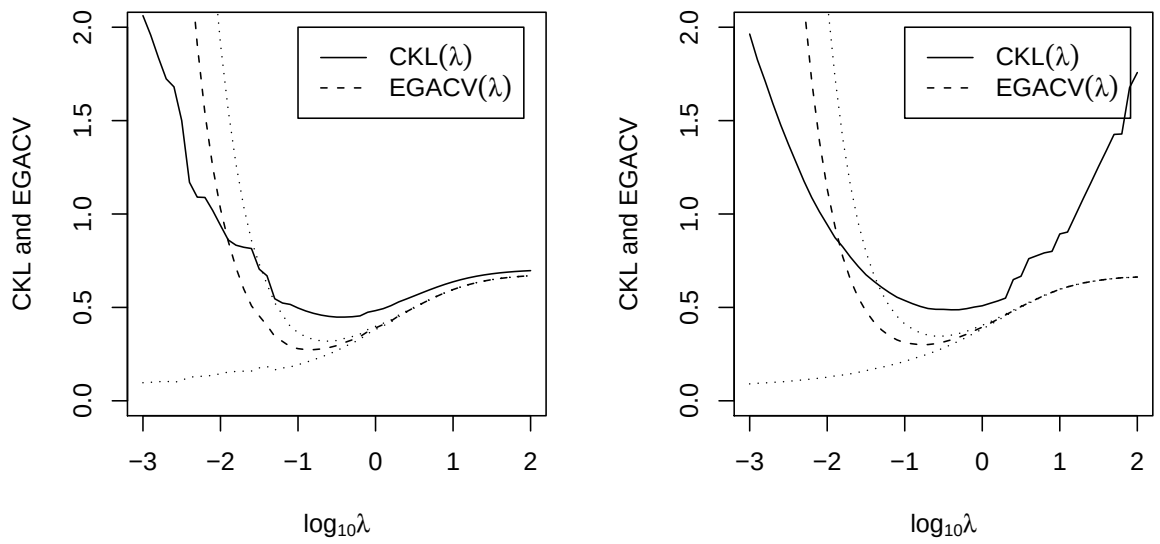

Figure 2.4: Left: An illustration plot of $CKL(\lambda)$ and $EGACV(\lambda)$ from the example in Section 2.6; Right: Average curves of $CKL(\lambda)$ and $EGACV(\lambda)$ based on 100 replications.

2.7 Numerical Examples

In this section, we examine the performance of the RPLR and compare it with some other classification methods. On two simulated examples, we compute the SVM, RSVM, PLR, and RPLR to compare their classification errors as well as accuracy of class probability estimation. On two real data examples, we compare the performance of class probability estimation of the PLR and RPLR. Note that the RSVM is a modified version of the SVM, which uses the truncated hinge loss instead of the standard hinge loss (Wu and Liu, 2007).

2.7.1 Simulation

In the two simulated examples, data are generated with the sample sizes of training, tuning and testing sets 100, 100, and 10^6, respectively. The training data sets are used to build classifiers, and λ is chosen by two different ways: by a grid search based on the tuning sets, and by a grid search based on the GACV calculated from the training set. The testing errors and probability estimation errors are evaluated using the testing sets.

Example 2.7.1.1 The data are generated as follows. First, (x_1, x_2) is sampled from the uniform distribution over a unit disk $\{(x_1, x_2) : x_1^2 + x_2^2 \leqslant 1\}$. Then, set $y = 1$ if $x_1 \geqslant x_2$, $y = -1$ otherwise. To demonstrate robustness of the RPLR, we randomly select v percent of the observations and change their class labels to the other classes, where $v = 0$, 5, 10 and 20. For each value of v, we repeat the classification procedure 100 times to capture variation of the results. Since the true boundary is linear, we focus on linear learning in this example. For the RSVM, we consider two different truncation locations $s = -1$ and 0, corresponding to $t = 2$, and 1. For the RPLR, we use $s = 0$, $-\log 3$, and $-\log 5$ which correspond to $t = \log 2$, $2\log 2$, and $3\log 2$, respectively. We also report misclassification rate of the RPLR when we tune s along with λ, as well as results of another version of logistic regression proposed by Croux and Haesbroeck (2003) for comparison. For class probability estimation, We apply scheme 3 to each t, but scheme 1 and scheme 2 are used only for $t = 3\log 2$ because they are valid only if $t > 2\log 2$. To evaluate accuracy of probability estimation, we use $\frac{1}{n}\sum_{i=1}^{n} |\hat{P}(\boldsymbol{x}_i) - P(\boldsymbol{x}_i)|$ to measure the probability estimation error.

Results are summarized in Tables 2.1 and 2.2. As shown in Table 2.1, the RPLR outperforms other classifiers in terms of classification accuracy. With no contamination, the performances of the RPLR and the PLR are very similar. As we increase the percent of contaimnation, the RPLR performs better than the PLR because the truncated loss is more robust against outliers. Similar conclusions can be drawn for the SVM and the RSVM, because the RPLR and the RSVM are the truncated versions of the original PLR and the SVM, respectively. The results of the RPLR using separate tuning sets are better than that of the RSVM. This may due to the difference of their loss functions as discussed in Section 2.4.1.

The location of truncation is an important issue. If the loss function is not truncated, it can be sensitive to outliers. If the loss function is truncated too much, we may underuse the information of those data points close to the decision boundary. The performance of the RPLR with $t = \log 2$ corresponding to the most truncation, is indeed suboptimal as shown in Tables 2.1 and 2.2. The RPLR with $t = 3\log 2$ works the best for the cases $v = 0$ and 5, but as the proportion of contamination grows, performance of the RPLR with $t = 2\log 2$ becomes the best. This is reasonable because more truncation helps for data with more outliers. In general, we recommend to use $t = 2\log 2$ for the truncation location for binary problems. This choice also

has good theoretical justification as mentioned in Section 2.4.2 in terms of Fisher consistency.

Regarding to the choice of λ, the one chosen based on the tuning set performed better than the one by the GACV. This may not be surprising because the first approach uses information from both the training set and the tuning set to choose λ, while the GACV approach uses the training set only. Hence a direct comparison may not be fair considering the difference in the amount of information used between the two approaches. Nevertheless we can see that the GACV approach works fairly well in this example.

As to the issue of class probability estimation, the RPLR with $t = 3\log 2$ works the best for non-contaminated data, but $t = 2\log 2$ becomes better as the rate of contamination increases. This agrees with the results of classification error. In general, better classification performance can be translated into better class probability estimation. Thus, the RPLR yields more accurate class probability estimation than that of the PLR. Among three different schemes, scheme 3 seems to perform the best overall.

To visualize the classification boundaries, we select a typical dataset and plot the corresponding boundaries yielded by the PLR and the RPLR on the left panel of Figure 2.5. Clearly, the RPLR is much less sensitive to outliers and deliver more accurate classification boundary than that of the PLR.

Example 2.7.1.2 We generate (x_1, x_2) uniformly from the unit disk $\{(x_1, x_2) : x_1^2 + x_2^2 \leqslant 1\}$ with y being 1 if $(x_1 - x_2)(x_1 + x_2) < 0$, and -1 otherwise. Then we flip the class labels using the same strategy as in Example 2.7.1.1. Linear learning does not work here due to its generation. We use nonlinear learning with Gaussian kernel $K(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp(-\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2/(2\sigma^2))$. We tune σ among the first quartile, the median, and the third quartile of the between-class pairwise Euclidean distances of training inputs (Wu and Liu, 2007). We use the same truncation location, class probability estimation schemes, and measure of probability estimation error as in Example 2.7.1.1. Results are reported in Table 2.3 and Table 2.4. Similarly, the RPLR with $t = 2\log 2$ works the best overall. When outliers exist in the data, truncation indeed improves both classification accuracy as well as class probability estimation. Similar to Example 2.7.1.1, we plot the results of one typical example on the right panel of Figure 2.5. Again, the RPLR is more robust and consequently its classification boundary is closer to the Bayes decision boundary.

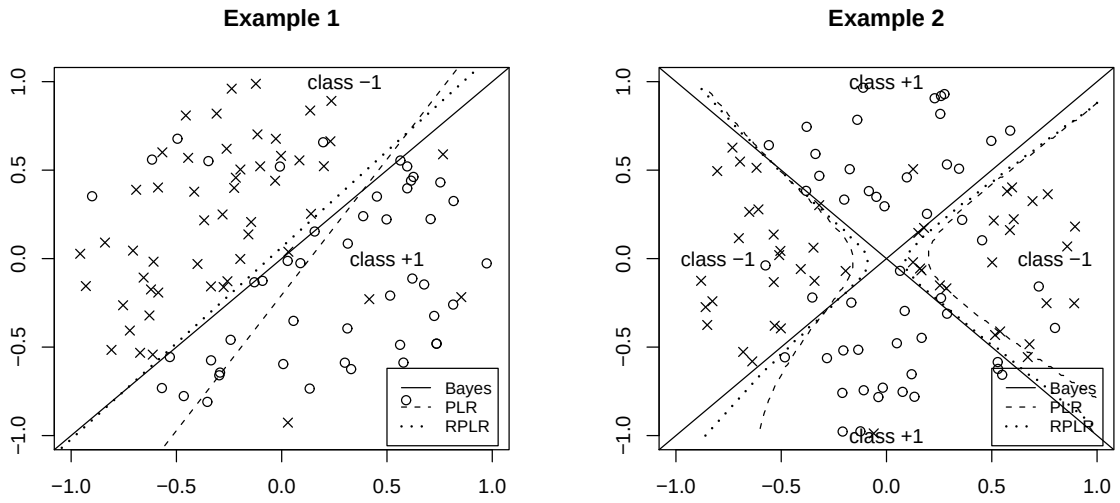

Figure 2.5: Plot of typical training sets for Example 2.7.1.1 (the left panel) and Example 2.7.2.2 (the right panel) as well as the corresponding decision boundaries.

2.7.2 Real Data

2.7.2.1 Leukaemia Data Here, we apply the PLR and the RPLR to the Leukaemia dataset described in Golub et al. (1999). This dataset is publicly available at http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi. It contains 72 samples with 7129 gene expression values. The goal is to classify the patients into two types of leukaemia: acute myeloid leukaemia (AML) and acute lymphoblastic leukaemia (ALL). Since the number of genes is much higher than the sample size, we performed prescreening to choose a subset of genes. In particular, we used the ratios of between-groups to within-groups sum of squares of the genes to sort them and chose the top 40 genes. Similar procedure was done in Dudoit et al. (2002).

This dataset includes a training set with 38 instances and a testing set with 34 instances. Heatmaps in Figure 2.6 are drawn for good visualization of the datasets. From the heatmap of the testing set, we can identify some observations are difficult to classify. Indeed, there are two subjects that the PLR and the RPLR fail to classify to the correct classes. The training set is used for model building, then performance of the model is evaluated on the testing set. More specifically, the tuning parameter λ is chosen by 5-fold cross validation on the training set. We also used EGACV and it gives very similar results. Using the RPLR coefficients estimated

Table 2.1: Testing errors of the simulated linear example (Example 2.7.1.1)

Method		$v = 0$	$v = 5$	$v = 10$	$v = 20$
SVM		0.0121(0.0077)	0.0728(0.0145)	0.1319(0.0207)	0.2326(0.0205)
RSVM	$t = 2$	0.0122(0.0085)	0.0642(0.0096)	0.1182(0.0136)	0.2233(0.0173)
	$t = 1$	0.0149(0.0112)	0.0697(0.0138)	0.1205(0.0141)	0.2231(0.0164)
PLR		0.0090(0.0064)	0.0726(0.0143)	0.1348(0.0210)	0.2371(0.0220)
RPLR (with validation on tuning set)	$t = 3\log 2$	0.0061(0.0053)	0.0606(0.0087)	0.1172(0.0147)	0.2271(0.0221)
	$t = 2\log 2$	0.0090(0.0064)	0.0613(0.0081)	0.1161(0.0123)	0.2198(0.0173)
	$t = \log 2$	0.0120(0.0084)	0.0663(0.0110)	0.1215(0.0145)	0.2248(0.0179)
	tuned	0.0097(0.0007)	0.0612(0.0008)	0.1150(0.0011)	0.2205(0.0016)
RPLR (with GACV)	$t = 3\log 2$	0.0187(0.0109)	0.0714(0.0123)	0.1280(0.0175)	0.2447(0.0674)
	$t = 2\log 2$	0.0188(0.0117)	0.0688(0.0126)	0.1222(0.0148)	0.2288(0.0335)
	$t = \log 2$	0.0306(0.0192)	0.0782(0.0463)	0.1301(0.0418)	0.2378(0.0325)
Croux and Haesbroeck		0.0104(0.0009)	0.0658(0.0010)	0.1286(0.0019)	0.2335(0.0021)
Bayes Error		0.00	0.05	0.10	0.20

from the training set with the selected λ, class probability of each instance in the testing set is estimated. Both linear and nonlinear learning with Gaussian kernel have been performed. The results show that linear learning works better for this problem.

Figure 2.7 shows the results of the PLR and the RPLR with $t = 2\log 2$. The results when $t = \log 2$ and $t = 3\log 2$ are not reported because they are barely different from the case when $t = 2\log 2$. The horizontal axis stands for the estimated value of linear predictor $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x}+b$, and the vertical axis stands for the estimated probability. The observations of the classes ALL and AML are plotted as circles and squares respectively, with a color scheme of blue for the training set and red for the testing set. The solid and dashed lines are the estimated density curves of the values of linear predictors for the ALL and AML classes, respectively. Here, the class probabilities for the PLR were estimated by $\hat{P}(\boldsymbol{x}) = e^{\hat{f}}/(1 + e^{\hat{f}})$. For the RPLR, we use scheme 3 to estimate the class probabilities. In both procedures of probability estimation, $\hat{f}(\boldsymbol{x}) > 0$ implies $\hat{P}(\boldsymbol{x}) > 0.5$, hence the $sign(\hat{f}(\boldsymbol{x}))$ gives class prediction. As shown in Figure 2.7, there are two common misclassified observations by the PLR and RPLR. This is not surprising considering the nature of the data revealed by the heatmaps. Besides the two misclassified observations, the PLR and the RPLR show different patterns in class probability estimation. The estimated class probabilities by the RPLR are either very close to 1 or 0, while estimated probabilities by the PLR have more variability. This is because that these two classifiers have

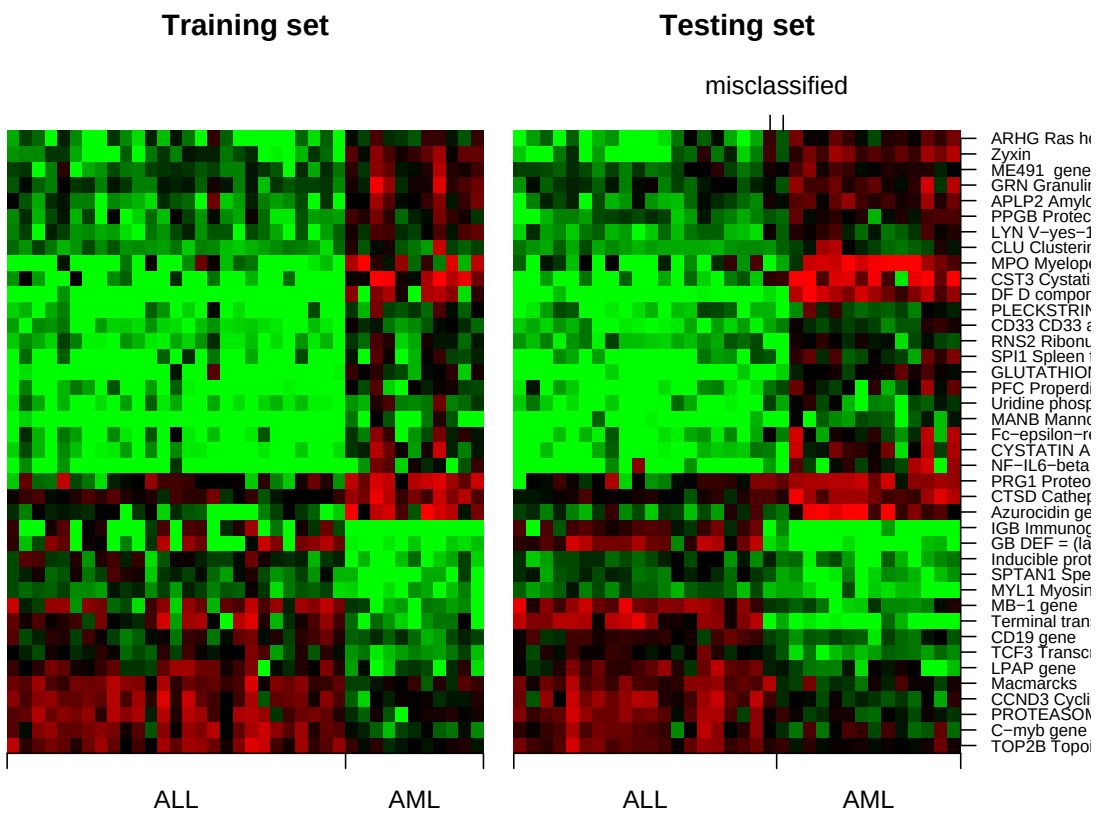

Figure 2.6: Heat maps of the Leukaemia data in Section 2.7.2.1. The left panel is for the training set and the right panel is for the testing set. The red and green colors represent high and low expression values respectively.

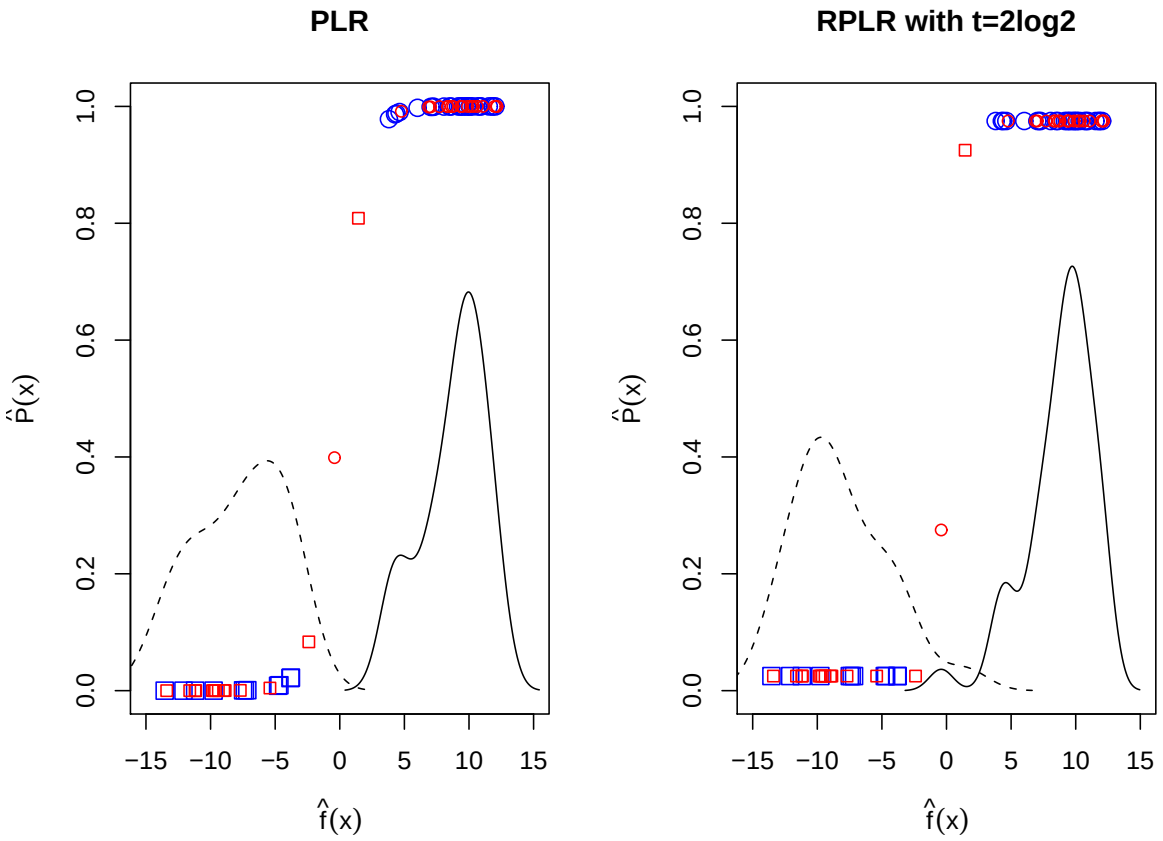

Figure 2.7: Plot of the estimated class probabilities against the estimated values of the linear predictor $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$ for the PLR and the RPLR with $t = 2\log 2$. The solid and the dashed lines are the estimated density curves of the values of linear predictor for ALL and AML class, respectively.

Table 2.2: Class probability estimation errors of the simulated linear example (Example 2.7.1.1)

Method		Scheme	$v = 0$	$v = 5$	$v = 10$	$v = 20$
PLR			0.0464(0.0599)	0.1342(0.0619)	0.1487(0.0461)	0.1350(0.0351)
RPLR (with validation on tuning set	$t = 3\log 2$	1	0.0207(0.0489)	0.1101(0.0290)	0.1350(0.0290)	0.1289(0.0303)
		2	0.0173(0.0394)	0.0994(0.0266)	0.1236(0.0327)	0.1270(0.0318)
		3	0.0438(0.0367)	0.0686(0.0339)	0.1022(0.0405)	0.1184(0.0412)
	$t = 2\log 2$	3	0.0614(0.0499)	0.0676(0.0321)	0.0934(0.0350)	0.1053(0.0409)
	$t = \log 2$	3	0.0758(0.0729)	0.0887(0.0793)	0.1057(0.0592)	0.1185(0.0403)
RPLR (with GACV)	$t = 3\log 2$	1	0.1152(0.0084)	0.1248(0.0155)	0.1323(0.0147)	0.1279(0.0262)
		2	0.0861(0.0072)	0.1034(0.0172)	0.1208(0.0211)	0.1254(0.0275)
		3	0.1053(0.0097)	0.0975(0.0192)	0.1084(0.0284)	0.1230(0.0403)
	$t = 2\log 2$	3	0.1193(0.0109)	0.0982(0.0279)	0.1054(0.0260)	0.1053(0.0337)
	$t = \log 2$	3	0.1707(0.0280)	0.1127(0.0648)	0.1096(0.0460)	0.1251(0.0560)
Croux and Haesbroeck			0.0104(0.0009)	0.0865(0.0015)	0.1208(0.0012)	0.1238(0.0015)

different sensitivity to outliers: since the PLR is sensitive to those two misclassified observations, the estimated probabilities of other observations are affected so that we lose some certainty about the class memberships for some of the other observations despite the clear pattern of the data. On the other hand, those two misclassified observations do not influence the RPLR as much, hence all the other class probabilities remain close to 0 or 1, which reflect the nature of the data better.

2.7.2.2 Lung Cancer Data In this section, we apply the RPLR to the Lung Cancer Dataset described in Liu et al. (2008). The dataset we use here has 12,625 genes of 188 lung cancer patients with 5 categories. There are five different categories: Adeno, Carcinoid, Colon, SmallCell, and Squamous with 128, 20, 13, 6, 21 patients, respectively. Except Colon, the other four are lung cancer subtypes. First, we calculate the ratio of the standard deviation and the sample mean of each gene, and choose 316 genes with the highest ratios. Then we standardize the genes so that each gene has sample mean 0 and sample standard deviation 1. Figure 2.8 is the biplot of the data after filtering and standardization on Principal Component Analysis (PCA). Out of all five types of cancer, the Adeno group has the most broad spectrum and overlaps much with other types. This matches the biological knowledge that Adeno is a very heterogeneous lung cancer subtype (Bhattacharjee et al., 2001). For that reason, we perform the RPLR to classify Adeno patients versus all other cancer patients.

Table 2.3: Testing errors of the simulated nonlinear example (Example 2.7.1.2)

Method		$v = 0$	$v = 5$	$v = 10$	$v = 20$
SVM		0.0416(0.0126)	0.1120(0.0203)	0.1728(0.0226)	0.2825(0.0307)
RSVM	$s = -1$	0.0420(0.0128)	0.0986(0.0169)	0.1577(0.0225)	0.2722(0.0293)
	$s = 0$	0.0484(0.0178)	0.1092(0.0220)	0.1677(0.0245)	0.2784(0.0295)
PLR		0.0396(0.0121)	0.1103(0.0206)	0.1695(0.0217)	0.2832(0.0309)
RPLR	$t = 3\log 2$	0.0396(0.0121)	0.1009(0.0204)	0.1611(0.0247)	0.2799(0.0321)
	$t = 2\log 2$	0.0396(0.0121)	0.0996(0.0196)	0.1594(0.0269)	0.2814(0.0372)
	$t = \log 2$	0.0464(0.0161)	0.1135(0.0230)	0.1667(0.0238)	0.2776(0.0301)
Bayes Error		0.00	0.05	0.10	0.20

Table 2.4: Class probability estimation errors of the simulated nonlinear example (Example 2.7.1.2)

Method		Scheme	$v = 0$	$v = 5$	$v = 10$	$v = 20$
PLR			0.4997(0.0010)	0.4496(0.0011)	0.3998(0.0008)	0.2999(0.0006)
RPLR	$t = 3\log 2$	1	0.4997(0.0010)	0.4496(0.0013)	0.3998(0.0008)	0.2999(0.0006)
		2	0.0721(0.0319)	0.1422(0.0369)	0.1736(0.0426)	0.1650(0.0341)
		3	0.0910(0.0323)	0.1415(0.0434)	0.1791(0.0475)	0.1764(0.0309)
	$t = 2\log 2$	3	0.0974(0.0373)	0.1581(0.0600)	0.1910(0.0559)	0.1984(0.0390)
	$t = \log 2$	3	0.1136(0.0329)	0.1409(0.0486)	0.1637(0.0405)	0.1693(0.0318)

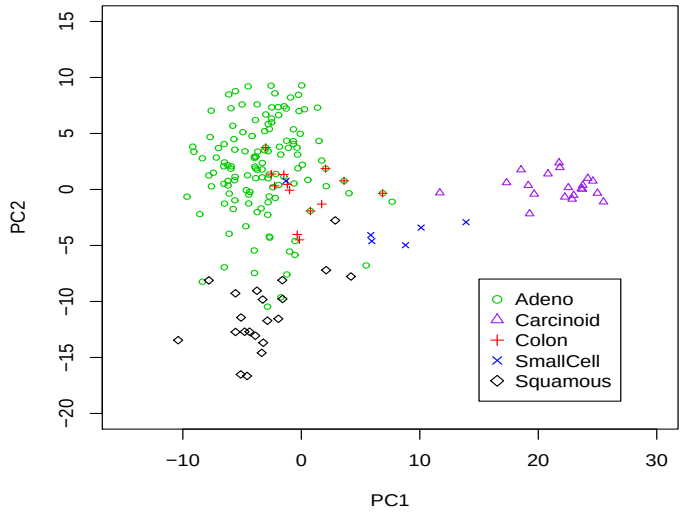

Figure 2.8: Biplot on PCA of the lung cancer data in Section 2.7.2.2.

Since there are 188 cancer patients in total, we randomly divide patients into training, tuning, and testing sets with sample sizes 63, 63, 62, respectively. Then we build a model for each value of λ and choose the λ that gives the smallest misclassification rate on the tuning set. Using the model with the selected λ, the misclassification rate on the testing set is calculated. This whole procedure is repeated for 10 times.

Table 2.5: Testing errors of the Lung Cancer Data example in Section 7.2.2.

Method		Testing Error
PLR		0.1274(0.0052)
RPLR	$t = 3\log 2$	0.1242(0.0051)
	$t = 2\log 2$	0.1210(0.0046)
	$t = \log 2$	0.1226(0.0054)

The results are reported in Table 2.5. We can see that although the difference is not very big, truncation indeed improves performance, and the truncation location that we suggest, $t = 2\log 2$, gives the best result.

2.8 Possible Future Work

We have used the L_2 penalty for the regularization term $J(f)$. It is now well known that one can use some other penalty functions to achieve variable selection. Examples of such penalty functions include the L_1 penalty (Tibshirani, 1996; Zhu et al., 2004), the adaptive L_1 penalty (Zou, 2006; Zhang and Lu, 2007), the SCAD penalty (Fan and Li, 2001; Zhang et al., 2006), the COSSO penalty (Lin and Zhang, 2006; Zhang, 2006; Yuan and Lin, 2006), etc. A natural extension of the RPLR is to use different penalty functions to achieve simultaneous variable selection and robust classification. Moreover, although we have focused on the binary case so far, the truncated logistic loss is applicable for multicategory classification problems as well. The work of Zhu and Hastie (2005) can be useful here. Further development is needed.

2.9 Proofs

2.9.1 Proof of Theorem 1

Since $E[h_\pi(Y)g_s(Yf(\boldsymbol{X}))] = E[E[h_\pi(Y)g_s(Yf(\boldsymbol{X}))|\boldsymbol{X}=\boldsymbol{x}]]$, we can minimize $E[h_\pi(Y)g_s(Yf(\boldsymbol{X}))]$ by minimizing $E[h_\pi(Y)g_s(Yf(\boldsymbol{X}))|\boldsymbol{X}=\boldsymbol{x}]$ for every $\boldsymbol{x}$. Note that $E[h_\pi(Y)g_s(Yf(\boldsymbol{X}))|\boldsymbol{X}=\boldsymbol{x}] = P(\boldsymbol{x})(1-\pi)g_s(f(\boldsymbol{x})) + (1-P(\boldsymbol{x}))\pi g_s(-f(\boldsymbol{x}))$. Because g_s is a nonincreasing function, the minimizer f_π^* should satisfy that $f_\pi^* \geqslant 0$ if $P(\boldsymbol{x})(1-\pi) > (1-P(\boldsymbol{x}))\pi$, $f_\pi^* \leqslant 0$ otherwise. Note that $P(\boldsymbol{x})(1-\pi) > (1-P(\boldsymbol{x}))\pi$ is equivalent to $P(\boldsymbol{x}) > \pi$. Hence, it is sufficient to show that $f=0$ is not a minimizer. We can assume $P(\boldsymbol{x}) > \pi$ without loss of generality. For $s=0$, $E[h_\pi(Y)g_s(0)|\boldsymbol{X}=\boldsymbol{x}] = P(\boldsymbol{x})(1-\pi)g_s(0) + (1-P(\boldsymbol{x}))\pi g_s(0)$, and $E[h_\pi(Y)g_s(1)|\boldsymbol{X}=\boldsymbol{x}] = P(\boldsymbol{x})(1-\pi)g_s(1) + (1-P(\boldsymbol{x}))\pi g_s(-1)$. Hence $E[h_\pi(Y)g_s(0)|\boldsymbol{X}=\boldsymbol{x}] > E[h_\pi(Y)g_s(1)|\boldsymbol{X}=\boldsymbol{x}]$ because $g_s(0) > g_s(1)$ and $g_s(0) = g_s(-1)$. Thus $f=0$ is not a minimizer in this case. For $s<0$, $\frac{d}{df(\boldsymbol{x})}E[h_\pi(Y)g_s(Yf(\boldsymbol{X}))|\boldsymbol{X}=\boldsymbol{x}]|_{f(\boldsymbol{x})=0} = \frac{d}{df(\boldsymbol{x})}[P(\boldsymbol{x})(1-\pi)g_s(f(\boldsymbol{x})) + (1-P(\boldsymbol{x}))\pi g_s(-f(\boldsymbol{x}))]|_{f(\boldsymbol{x})=0} = P(\boldsymbol{x})(1-\pi)g_s'(0) + (1-P(\boldsymbol{x}))\pi g_s'(0) = (P(\boldsymbol{x})-\pi)g_s'(0) < 0$ because $g_s'(0) < 0$. Thus $f=0$ is not a minimizer. Hence, $f_{pi}^*(\boldsymbol{x})$ has the same sign as $P(\boldsymbol{x})-\pi$. □

2.9.2 Proof of Theorem 2

Define $A(f) = E[h_\pi(Y)g_s(Yf(\boldsymbol{X}))|\boldsymbol{X}=\boldsymbol{x}]$. Observe that $A(f) = P(\boldsymbol{x})(1-\pi)\min(t, \log(1+e^{-f(\boldsymbol{x})})) + (1-P(\boldsymbol{x}))\pi\min(t, \log(1+e^{f(\boldsymbol{x})}))$, where $t = \log(1+e^{-s})$. We consider three cases, $s \leqslant f \leqslant -s$, $f < s$, and $f > -s$.

First, when $s \leqslant f \leqslant -s$, $A'(f) = \frac{d}{df(\boldsymbol{x})}[P(\boldsymbol{x})(1-\pi)\log(1+e^{-f}) + (1-P(\boldsymbol{x}))\pi\log(1+e^{f})] = \frac{1}{1+e^f}[-P(\boldsymbol{x})(1-\pi) + (1-P(\boldsymbol{x}))\pi e^f]$, and $A''(f) = (P(\boldsymbol{x})(1-\pi) + (1-P(\boldsymbol{x}))\pi)e^f/(1+e^f)^2$. Note that $A''(f) > 0$ for any $f \in [s,-s]$, and $A'(\tilde{f}) = 0$ when $\tilde{f} = \log\frac{(1-\pi)P(\boldsymbol{x})}{\pi(1-P(\boldsymbol{x}))} = \log\tau(P(\boldsymbol{x}),\pi)$. Hence, $\tilde{f}$ is the minimizer of $A(f)$ for $f \in [s,-s]$. Note that $A(\tilde{f}) = (P(\boldsymbol{x})(1-\pi) + (1-P(\boldsymbol{x}))\pi)\log(P(\boldsymbol{x})(1-\pi) + (1-P(\boldsymbol{x}))\pi) - P(\boldsymbol{x})(1-\pi)\log(P(\boldsymbol{x})(1-\pi)) - (1-P(\boldsymbol{x}))\pi\log((1-P(\boldsymbol{x}))\pi)$.

Second, when $f < s$, note that $A(f) = P(\boldsymbol{x})(1-\pi)t + (1-P(\boldsymbol{x}))\pi\log(1+e^{f(\boldsymbol{x})})$ and it is an increasing function in f. Thus, the minimum of $A(f)$ in this case is $\lim_{f\to-\infty}A(f) = P(\boldsymbol{x})(1-\pi)t$.

Similarly, when $f > -s$, $A(f) = P(\boldsymbol{x})(1-\pi)\log(1+e^{-f(\boldsymbol{x})}) + (1-P(\boldsymbol{x}))\pi t$ and it is

a decreasing function in f. Likewise, the minimum of $A(f)$ in this case is $\lim_{f\to\infty} A(f) = (1-P(\boldsymbol{x}))\pi t$.

Hence, $\tilde{f}$ is the minimizer of $A(f)$ if $A(\tilde{f}) < \lim_{f\to-\infty} A(f) = P(\boldsymbol{x})(1-\pi)t$ and $A(\tilde{f}) < \lim_{f\to\infty} A(f) = (1-P(\boldsymbol{x}))\pi t$. If $A(\tilde{f}) > \lim_{f\to-\infty} A(f) = P(\boldsymbol{x})(1-\pi)t$ and $\lim_{f\to\infty} A(f) = (1-P(\boldsymbol{x}))\pi t > \lim_{f\to-\infty} A(f) = P(\boldsymbol{x})(1-\pi)t$, $f=-\infty$ is the minimizer of $A(f)$. Similarly, $f=\infty$ is the minimizer of $A(f)$ if $A(\tilde{f}) > \lim_{f\to\infty} A(f) = P(\boldsymbol{x})(1-\pi)t$ and $\lim_{f\to\infty} A(f) = (1-P(\boldsymbol{x}))\pi t < \lim_{f\to-\infty} A(f) = P(\boldsymbol{x})(1-\pi)t$. Finally, if $A(\tilde{f}) > \lim_{f\to\infty} A(f) = P(\boldsymbol{x})(1-\pi)t = \lim_{f\to-\infty} A(f) = P(\boldsymbol{x})(1-\pi)t$, then $f=-\infty,\infty$ is the minimizer of $A(f)$. The desired results can follow with that $H_1(\pi, P(\boldsymbol{x})) = tA(\tilde{f})/\lim_{f\to-\infty} A(f)$ and $H_2(\pi, P(\boldsymbol{x})) = tA(\tilde{f})/\lim_{f\to\infty} A(f)$. □

2.9.3 Proof of Lemma 1

Let $\boldsymbol{z}^{(-i)} = (z_1, \cdots, z_{i-1}, P_\lambda^{(-i)}(\boldsymbol{x}_i), z_{i+1}, \cdots, z_n)^T$, and $-\tilde{l}^*(z,\tau) = -z\tau + \log(1+e^\tau)$. Since $-\frac{\partial \tilde{l}^*(z,\tau)}{\partial \tau} = -z + 1/(1+e^{-\tau})$ and $-\frac{\partial^2 \tilde{l}^*(z,\tau)}{\partial \tau^2} = e^\tau/(1+e^\tau)^2 \geqslant 0$, for any fixed z, the minimizer of $-\tilde{l}^*(z,\tau)$ is τ which satisfies $z = 1/(1+e^{-\tau})$. Therefore, using $P_\lambda^{(-i)}(\boldsymbol{x}_i) = 1/(1+e^{-f_\lambda^{(-i)}(\boldsymbol{x}_i)})$, we have $-\tilde{l}^*(P_\lambda^{(-i)}(\boldsymbol{x}_i), f_\lambda^{(-i)}(\boldsymbol{x}_i)) \leqslant -\tilde{l}^*(P_\lambda^{(-i)}(\boldsymbol{x}_i), f_\lambda(\boldsymbol{x}_i))$. This implies

$$-\tilde{l}(P_\lambda^{(-i)}(\boldsymbol{x}_i), f_\lambda^{(-i)}(\boldsymbol{x}_i)) \leqslant -\tilde{l}(P_\lambda^{(-i)}(\boldsymbol{x}_i), f_\lambda(\boldsymbol{x}_i)) \tag{2.35}$$

since $-\tilde{l}(z_i, f(\boldsymbol{x}_i)) = \min\{t, -\tilde{l}^*(z_i, f(\boldsymbol{x}_i))\}$. Hence, for any $\boldsymbol{f}$, we have

$$\begin{aligned} I_\lambda(\boldsymbol{f}, \boldsymbol{z}^{(-i)}) &= -\tilde{l}(P_\lambda^{(-i)}(\boldsymbol{x}_i), f(\boldsymbol{x}_i)) - \textstyle\sum_{j\neq i} \tilde{l}(z_j, f(\boldsymbol{x}_j)) + n\lambda J(f) \\ &\geqslant -\tilde{l}(P_\lambda^{(-i)}(\boldsymbol{x}_i), f^{(-i)}(\boldsymbol{x}_i)) - \textstyle\sum_{j\neq i} \tilde{l}(z_j, f(\boldsymbol{x}_j)) + n\lambda J(f) \\ &\geqslant -\tilde{l}(P_\lambda^{(-i)}(\boldsymbol{x}_i), f^{(-i)}(\boldsymbol{x}_i)) - \textstyle\sum_{j\neq i} \tilde{l}(z_j, f_\lambda^{(-i)}(\boldsymbol{x}_j)) + n\lambda J(f_\lambda^{(-i)}) \end{aligned} \tag{2.36}$$

using (2.35) and the definition of $f_\lambda^{(-i)}$. Therefore, we have $f^*(i, P_\lambda^{(-i)}(\boldsymbol{x}_i), \cdot) = f_\lambda^{(-i)}(\cdot)$. □

Chapter 3

Bounded Constraint Machine

3.1 Introduction

The Support Vector Machine (SVM) has been very popular due to its success in many applications (Vapnik, 1998; Cristianini and Shawe-Taylor, 2000). It was originally proposed using the idea of maximal separation. It is well known that the SVM can be fit in $loss + penalty$ framework using the hinge loss. In this regularization framework, *loss* measures goodness of fit on the training data, and *penalty* reflects smoothness of the resulting model. Viewing the SVM in the regularization framework with the hinge loss helps us understand how the SVM uses the training data to build a classifier.

Despite its success, the SVM has some drawbacks. One known drawback is that the SVM classifier only depends on the set of SVs, which include training data points that are correctly classified but relatively close to the boundary as well as those misclassified training points. As a result, extreme outliers can have relatively big impact on the resulting classifier. In the literature, there have been some attempts to modify the SVM to gain robustness to outliers (Shen et al., 2003; Liu and Shen, 2006; Collobert et al., 2006; Wu and Liu, 2007). The idea is to truncate the unbounded hinge loss function so that the effect of extreme outliers can be bounded. The corresponding optimization, however, involves challenging nonconvex minimization. Another drawback is that the standard SVM was originally designed for binary classification. Its extension to multicategory classification is nontrivial. Previous attempts include Vapnik (1998); Weston and Watkins (1999); Crammer and Singer (2001); Lee et al. (2004). Despite these extensions seem natural and reasonable, not all of them are Fisher consistent (Liu, 2007).

Our motivation here is to modify the criterion of the SVM. Instead of the maximum separation criterion whose solution only depends on a subset of the training data, we propose to use an alternative criterion so that all data points can influence the solution. One main advantage of using all data points for the classifier is that the resulting classifier may depend less heavily on a smaller subset and consequently can be more robust to outliers. More specifically, we propose the Bounded Constraint Machine (BCM), which minimizes the sum of the signed distance to the classification boundary subject to some constraints on the solution. Our focus in this chapter is on binary classification. However, the BCM can be extended for multicategory classification directly with Fisher consistency.

To further study the relationship between the SVM and the BCM, we investigate another method, the Balancing Support Vector Machine (BSVM). The BSVM can be viewed as a modification of the SVM with all training points influencing the resulting classifier. The BSVM is characterized using the parameter v with $v = 0$ corresponding to the SVM and $v = \infty$ corresponding to the BCM. As a result, the BSVM helps to build a continuous path from the SVM to the BCM by changing the value of v. Along with the effect of v, the properties of the BSVM including Fisher consistency and asymptotic behaviors of the coefficients are investigated.

In practice, the performance of these methods may vary from problem to problem. Therefore, it may be desirable to treat v data dependent. To improve the computational efficiency, we establish the entire solution path with respect to the value of v, so that we can get the solution of the BSVM for every value of v efficiently.

The rest of this chapter is organized as follows. Section 3.2 briefly reviews the standard SVM and proposes the BCM. In Section 3.3, we investigate the BSVM and describe its behavior using the Lagrange dual problem. The effect of v is explored and we show how the BSVM builds connection from the SVM to the BCM. Section 3.4 shows Fisher consistency of the BSVM and the BCM, as well as some asymptotic properties. Section 3.5 develops the regularized solution path with respect to v. Numerical results are reported in Section 3.6 and Section 3.7 gives some discussion. The proofs of our theorems are included in Section 3.8.

3.2 The SVM and the BCM

3.2.1 The Standard SVM

The SVM is a typical method of form (1.1). In particular, it employs the hinge loss function $L(yf(\boldsymbol{x})) = [1 - yf(\boldsymbol{x})]_+$, and the penalty term $J(f) = \frac{1}{2}\|\boldsymbol{w}\|^2$. Note that the value of the hinge loss $L(yf(\boldsymbol{x}))$ increases as $yf(\boldsymbol{x})$ becomes smaller and it stays at zero when $yf(\boldsymbol{x}) \geqslant 1$. That is, the SVM puts loss on the misclassified data points but nothing on the correctly classified observations once $yf(\boldsymbol{x})$ becomes greater than 1. Hence the data points with $yf(\boldsymbol{x}) \geqslant 1$ have no influence on the SVM solution. To further explain, we express the dual problem

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle - \sum_{i=1}^{n} \alpha_i \\ &\text{subject to} \quad \sum_{i=1}^{n} y_i \alpha_i = 0; 0 \leqslant \alpha_i \leqslant C, \forall i = 1, \cdots, n. \end{aligned} \tag{3.1}$$

Using the α_i obtained from (3.1), $\boldsymbol{w}$ can be calculated as $\sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i$, and b can be obtained by the KKT conditions. Thus the classification function can be written as $f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i y_i \langle \boldsymbol{x}_i, \boldsymbol{x} \rangle + b$. Furthermore, $\alpha_i > 0$ implies $y_i f(\boldsymbol{x}_i) \leqslant 1$ and actually that is the only case that $(\boldsymbol{x}_i, y_i)$ can affect the solution. On the other hand, when $\alpha_i = 0$, the observation $(\boldsymbol{x}_i, y_i)$ has no impact on the solution. A point $\boldsymbol{x}_i$ with $\alpha_i > 0$ is a SV, which is the observation satisfying $y_i f(\boldsymbol{x}_i) \leqslant 1$.

3.2.2 The BCM

Due to the design of the SVM, its solution only depends on the set of SVs. This helps to simplify the solution. However, if the training dataset is noisy with outliers, the solution can be deteriorated. To solve the problem, we propose a different optimization criterion. In particular, we propose to minimize the sum of signed distances to the boundary and solve the following problem

$$\begin{aligned} &\min_f J(f) - C \textstyle\sum_{i=1}^{n} y_i f(\boldsymbol{x}_i) \\ &\text{subject to} \quad -1 \leqslant f(\boldsymbol{x}_i) \leqslant 1, \forall i = 1, \cdots, n. \end{aligned} \tag{3.2}$$

That is, we try to maximize $\sum_{i=1}^{n} y_i f(\boldsymbol{x}_i)$, while forcing all the training data to stay between the hyperplanes $f(\boldsymbol{x}) = \pm 1$. One can view that the BCM uses the hinge loss of the SVM with $y_i f(\boldsymbol{x}_i) \in [-1, 1]$. With the constraints, the BCM makes use of all training points to obtain the

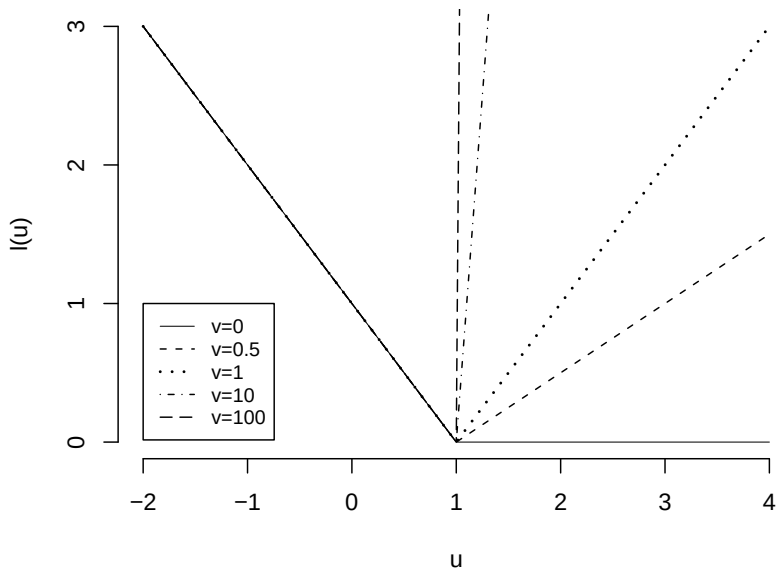

Figure 3.1: Plot of loss function $g(u)$ with different values of v

resulting classifier.

One advantage of the BCM is that it can be extended to the multicategory case directly. Assume that we have a k-class problem with $y \in \{1, \cdots, k\}$. Let $\boldsymbol{f} = (f_1, \cdots, f_k)$ be the decision function vector with $\sum_{j=1}^{k} f_j = 0$. Then the multicategory BCM solves the following problem

$$\begin{aligned} &\min_{\boldsymbol{f}} \sum_{j=1}^{k} \|f_j\|^2 - C \sum_{i=1}^{n} f_{y_i}(\boldsymbol{x}_i) \\ &\text{subject to} \sum_{j=1}^{k} f_j(\boldsymbol{x}_i) = 0; f_l(\boldsymbol{x}_i) \geqslant -1; \forall i = 1, \cdots, n, l = 1, \cdots, k. \end{aligned} \tag{3.3}$$

It can be shown that the multicategory BCM is Fisher consistent, as discussed in Section 3.4.1.

To further understand the connection between the SVM and the BCM, we discuss the BSVM in Section 3.3 and use the BSVM as a bridge to connect the SVM and the BCM.

3.3 The BSVM: A Bridge Between the SVM and the BCM

The SVM only uses the SV set to calculate its solution, while the BCM utilizes all training points. To connect these two, we study the BSVM using the following loss function

$$g(u) = \begin{cases} 1-u & \text{if } u \leqslant 1, \\ v(u-1) & \text{otherwise,} \end{cases} \tag{3.4}$$

where v is the slope of the loss function when $u \in (1, \infty)$, as shown in Figure 3.1. Note that v determines how much the solution will rely on the data points with $yf(\boldsymbol{x}) \geqslant 1$, and the problem becomes equivalent to the SVM when $v = 0$. Here, we would like to acknowledge that the loss $g(u)$ was previously presented by Ming Yuan in the Statistical Learning Conference at Snowbird, UT in 2007. We use the BSVM as a bridge to connect the SVM with the proposed BCM.

Note that when $v = \infty$, the BSVM becomes equivalent to solving

$$\begin{aligned} &\min_{(b,\boldsymbol{w})} J(f) - C\sum_{i=1}^{n} y_i f(\boldsymbol{x}_i) \\ &\text{subject to} \quad f(\boldsymbol{x}_i) \leqslant 1, \forall i = 1, \cdots, n. \end{aligned} \tag{3.5}$$

Comparing to the BCM in (3.2), the only difference is that the BCM has the constraint $f(\boldsymbol{x}_i) \geqslant -1$ but the BSVM with $v = \infty$ does not. Typically this difference does not matter since the solution of (3.5) usually induces $f(\boldsymbol{x}_i) \geqslant -1$. The only case that the BCM actually works differently from the BSVM with $v = \infty$ is when a data point moves far away from its own class, even further than the other class. This rarely happens in practice. Thus, the BSVM with $v = \infty$ can be viewed as a good approximation of the BCM. Overall, the BSVM builds a continuum from the standard SVM ($v = 0$) to the BCM ($v = \infty$).

3.3.1 Interpretation of the BSVM

Since the loss $g(u)$ for the BSVM is not a decreasing function and it imposes big loss values even on the correctly classified data points as well as misclassified observations, it might seem counterintuitive. However, the increasing part with $y_i f(\boldsymbol{x}_i) > 1$ may help to bring the decision boundary towards the correctly classified points, which can be desirable in some situations. To

understand the behavior of the BSVM further, we rewrite its primal problem as follows

$$\begin{aligned}&\min_{(b,\boldsymbol{w})} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i \\ &\text{subject to} \quad \xi_i \geqslant 1 - y_i f(\boldsymbol{x}_i); \xi_i \geqslant v(y_i f(\boldsymbol{x}_i) - 1), \forall i = 1, \cdots, n.\end{aligned}$$

The corresponding Lagrange primal can be written as

$$\mathfrak{L}(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\gamma_i[1 - y_i f(\boldsymbol{x}_i) - \xi_i] + \sum_{i=1}^{n}\delta_i[vy_i f(\boldsymbol{x}_i) - v - \xi_i]. \tag{3.6}$$

Setting derivatives to zero gives

$$\begin{aligned}\frac{\partial \mathfrak{L}}{\partial \boldsymbol{w}} &= \boldsymbol{w} - \sum_{i=1}^{n} y_i\gamma_i\boldsymbol{x}_i + \sum_{i=1}^{n} vy_i\delta_i\boldsymbol{x}_i = \mathbf{0} &(3.7)\\ \frac{\partial \mathfrak{L}}{\partial b} &= -\sum_{i=1}^{n} y_i\gamma_i + v\sum_{i=1}^{n} y_i\delta_i = 0 &(3.8)\\ \frac{\partial \mathfrak{L}}{\partial \xi_i} &= C - \gamma_i - \delta_i = 0, &(3.9)\end{aligned}$$

and KKT conditions are

$$\gamma_i(1 - y_i f(\boldsymbol{x}_i) - \xi_i) = 0 \tag{3.10}$$

$$\delta_i(vy_i f(\boldsymbol{x}_i) - v - \xi_i) = 0. \tag{3.11}$$

Then, writing $\alpha_i = \gamma_i - v\delta_i$, the corresponding dual problem becomes

$$\begin{aligned}&\min_{\boldsymbol{\alpha}} \frac{1}{2}\sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle - \sum_{i=1}^{n}\alpha_i \\ &\text{subject to} \quad \sum_{i=1}^{n} y_i\alpha_i = 0; -Cv \leqslant \alpha_i \leqslant C, \forall i = 1, \cdots, n. \end{aligned} \tag{3.12}$$

Once the solution of (3.12) is obtained, $\boldsymbol{w}$ can be calculated as $\sum_{i=1}^{n}\alpha_i y_i \boldsymbol{x}_i$ and b can be determined by KKT conditions. This problem is very similar to the SVM problem. The difference is on the constraint. In particular, we have $0 \leqslant \alpha_i \leqslant C$ for the SVM, but $-Cv \leqslant \alpha_i \leqslant C$ for the BSVM. This helps to explain the difference in behaviors between the SVM and the

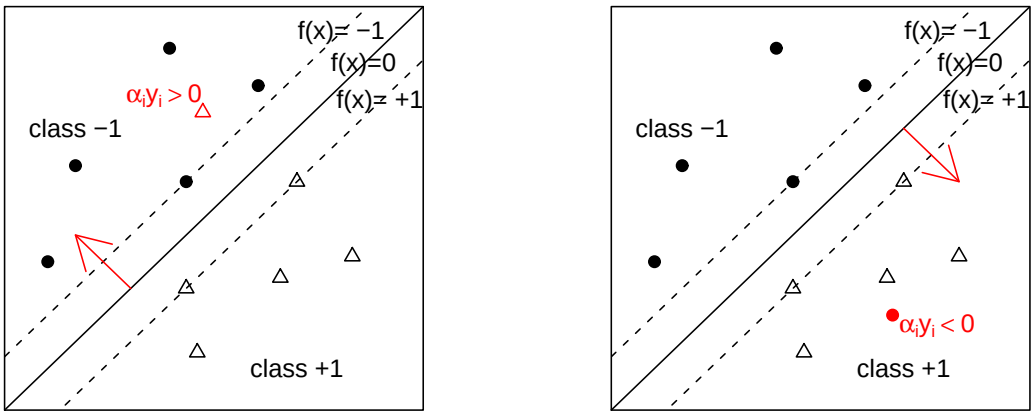

Figure 3.2: Illustration of the effect of $\alpha_i y_i$ in the standard SVM. The left and right panel illustrates that a positive and negative $\alpha_i y_i$ tends to push the boundary towards to the left and right side, respectively.

BSVM. In contrast to the SVM, the BSVM with $v > 0$ makes use of all data points to determine the solution. Points with $y_i f_i \leqslant 1$ may help to reduce the effect of outliers and consequently the BSVM classifier can be more robust against outliers.

In order to further explain the BSVM, we first give some geometric interpretation of the SVM. In the SVM, the support vectors have $\alpha_i > 0$ and $y_i f_i \leqslant 1$, and these are the only observations that affect the resulting decision boundary. They are either of these two cases: when $\alpha_i y_i > 0$ or when $\alpha_i y_i < 0$. First, when $\alpha_i y_i > 0$, $y_i = 1$ because α_i is positive. This implies $f_i \leqslant 1$, which means the observation $\boldsymbol{x}_i$ belongs to class $+1$ but lies close to observations of class -1, like the red point in the left panel of Figure 3.2. In this figure, the triangles and the dots represents the data points which belong to class $+1$ and -1, respectively. The solid line and the dashed lines are the decision boundary ($f(\boldsymbol{x}) = 0$) and the soft margins ($f(\boldsymbol{x}) = \pm 1$) based on the black data points. Adding the red point, the decision boundary $f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i y_i \langle \boldsymbol{x}_i, \boldsymbol{x} \rangle + b$ increases by $\alpha_i y_i \langle \boldsymbol{x}_i, \boldsymbol{x} \rangle$. If we assume $\langle \boldsymbol{x}_i, \boldsymbol{x} \rangle \geqslant 0$ (it is often true when we use kernel representation), then we can see that the SV with $\alpha_i y_i > 0$ increases the value of the decision boundary, which causes the decision boundary to move towards the class -1 side. Similarly, when $\alpha_i y_i < 0$, we can see

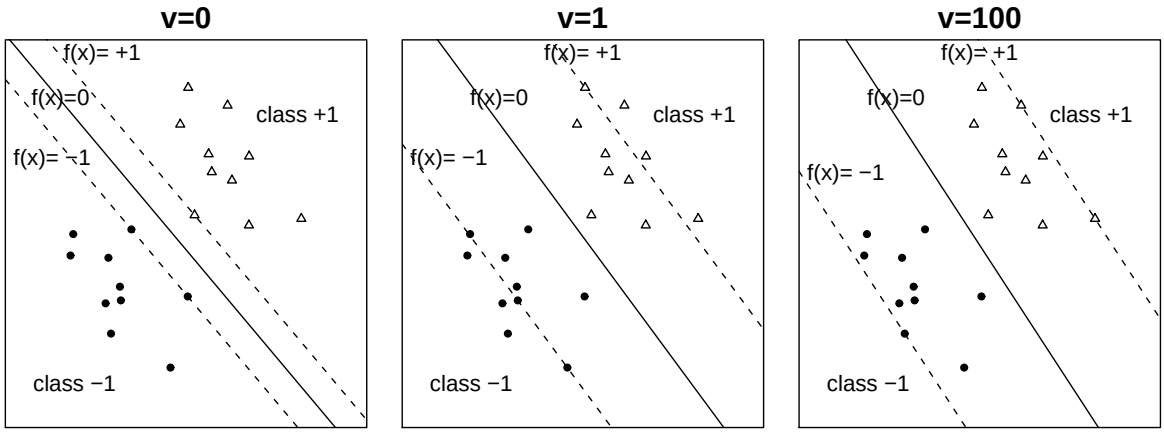

Figure 3.3: Plots of the effect of different values of v on the BSVM.

that $y_i = -1$ and $f_i \geqslant -1$, like the red point in the right panel of Figure 3.2, and this induces decrease in the value of the decision boundary. Thus the decision boundary moves towards the observations of class $+1$. Hence, we can say that misclassified data or data inside of the soft margins pulls the decision boundary to themselves.

Comparing to the SVM, the BSVM uses the data points' information differently because α_i can take negative values as opposed to the SVM case. Note that $\alpha_i > 0$ implies $y_i f_i \leqslant 1$, and $\alpha_i < 0$ implies $y_i f_i \geqslant 1$ by KKT conditions. When $\alpha_i y_i > 0$, not like in the SVM, y_i can be either $+1$ or -1. If $y_i = 1$, things are the same with the SVM case, that is, $\boldsymbol{x}_i$ is a member of the class $+1$ but located close to the class -1, resulting the decision boundary pulled towards the class -1. But if $y_i = -1$, then $\alpha_i < 0$ and $f_i \leqslant -1$, which implies that $\boldsymbol{x}_i$ is correctly classified as class -1. But the effect on decision boundary is the same: it increases the value of the decision boundary by $\alpha_i y_i \langle \boldsymbol{x}_i, \boldsymbol{x} \rangle$. Hence, not only the class $+1$ members close to class -1 but also the correctly classified class -1 entries pull the decision boundary to the side of class -1. Likewise, when $\alpha_i y_i < 0$, we can show that the correctly classified class 1 observations as well as class -1 members located close to class $+1$ pull the decision boundary towards the class $+1$. This unique feature of the BSVM may help to bring robustness against outliers. We discuss further about the effect of v in Section 3.3.2.

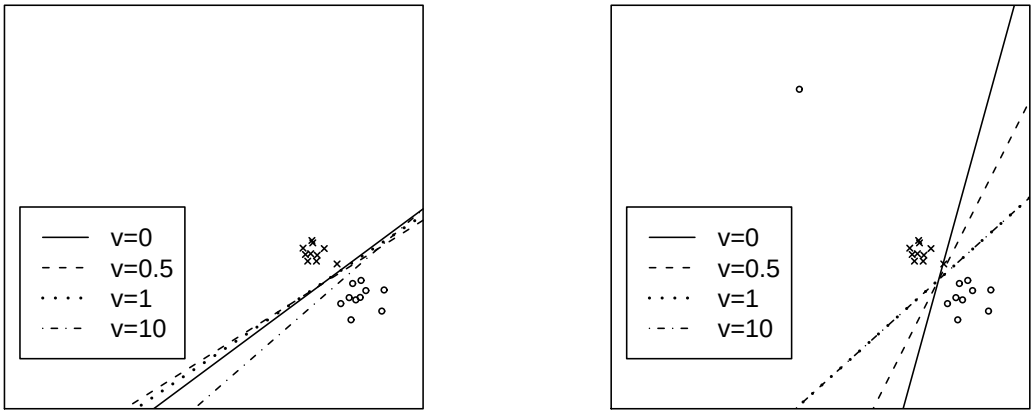

Figure 3.4: A graphical illustration of the robustness of the BSVM: the decision boundary of the BSVM stays stable when there is an extreme outlier, while that of the SVM moves dramatically towards the outlier.

3.3.2 Effect of v

In the separable case, the standard SVM, i.e. the BSVM with $v = 0$, finds the decision boundary which maximizes the distance from the decision boundary to the nearest data point, that is, the distance between $f(\boldsymbol{x}) = \pm 1$ is maximized. Here, the soft margins $f(\boldsymbol{x}) = \pm 1$ are the hyperplanes that bound the data points of each class, so that the observations are forced to lie outside of the soft margins. The BSVM with $v > 0$ maximizes the distance between $f(\boldsymbol{x}) = \pm 1$ as well, but the observations are clustered around the hyperplanes $f(\boldsymbol{x}) = \pm 1$ without being forced to be outside of the margin lines. When $v = 1$, the BSVM minimizes $\sum_i |1 - y_i f(\boldsymbol{x}_i)|$, resulting data points laid inside and outside of $f(\boldsymbol{x}) = \pm 1$ evenly as shown in the middle panel of the Figure 3.3. As the value of v becomes high, the value of $v[y_i f(\boldsymbol{x}_i) - 1]_+$, which is the distance between the hyperplanes $f(\boldsymbol{x}) = \pm 1$ and the observations outside of them, becomes larger. Thus the hyperplanes $f(\boldsymbol{x}) = \pm 1$ move towards outside to reduce it. As v goes to infinity, the BSVM reduces to the BCM and the hyperplanes $f(\boldsymbol{x}) = \pm 1$ go far enough to bound all data points. The right panel of the Figure 3.3 illustrates the behavior of the BCM with large v.

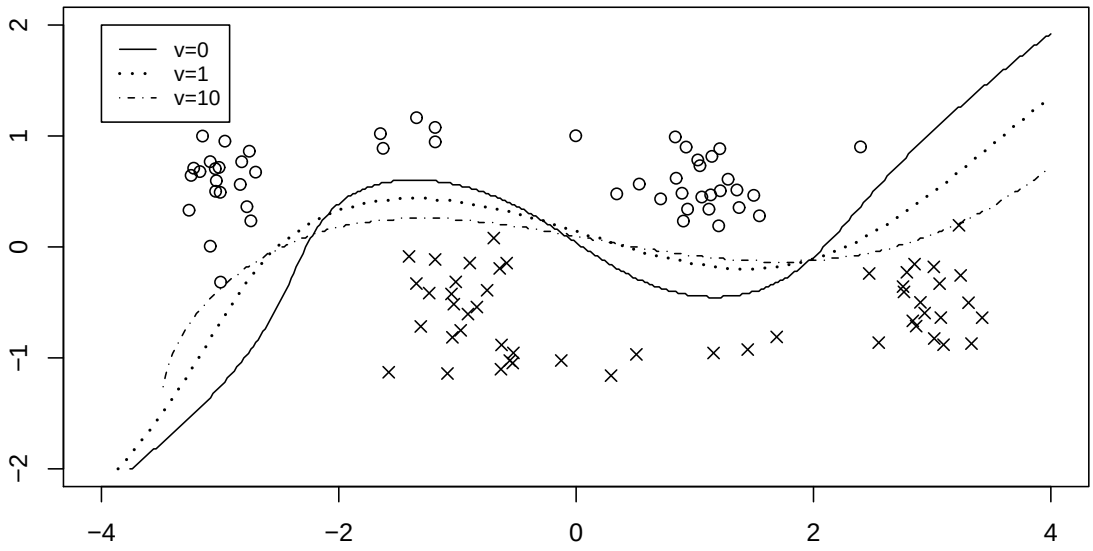

Figure 3.5: A graphical comparison of the SVM vs. BSVM: the decision boundary of the SVM reflects the wavy shaped structure of the data near the border, while that of the BSVM is flatten by the observations far from the border.

Since v decides how much the decision boundary depends on the correctly classified observations, performance of the BSVM is affected by the value of v. The BSVM with big value of v tends to depend on the correctly classified data, which makes it less sensitive against outliers. The BCM can be viewed as the most extreme case with $v = \infty$. The toy example in Figure 3.4 illustrates this behavior. When there is no outlier as shown on the left panel, the SVM and the BSVM with different values of v perform similarly. However, when an observation moves far away from its own class, the decision boundary of the SVM moves towards the outlier, resulting a data point misclassified. In contrast, the BSVM with large v is more stable because the effect of the outlier is greatly reduced by the correctly classified data. Therefore, correctly classified data in the BSVM help to robustify the decision boundary so that a small number of outliers will not cause a drastic change on the decision boundary.

The BSVM may not always produce better results than that of the SVM. It can be suboptimal in a situation as the toy example shown in Figure 3.5. The true boundary is wavy shaped, but

the observations far away from the boundary are aligned in parallel. The SVM works fairly well, but the decision boundary of the BSVM becomes flat as the value of the v goes large due to the influences of the data points far from the boundary. Hence, choice of v should be made carefully based on the characteristic of the problem.

3.4 Properties of the BSVM and the BCM

3.4.1 Fisher Consistency of the BSVM and the BCM

In this section, we discuss Fisher consistency of the BSVM and the BCM. Fisher consistency, also known as classification-calibration (Bartlett et al., 2006), requires that the population minimizer of a loss function has the same sign as $P(x)-1/2$ in the binary case (Lin, 2004). This is a desirable property for a loss function. The following theorem establishes Fisher consistency of the loss function of the BSVM.

Theorem 3. *The minimizer f^* of $E[g(Yf(\boldsymbol{X}))|\boldsymbol{X}=\boldsymbol{x}]$ is sign$[P(\boldsymbol{x})-1/2]$.*

Theorem 3 shows that if $P(\boldsymbol{x}) > 1/2$, we have the theoretical minimizer $f^* = 1$, and otherwise, $f^* = -1$. This matches the fact that the observations are clustered around the hyperplanes $f(\boldsymbol{x}) = \pm 1$.

For the BCM, we consider multicategory case due to its simple extension. In multicategory case, Fisher consistency requires that $\text{argmax}_j f_j^* = \text{argmax}_j P_j$, where $\boldsymbol{f}^*(\boldsymbol{x}) = (f_1^*(\boldsymbol{x}), \ldots, f_k^*(\boldsymbol{x}))$ denotes the minimizer of expected value of the loss function. The following theorem shows Fisher consistency of the loss function of the multicategory BCM.

Theorem 4. *The minimizer $\boldsymbol{f}^*$ of $E[-f_Y(\boldsymbol{X})]$, subject to $\sum_j^k f_j(\boldsymbol{x}) = 0$ and $f_l(\boldsymbol{x}) \geqslant -1$ for $\forall l$, satisfies the following: $f_j^*(\boldsymbol{x}) = k-1$ if $j = argmax_j P_j(\boldsymbol{x})$ and -1 otherwise.*

3.4.2 Asymptotic Property of the BSVM

In this section, we study asymptotic distributions of the coefficients in the BSVM. Koo et al. (2008) established Bahadur type representation (Bahadur, 1966; Chaudhuri, 1991) of the classical SVM coefficients to study their asymptotic behavior. This representation allows us to see how the margin lines of the SVM and the underlying probability distribution of observations

affects asymptotic behavior of the coefficients of large samples. This idea can be generalized to the BSVM with some changes on the Bahadur representation of the coefficients and regularity conditions to adopt the loss function of the BSVM. We show that the coefficients of the BSVM have asymptotic normality, as that of the standard SVM.

First, we introduce new notations for convenience. Let $\boldsymbol{\beta} = (\beta_0, \boldsymbol{\beta}_+)$ denote $(b, \boldsymbol{w})$ which is the coefficients in the BSVM. Let $\tilde{\boldsymbol{x}} = (1, \boldsymbol{x}^T)^T = (1, x_1, \cdots, x_d)^T = (\tilde{x}_0, \tilde{x}_1, \cdots, \tilde{x}_d)^T$ and denote the linear decision function for given $\boldsymbol{X} = \boldsymbol{x}$ as $f(\boldsymbol{x}; \boldsymbol{\beta}) = \boldsymbol{x}^T\boldsymbol{\beta} = \beta_0 + \boldsymbol{x}^T\boldsymbol{\beta}_+$. Let $\pi_+ = P(Y = 1) > 0$ and $\pi_- = P(Y = -1) > 0$, with $\pi_+ + \pi_- = 1$. Let h_+ and h_- be the density functions of $\boldsymbol{X}$ given $Y = 1$ and -1, respectively. Denote the objective function of the BSVM

$$q_{\lambda,n}(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n} g(y_i f(\boldsymbol{x}_i; \boldsymbol{\beta})) + \frac{\lambda}{2}\|\boldsymbol{\beta}_+\|. \tag{3.13}$$

The population version of (3.13) without the penalty term is denoted by

$$Q(\boldsymbol{\beta}) = E[g(Y f(\boldsymbol{X}; \boldsymbol{\beta}))] \tag{3.14}$$

and the minimizers of (3.13) and (3.14) are denoted by $\hat{\boldsymbol{\beta}}_{\lambda,n}$ and $\boldsymbol{\beta}^*$. Defining the indicator function $\rho(z) = I_{\{z \geqslant 0\}}$ for $z \in \mathbb{R}$, we denote the (d+1)-dimensional vector $S(\boldsymbol{\beta}) = E[-\rho(1 - Y f(\boldsymbol{X}; \boldsymbol{\beta}))Y\tilde{\boldsymbol{X}} + v\rho(Y f(\boldsymbol{X}; \boldsymbol{\beta}) - 1)Y\tilde{\boldsymbol{X}}]$ and the $(d+1) \times (d+1)$ matrix $H(\boldsymbol{\beta}) = (1+v)E[\delta(1 - Y f(\boldsymbol{X}; \boldsymbol{\beta}))\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^T]$, where δ is the Dirac delta function. It is proved in the Appendix that $S(\boldsymbol{\beta})$ and $H(\boldsymbol{\beta})$ are the gradient and Hessian matrix of $Q(\boldsymbol{\beta})$, respectively.

Now we state the regularity conditions for the asymptotic results. Here, $C_1, C_2, \cdots$ are positive constants which do not depend on n.

A1 The densities h_+ and h_- are continuous and have finite second moments.

A2 There exists $B(\boldsymbol{x}_0, r_0)$, a ball centered at $\boldsymbol{x}_0$ with radius $r_0 > 0$ such that $\pi_+ h_+(\boldsymbol{x}) + \pi_- h_-(\boldsymbol{x}) > C_1$ for every $\boldsymbol{x} \in B(\boldsymbol{x}_0, r_0)$

A3 For some $1 \leqslant i^* \leqslant d$,

$$\pi_+ \left\{ \int_{\mathcal{X}} (I_{\{x_{i^*} \leqslant F_{i^*}^+\}} - v I_{\{x_{i^*} > F_{i^*}^+\}}) x_{i^*} h_+(\boldsymbol{x}) d\boldsymbol{x} \right\} > \pi_- \left\{ \int_{\mathcal{X}} (I_{\{x_{i^*} \geqslant G_{i^*}^-\}} - v I_{\{x_{i^*} < G_{i^*}^-\}}) x_{i^*} h_-(\boldsymbol{x}) d\boldsymbol{x} \right\}$$

or

$$\pi_+\left\{\int_{\mathcal{X}}(I_{\{x_{i^*}\geqslant F^-_{i^*}\}}-vI_{\{x_{i^*}<F^-_{i^*}\}})x_{i^*}h_+(\boldsymbol{x})d\boldsymbol{x}\right\}<\pi_-\left\{\int_{\mathcal{X}}(I_{\{x_{i^*}\leqslant G^+_{i^*}\}}-vI_{\{x_{i^*}>G^+_{i^*}\}})x_{i^*}h_-(\boldsymbol{x})d\boldsymbol{x}\right\}$$

for $F^+_{i^*}, G^+_{i^*}, F^-_{i^*}, G^-_{i^*}\in[-\infty,\infty]$ such that

$$\begin{aligned}
&\int_{\mathcal{X}} I_{\{x_{i^*}\leqslant F^+_{i^*}\}}h_+(\boldsymbol{x})d\boldsymbol{x}=\min\left\{1,\tfrac{\frac{\pi_-}{\pi_+}+v}{1+v}\right\},\quad \int_{\mathcal{X}} I_{\{x_{i^*}\leqslant G^+_{i^*}\}}h_-(\boldsymbol{x})d\boldsymbol{x}=\min\left\{1,\tfrac{\frac{\pi_+}{\pi_-}+v}{1+v}\right\},\\
&\int_{\mathcal{X}} I_{\{x_{i^*}\geqslant F^-_{i^*}\}}h_+(\boldsymbol{x})d\boldsymbol{x}=\min\left\{1,\tfrac{\frac{\pi_-}{\pi_+}+v}{1+v}\right\},\quad \int_{\mathcal{X}} I_{\{x_{i^*}\geqslant G^-_{i^*}\}}h_-(\boldsymbol{x})d\boldsymbol{x}=\min\left\{1,\tfrac{\frac{\pi_+}{\pi_-}+v}{1+v}\right\}.
\end{aligned}$$

A4 For an orthogonal transformation A_{j^*} that maps $\boldsymbol{\beta}^*_+/\|\boldsymbol{\beta}^*_+\|$ to the j^*-th unit vector e_{j^*} for some $1\leqslant j^*\leqslant d$, there exist rectangles

$$\mathcal{D}^+=\{\boldsymbol{x}\in M^+ : l_i\leqslant (A_{j^*}\boldsymbol{x})_i\leqslant v_i \text{ with } l_i<v_i \text{ for } i\neq j^*\}$$

and

$$\mathcal{D}^-=\{\boldsymbol{x}\in M^- : l_i\leqslant (A_{j^*}\boldsymbol{x})_i\leqslant v_i \text{ with } l_i<v_i \text{ for } i\neq j^*\}$$

such that $h_+(\boldsymbol{x})\geqslant C_2>0$ on $\mathcal{D}^+$, and $h_-(\boldsymbol{x})\geqslant C_3>0$ on $\mathcal{D}^-$, where $M^+=\{\boldsymbol{x}\in\mathcal{X}|\beta^*_0+\boldsymbol{x}^T\boldsymbol{\beta}^*_+=1\}$ and $M^-=\{\boldsymbol{x}\in\mathcal{X}|\beta^*_0+\boldsymbol{x}^T\boldsymbol{\beta}^*_+=-1\}$

Note that **A1** is needed to guarantee that $S(\boldsymbol{\beta})$ and $H(\boldsymbol{\beta})$ are well-defined and continuous in $\boldsymbol{\beta}$. If **A1** is met, the condition that $h_+(bx_0)>0$ or $h_-(bx_0)>0$ for some $\boldsymbol{x}_0$ implies **A2**. **A3** is the condition to ensure that $\boldsymbol{\beta}^*_+\neq\mathbf{0}$, and if $\pi_+=\pi_-$, then it simply means that the mean vectors of conditional class distribution are different. **A4** ensures the positive-definiteness of $H(\boldsymbol{\beta})$ around $\boldsymbol{\beta}^*$. This condition is easily satisfied when the supports of h_+ and h_- are convex. Assuming these regularity conditions, we have a Bahadur-type representation of $\hat{\boldsymbol{\beta}}_{\lambda,n}$ as shown in Theorem 5. This induces the asymptotic normality of $\hat{\boldsymbol{\beta}}_{\lambda,n}$ (Theorem 6).

Theorem 5. *Suppose* **A1**-**A4** *are satisfied. Then, for* $\lambda=o(n^{-1/2})$,

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_{\lambda,n}-\boldsymbol{\beta}^*)=-\frac{1}{\sqrt{n}}H(\boldsymbol{\beta}^*)^{-1}\sum_{i=1}^{n}(I_{\{y_if(\boldsymbol{X}_i;\boldsymbol{\beta}^*)\leqslant 1\}}-vI_{\{y_if(\boldsymbol{X}_i;\boldsymbol{\beta}^*)>1\}})y_i\tilde{\boldsymbol{X}}_i+o_{\mathbb{P}}(1).$$

Theorem 6. *Suppose* **A1-A4** *are satisfied. Then, for* $\lambda = o(n^{-1/2})$,

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_{\lambda,n} - \boldsymbol{\beta}^*) \to N(0, H(\boldsymbol{\beta}^*)^{-1} G(\boldsymbol{\beta}^*) H(\boldsymbol{\beta}^*)^{-1})$$

in distribution as $n \to \infty$, *where*

$$G(\boldsymbol{\beta}) = E[(I_{\{y_i f(\boldsymbol{X}_i;\boldsymbol{\beta}^*) \leqslant 1\}} + v^2 I_{\{y_i f(\boldsymbol{X}_i;\boldsymbol{\beta}^*) > 1\}}) \tilde{\boldsymbol{X}} \tilde{\boldsymbol{X}}^T].$$

This result can be used for building a confidence bound for $\boldsymbol{\beta}$ or $f(\boldsymbol{x};\boldsymbol{\beta})$ for a specific $\boldsymbol{x}$. The proofs are in Section 3.8.

To illustrate the result on asymptotics, we introduce a simple toy example as follows. Let the one-dimensional explanatory variable x has normal distribution with mean 1 and variance 1 if it belongs to class 1, and otherwise normal distribution with mean -1 and variance 1. Then it can be easily shown that $\beta_0^* = 0$ and $\beta_+^* = 1$, which gives

$$H(\beta^*) = (1+v) \begin{pmatrix} (2\pi)^{-1/2} & 0 \\ 0 & (2\pi)^{-1/2} \end{pmatrix},$$

and

$$G(\beta^*) = \begin{pmatrix} \frac{1}{2}(1+v^2) & 0 \\ 0 & (1+v^2) + \sqrt{\frac{2}{\pi}}(v^2-1) \end{pmatrix}.$$

Thus, by Theorem 6, we have

$$\sqrt{n} \begin{pmatrix} \hat{\boldsymbol{\beta}}_0 \\ \hat{\boldsymbol{\beta}}_+ \end{pmatrix} \to N \left(\begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \frac{1}{(1+v)^2} \begin{pmatrix} \pi(1+v^2) & 0 \\ 0 & 2\pi(1+v^2) + 2\sqrt{2\pi}(v^2-1) \end{pmatrix} \right). \tag{3.15}$$

The asymptotic variances of coefficients shown in (3.15) depends on v. As shown in Figure 3.6, the variances of both coefficients decrease as v increases for a while, then increase in v. Thus in this example the middle range values of v give smaller asymptotic variances.

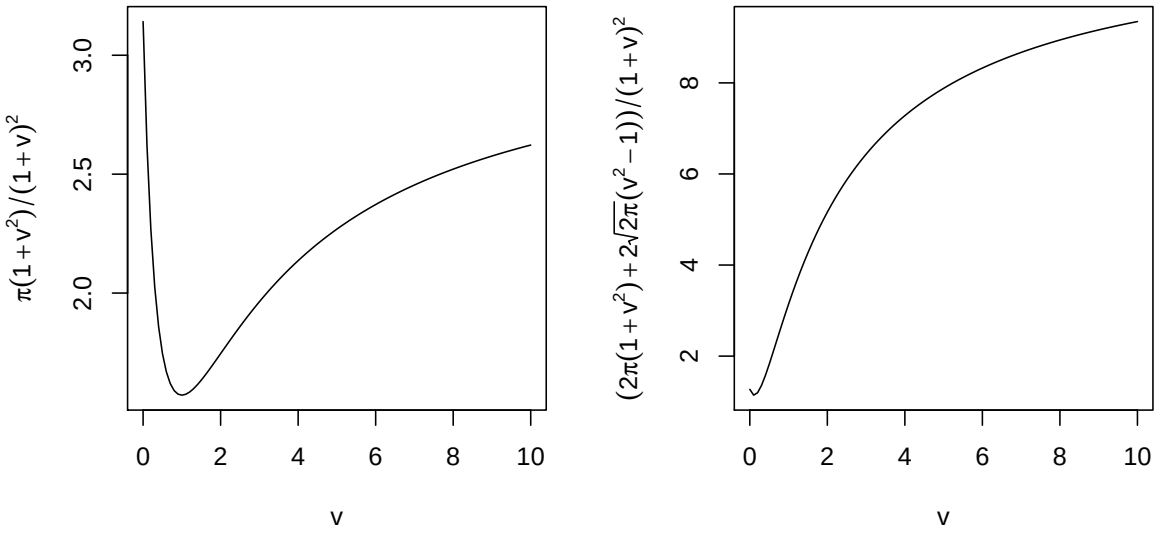

Figure 3.6: Plots of the asymptotic variances in (3.15).

3.5 Regularized Solution Path of the BSVM with respect to v

In this section, we discuss how to obtain the entire solution path efficiently with respect to v. Using this path, we can compare the performances of the BSVM with different values of v without additional computational burden. Hastie et al. (2004) established the entire regularization path for the SVM for every value of λ. In the BSVM procedure, we have two parameters to choose, λ and v, and here we derive an algorithm that fits the BSVM with respect to v for a fixed λ.

We first categorize the observations according to their relative positions to the hyperplane $f(\boldsymbol{x}) = \pm 1$. In particular, let $\mathcal{E} = \{i : y_i f(\boldsymbol{x}_i) = 1\}$, $\mathcal{L} = \{i : y_i f(\boldsymbol{x}_i) < 1\}$, and $\mathcal{R} = \{i : y_i f(\boldsymbol{x}_i) > 1\}$. From (3.9) -(3.11), notice that

$$\text{For any } i \in \mathcal{L}, \quad \gamma_i = C, \delta_i = 0, \text{ thus } \alpha_i = C \tag{3.16}$$

$$\text{For any } i \in \mathcal{R}, \quad \gamma_i = 0, \delta_i = C, \text{ thus } \alpha_i = -Cv \tag{3.17}$$

$$\text{For any } i \in \mathcal{E}, \quad \alpha_i \text{ can be any number in } [-Cv, C]. \tag{3.18}$$

For a fixed C, we start with a sufficiently large v which induces $y_i f(\boldsymbol{x}_i) \leqslant 1, \forall i = 1, \cdots, n$,

and go down to a smaller v. As the value of v decreases, the memberships of $\mathcal{E}, \mathcal{L}$, and $\mathcal{R}$ change. We say that an *event* occurred when any point changes its membership. There are three kinds of events:

E1. A point from $\mathcal{L}$ has just entered $\mathcal{E}$.

E2. A point from $\mathcal{R}$ has just entered $\mathcal{E}$.

E3. One or more points from $\mathcal{E}$ has entered either $\mathcal{L}$ or $\mathcal{R}$.

Once an event occurs, the sets $\mathcal{E}$, $\mathcal{L}$, and $\mathcal{R}$ will stay stable for a while until the next event occurs. This is because, for an observation to pass through $\mathcal{E}$, its α_i must change from C to $-Cv$ or vice versa. Therefore, we denote by v_1 our starting point, and let $v_2 > v_3 > \cdots$ be the values of v at which each of the events occurs.

Given v_l, we next study how to obtain v_{l+1}, and establish paths of α_i for $v \in [v_l, v_{l+1}]$. Let $\tau_i = \alpha_i/v = (\gamma_i - v\delta_i)/v$ for $i = 1, \cdots, n$ and $\tau_0 = b/v$. We use superscript or subscript l to denote anything given $v = v_l$. For now, we assume $\mathcal{E}^l \neq \emptyset$. For $v_l > v > v_{l+1}$, we have

$$\begin{aligned} f(\boldsymbol{x}) &= f(\boldsymbol{x}) - \frac{v}{v_l} f^l(\boldsymbol{x}) + \frac{v}{v_l} f^l(\boldsymbol{x}) \\ &= v\left[\sum_{j=1}^{n} \tau_j y_j \boldsymbol{x}_j^T \boldsymbol{x} + \tau_0 - \tau_j^l y_j \boldsymbol{x}_j^T \boldsymbol{x} - \tau_0^l + \frac{1}{v_l} f^l(\boldsymbol{x})\right] \\ &= v\left[\sum_{j=1}^{n} (\tau_j - \tau_j^l) y_j \boldsymbol{x}_j^T \boldsymbol{x} + (\tau_0 - \tau_0^l) + \frac{1}{v_l} f^l(\boldsymbol{x})\right] \\ &= v\left[C\left(\frac{1}{v} - \frac{1}{v_l}\right) \sum_{j \in \mathcal{L}^l} y_j \boldsymbol{x}_j^T \boldsymbol{x} + \sum_{j \in \mathcal{E}^l} (\tau_j - \tau_j^l) y_j \boldsymbol{x}_j^T \boldsymbol{x} + (\tau_0 - \tau_0^l) + \frac{1}{v_l} f^l(\boldsymbol{x})\right]. \quad (3.19) \end{aligned}$$

The last equality in (3.19) follows from the fact that $\tau_j - \tau_j^l = C(\frac{1}{v} - \frac{1}{v_l})$ for $j \in \mathcal{L}^l$ and $\tau_j - \tau_j^l = 0$ for $j \in \mathcal{R}^l$. Thus, for $i \in \mathcal{E}^l$,

$$\frac{1}{v} = \frac{1}{v} y_i f(\boldsymbol{x}_i) = C\left(\frac{1}{v} - \frac{1}{v_l}\right) \sum_{j \in \mathcal{L}^l} y_i y_j \boldsymbol{x}_j^T \boldsymbol{x}_i + \sum_{j \in \mathcal{E}^l} (\tau_j - \tau_j^l) y_i y_j \boldsymbol{x}_j^T \boldsymbol{x}_i + y_i(\tau_0 - \tau_0^l) + \frac{1}{v_l}.$$

Writing $\kappa_j = \tau_j - \tau_j^l$ for $j \in \{0\} \cup \mathcal{E}^l$, we have

$$\sum_{j \in \mathcal{E}^l} \kappa_j y_i y_j \boldsymbol{x}_j^T \boldsymbol{x}_i + y_i \kappa_0 = \left(\frac{1}{v} - \frac{1}{v_l}\right)\left[1 - C \sum_{j \in \mathcal{L}^l} y_i y_j \boldsymbol{x}_j^T \boldsymbol{x}_i\right]. \quad (3.20)$$

Let m be the number of points in $\mathcal{E}^l$. We can rewrite (3.20) in a matrix form

$$\boldsymbol{K}_l\boldsymbol{\kappa} + \kappa_0\boldsymbol{y}_l = \left(\frac{1}{v} - \frac{1}{v_l}\right)\boldsymbol{d}_l,$$

where $\boldsymbol{K}_l$ is the $m \times m$ matrix with ij-th entry $y_i y_j \boldsymbol{x}_j^T \boldsymbol{x}_i$ for $i, j \in \mathcal{E}^l$, and $\boldsymbol{\kappa}$, $\boldsymbol{y}_l$, and $\boldsymbol{d}_l$ are the $m \times 1$ matrices with i-th entry κ_i, y_i, and $1 - C\sum_{j\in\mathcal{L}^l} y_i y_j \boldsymbol{x}_j^T \boldsymbol{x}_i$ for $i \in \mathcal{E}^l$, repectively.

From (3.8), we have $\sum_{j=1}^n \tau_i y_i = 0$. Thus,

$$0 = \sum_{j=1}^{n}(\tau_j - \tau_j^l)y_j = \sum_{j\in\mathcal{E}^l} \kappa_j y_j + C\left(\frac{1}{v} - \frac{1}{v_l}\right)\sum_{j\in\mathcal{L}^l} y_j. \tag{3.21}$$

Using the matrix form, we have

$$\boldsymbol{y}_l^T\boldsymbol{\kappa} = -C\left(\frac{1}{v} - \frac{1}{v_l}\right)\sum_{j\in\mathcal{L}^l} y_j. \tag{3.22}$$

Combining (3.21) and (3.22), we have the linear equations

$$\boldsymbol{A}_l\boldsymbol{\kappa}^* = \left(\frac{1}{v} - \frac{1}{v_l}\right)\boldsymbol{d}_l^*,$$

where

$$\boldsymbol{A}_l = \begin{pmatrix} 0 & \boldsymbol{y}_l^T \\ \boldsymbol{y}_l & \boldsymbol{K}_l \end{pmatrix}, \quad \boldsymbol{\kappa}^* = \begin{pmatrix} \kappa_0 \\ \boldsymbol{\kappa} \end{pmatrix}, \quad \boldsymbol{d}_l^* = \begin{pmatrix} -C\sum_{j\in\mathcal{L}^l} y_j \\ \boldsymbol{d}_l \end{pmatrix}.$$

Define $\boldsymbol{s}_l = \boldsymbol{A}_l^{-1}\boldsymbol{d}_l^*$, and denote its entries by s_j for $j \in \mathcal{E}^l$, then we have

$$\boldsymbol{\kappa}^* = \left(\frac{1}{v} - \frac{1}{v_l}\right)\boldsymbol{s}_l \quad \text{for } j \in \{0\} \cup \mathcal{E}^l, \tag{3.23}$$

which implies

$$\alpha_j = \left(\frac{\alpha_j^l - s_j^l}{v_l}\right)v + s_j^l \quad \text{for } j \in \mathcal{E}^l \tag{3.24}$$

$$b = \left(\frac{b^l - s_0^l}{v_l}\right)v + s_0^l. \tag{3.25}$$

Hence, α_j and b are piecewise linear in v.

Combining (3.19) and (3.23) gives

$$f(\boldsymbol{x}) = \frac{v}{v_l} f^l(\boldsymbol{x}) + vC\left(\frac{1}{v} - \frac{1}{v_l}\right) \sum_{j \in \mathcal{L}^l} y_j \boldsymbol{x}_j^T \boldsymbol{x} + \sum_{j \in \mathcal{E}^l} s_j^l y_j \boldsymbol{x}_j^T \boldsymbol{x} + b_0^l - \frac{v}{v_l}\left[\sum_{j \in \mathcal{E}^l} s_j^l y_j \boldsymbol{x}_j^T \boldsymbol{x} + b_0^l\right]. \quad (3.26)$$

Writing $h^l(\boldsymbol{x}) = \sum_{j \in \mathcal{E}^l} s_j^l y_j \boldsymbol{x}_j^T \boldsymbol{x} + b_0^l$, we have

$$f(\boldsymbol{x}) = \frac{v}{v_l}\left[f^l(\boldsymbol{x}) - h^l(\boldsymbol{x})\right] + h^l(\boldsymbol{x}) + vC\left(\frac{1}{v} - \frac{1}{v_l}\right) \sum_{j \in \mathcal{L}^l} y_j \boldsymbol{x}_j^T \boldsymbol{x}. \quad (3.27)$$

The path (3.24)-(3.27) continues until one of the following occurs.

P1. One of the observations in $\mathcal{L}^l$ or $\mathcal{R}^l$ attains $y_i f(\boldsymbol{x}_i) = 1$.

P2. One of the α_i for $i \in \mathcal{E}^l$ reaches a boundary ($-Cv$ or C).

Note that **P1** implies the event **E1** or **E2**, and **P2** precedes **E3** or they coincide. Hence, we can obtain v_{l+1} by choosing the largest $v < v_l$ which incudes for which any of **P1** or **P2** occurs. Since $f(\boldsymbol{x}_i) = 1/y_i = y_i$ when **P1** happens, from (3.27), we have

$$v_l y_i = v[f^l(\boldsymbol{x}) - h^l(\boldsymbol{x})] + v_l h^l(\boldsymbol{x}) + v_l C \sum_{j \in \mathcal{L}^l} y_j \boldsymbol{x}_j^T \boldsymbol{x} - vC \sum_{j \in \mathcal{L}^l} y_j \boldsymbol{x}_j^T \boldsymbol{x}.$$

Thus, v for which **P1** happens is

$$v = \frac{v_l y_i - v_l h^l(\boldsymbol{x}) - v_l C \sum_{j \in \mathcal{L}^l} y_j \boldsymbol{x}_j^T \boldsymbol{x}}{f^l(\boldsymbol{x}) - h^l(\boldsymbol{x}) - C \sum_{j \in \mathcal{L}^l} y_j \boldsymbol{x}_j^T \boldsymbol{x}}. \quad (3.28)$$

Furthermore, for **P2** to happens, either $\alpha_i = -Cv$ or $\alpha_i = C$ should happen. From (3.24), this implies

$$v = \frac{v_l s_i^l}{s_i^l - C v_l - \alpha_i^l} \quad (3.29)$$

or

$$v = \frac{v_l (C - s_i^l)}{a_i^l - s_i^l}. \quad (3.30)$$

Hence, given v_l, we compute (3.28), (3.29), and (3.30), then set the largest v among the ones smaller than v_l as v_{l+1}. For $v \in (v_{l+1}, v_l)$, the solutions are calculated by (3.24), (3.25), and (3.27). We repeat this procedure until v runs all the way down to zero to obtain the whole

solution path for every value of v.

So far we assume $\mathcal{E}$ is nonempty. It is a reasonable assumption since we can force $\mathcal{E}$ to be nonempty, by selecting a good b. This is possible because b is not uniquely determined when $\mathcal{E}$ is empty. More specifically, suppose $\mathcal{E} = \emptyset$ for $v \in [v_0 - \epsilon, v_0]$, with $\epsilon > 0$. By (3.8), (3.16), and (3.17), we have

$$0 = \sum_{i=1}^{n} (\gamma_i - v\delta_i) y_i = c \sum_{i \in \mathcal{L}} y_i - Cv \sum_{i \in \mathcal{R}} y_i,$$

for $v \in [v_0 - \epsilon, v_0]$. Thus, we have

$$\sum_{i \in \mathcal{L}} y_i = \sum_{i \in \mathcal{R}} y_i = 0.$$

Now consider the objective function. Solving (1.1) with $g(u)$ in (3.4) is equivalent to minimizing

$$\begin{aligned} & \frac{1}{2}\|\boldsymbol{w}\|^2 + C[\sum_{i \in \mathcal{L}} (1 - y_i f(\boldsymbol{x}_i)) + \sum_{i \in \mathcal{R}} v(y_i f(\boldsymbol{x}_i) - 1)] \\ = \; & \frac{1}{2}\|\boldsymbol{w}\|^2 + C[c_L - vc_R - \sum_{i \in \mathcal{L}} y_i \boldsymbol{x}_i^T \boldsymbol{w} + v \sum_{i \in \mathcal{R}} y_i \boldsymbol{x}_i^T \boldsymbol{w} + (-\sum_{i \in \mathcal{L}} y_i + v \sum_{i \in \mathcal{R}} y_i) b], \quad (3.31) \end{aligned}$$

where c_L and c_R are the number of entries in $\mathcal{L}$ and $\mathcal{R}$, respectively. Note that b in (3.31) vanishes because $-\sum_{i \in \mathcal{L}} y_i + v \sum_{i \in \mathcal{R}} y_i = 0$. Hence, given $\boldsymbol{w}$, minimizer b could be any value in the set B, where

$$B = \left\{ b \in \mathbb{R} : \frac{1}{2}\|\boldsymbol{w}\|^2 + C \sum_{i=1}^{n} g(y_i f(\boldsymbol{x}_i)) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \left[\sum_{i \in \mathcal{L}} (1 - y_i f(\boldsymbol{x}_i)) + \sum_{i \in \mathcal{R}} v(y_i f(\boldsymbol{x}_i) - 1) \right] \right\},$$

that is, b can take any value unless it moves any points from $\mathcal{L}$ to $\mathcal{R}$, or vice versa. Hence, we can take any b satisfying

$$\begin{aligned} y_i f(\boldsymbol{x}_i) &\leqslant 1 \quad \text{for } i \in \mathcal{L} \\ y_i f(\boldsymbol{x}_i) &\geqslant 1 \quad \text{for } i \in \mathcal{R}, \end{aligned}$$

which is equivalent to

$$\begin{aligned} b &\leqslant 1 - \boldsymbol{x}_i^T \boldsymbol{w} \quad \text{for } i \in \mathcal{L}_+ \\ b &\geqslant -1 - \boldsymbol{x}_i^T \boldsymbol{w} \quad \text{for } i \in \mathcal{L}_- \\ b &\geqslant 1 - \boldsymbol{x}_i^T \boldsymbol{w} \quad \text{for } i \in \mathcal{R}_+ \\ b &\leqslant -1 - \boldsymbol{x}_i^T \boldsymbol{w} \quad \text{for } i \in \mathcal{R}_-, \end{aligned}$$

where $\mathcal{L}_+ = \mathcal{L} \cap \{i : y_i = 1\}$, $\mathcal{L}_- = \mathcal{L} \cap \{i : y_i = -1\}$, $\mathcal{R}_+ = \mathcal{R} \cap \{i : y_i = 1\}$, and $\mathcal{R}_- = \mathcal{R} \cap \{i : y_i = -1\}$. Letting

$$\begin{aligned} i_{L+} &= \arg\max_{i \in \mathcal{L}_+} \boldsymbol{x}_i^T \boldsymbol{w} \\ i_{L-} &= \arg\min_{i \in \mathcal{L}_-} \boldsymbol{x}_i^T \boldsymbol{w} \\ i_{R+} &= \arg\min_{i \in \mathcal{R}_+} \boldsymbol{x}_i^T \boldsymbol{w} \\ i_{R-} &= \arg\max_{i \in \mathcal{R}_-} \boldsymbol{x}_i^T \boldsymbol{w}, \end{aligned}$$

we have

$$\max\{-1 - \boldsymbol{x}_{i_{L-}}^T \boldsymbol{w}, 1 - \boldsymbol{x}_{i_{R+}}^T \boldsymbol{w}\} \leqslant b \leqslant \min\{1 - \boldsymbol{x}_{i_{L+}}^T \boldsymbol{w}, -1 - \boldsymbol{x}_{i_{R-}}^T \boldsymbol{w}\}.$$

Without loss of generality, we can assume $1 - \boldsymbol{x}_{i_{L+}}^T \boldsymbol{w} \leqslant -1 - \boldsymbol{x}_{i_{R-}}^T \boldsymbol{w}$. Then take $b = 1 - \boldsymbol{x}_{i_{L+}}^T \boldsymbol{w}$. This b belongs to B and we have $i_{L+} \in \mathcal{E}$. Consequently, we choose b that induces $\mathcal{E} \neq \emptyset$. Hence the case of empty $\mathcal{E}$ is resolved.

In summary, one can get the entire solution path for the BSVM with respect to v as follows:

Step 1. Start with a sufficiently large v_0 and let $v_l = v_0$.

Step 2. For v_l, obtain the solution of the BSVM. If $\mathcal{E}^l$ is empty, choose b as either upper or lower bound of (3.32) so that $\mathcal{E}^l$ becomes nonempty.

Step 3. Calculate (3.28), (3.29), and (3.30), then set the minimum of them as v_{l+1}, at which the next event happens.

Step 4. For $v \in (v_{l+1}, v_l)$, compute the path using (3.27).

Step 5. If $v_{l+1} \leqslant 0$, then set $v_{l+1} = 0$ and obtain the solution of the BSVM for $v_{l+1} = 0$ and stop. Otherwise, then set $v_l = v_{l+1}$ and go to **Step 2**.

3.6 Numerical Results

In this section, numerical studies are carried out to examine the performance of the BSVM, the BCM, and the RSVM (Wu and Liu, 2007). We note that the RSVM with truncation location at 0 is equivalent to Psi-learning (Liu et al., 2005).

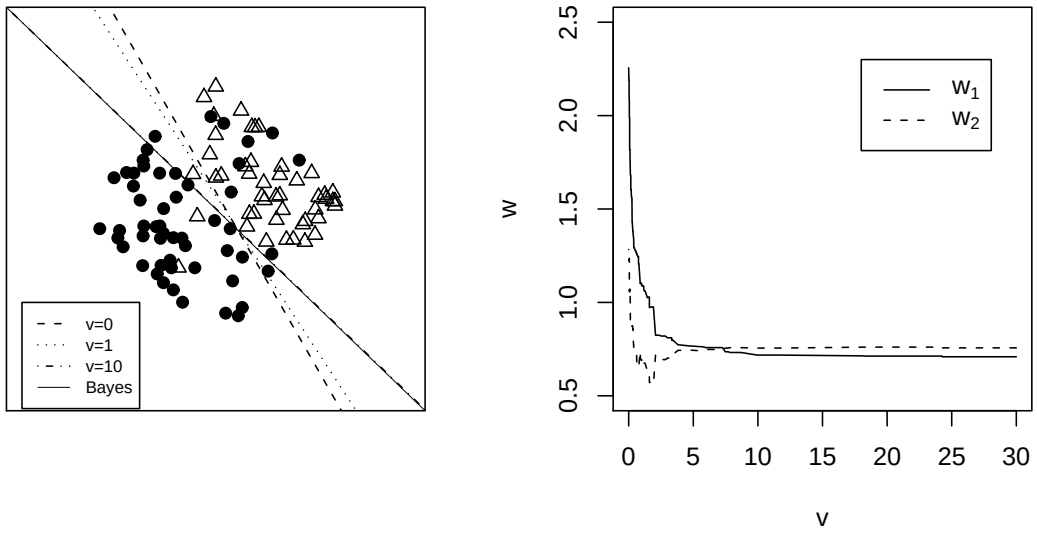

Figure 3.7: Left: Illustration of the data set in Example 3.6.1.1. Right: Illustration of the path of $\boldsymbol{w}$ with respect to v in Example 3.6.1.1.

3.6.1 Simulation

In two simulated data sets, we generate training sets, tuning sets, and testing sets with sample sizes 100, 100, and 10^6, respectively. For each value of $v = 0, 0.1, 0.2, 0.5, 1, 2, 5, 10, 50$, the tuning parameter λ is chosen by a grid search based on the tuning error. The misclassification rate is calculated based on the testing set to evaluate the performance. Each procedure is repeated for 100 times on 100 different training and tuning sets and the corresponding mean performance is reported.

Example 3.6.1.1 The data are generated as follows. First, (x_1, x_2) is sampled from a square $\{(x_1, x_2) : -\sqrt{2} < x_1 + x_2 < \sqrt{2}, -\sqrt{2} < x_1 - x_2 < \sqrt{2}\}$. Then, set $y = 1$ if $x_1 + x_2 > 0$ and $y = -1$ otherwise. To illustrate the effect of outliers, we randomly flip the class membership of 0%, 5%, and 10% of data. A typical example of training data set and the resulting BSVM boundaries are plotted in the left panel of Figure 3.7. The corresponding solution path of $\boldsymbol{w}$ is provided in the right panel of Figure 3.7. Interestingly, the solution doesn't change once the value v gets sufficiently large. Note that performance of the RSVM is pretty good as well

especially when there are outliers, but the BSVM with larger v works better.

Table 3.1: Testing errors of the simulated linear example (Example 3.6.1.1)

		Data contamination rates		
Method		0%	5%	10%
BSVM (with tuning set)	$v = 0$	0.0150(0.0101)	0.0730(0.0156)	0.1289(0.0212)
	$v = 0.1$	0.0239(0.0165)	0.0747(0.0169)	0.1295(0.0191)
	$v = 0.2$	0.0247(0.0162)	0.0753(0.0163)	0.1283(0.0183)
	$v = 0.5$	0.0243(0.0147)	0.0729(0.0138)	0.1254(0.0161)
	$v = 1$	0.0222(0.0128)	0.0707(0.0130)	0.1224(0.0148)
	$v = 2$	0.0186(0.0113)	0.0673(0.0107)	0.1176(0.0107)
	$v = 5$	0.0137(0.0080)	0.0620(0.0087)	0.1112(0.0072)
	$v = 10$	0.0107(0.0069)	0.0593(0.0066)	0.1091(0.0069)
	$v = 50$	0.0100(0.0073)	0.0586(0.0059)	0.1080(0.0062)
BCM		0.0095(0.0066)	0.0576(0.0053)	0.1079(0.0062)
RSVM	$s = -1$	0.0150(0.0103)	0.0649(0.0099)	0.1169(0.0136)
	$s = 0$	0.0161(0.0110)	0.0700(0.0136)	0.1225(0.0154)
Bayes Error		0.00	0.05	0.10

Test error results are summarized in Table 3.1. Regarding to the effect of v, the higher v produces the better result. This is not surprising because of the structure of this data set. Because the data points are aligned quite parallel to the true boundary, the observations far from the boundary reflects the overall structure of the data set, resulting in favor to the BSVM with high v which uses a lot of information from those data far from the boundary. As the limit of the BSVM, the BCM gives the best performance in this example. Notice that the RSVM works reasonably well for this example.

Example 3.6.1.2 We generate equal numbers of data points for class 1 and class -1. For class 1, 40%, 40%, and 20% of the observations are generated from $N((1, 0.5)^T, \sigma^2 I)$, $N((-3, 0.5)^T, \sigma^2 I)$, and $N((0, 1)^T, \Sigma)$, respectively, where I is 2×2 identity matrix and $\Sigma = \text{diag}((4\sigma)^2, (\sigma/3)^2)$. For class 2, 40%, 40%, and 20% of the observations are generated from $N((3, -0.5)^T, \sigma^2 I)$, $N((-1, -0.5)^T, \sigma^2 I)$, and $N((0, -1)^T, \Sigma)$. We use two different values of σ, 0.3 and 0.5, and a typical example of data sets when $\sigma = 0.3$ is plotted in Figure 3.5. As shown in Table 3.2, the result seems the opposite to the Example 3.6.1.1: the smaller v gives the better result. This is not surprising considering the nature of this data set. Since the information about observations near the boundary is critical for classification in this data set, it is better to use more information

Table 3.2: Testing errors of the simulated nonlinear example (Example 3.6.1.2)

		Standard deviation	
Method		$\sigma = 0.3$	$\sigma = 0.5$
BSVM (with tuning set)	$v = 0$	0.0052(0.0046)	0.0574(0.0177)
	$v = 0.1$	0.0055(0.0048)	0.0695(0.0212)
	$v = 0.2$	0.0060(0.0054)	0.0749(0.0197)
	$v = 0.5$	0.0083(0.0059)	0.0857(0.0176)
	$v = 1$	0.0107(0.0060)	0.0954(0.0148)
	$v = 2$	0.0150(0.0075)	0.1073(0.0163)
	$v = 5$	0.0233(0.0100)	0.1164(0.0128)
	$v = 10$	0.0265(0.0108)	0.1212(0.0131)
	$v = 50$	0.0288(0.0097)	0.1231(0.0139)
BCM		0.0267(0.0114)	0.1214(0.0174)
RSVM	$s = -1$	0.0052(0.0045)	0.0528(0.0126)
	$s = 0$	0.0039(0.0018)	0.0517(0.0121)
Bayes Error		0.000159	0.022104

about those observations. If we use higher v, the data far from the boundary pull the decision boundary resulting in a flat decision boundary which does not reflect well the data structure around the boundary.

3.6.2 Real Data

Table 3.3: Testing errors of the lung cancer data example in Section 3.6.2.

Method		Testing errors
BSVM	$v = 0$	0.0203(0.0170)
	$v = 0.1$	0.0174(0.0178)
	$v = 0.2$	0.0145(0.0181)
	$v = 0.5$	0.0145(0.0181)
	$v = 1$	0.0145(0.0181)
	$v = 2$	0.0145(0.0181)
	$v = 5$	0.0145(0.0181)
	$v = 10$	0.0145(0.0181)
	$v = 50$	0.0145(0.0181)
BCM		0.0145(0.0181)
RSVM	$s = -1$	0.0203(0.0170)
	$s = 0$	0.0203(0.0170)

In this section, we apply the BSVM and the BCM to the lung cancer data described in Liu

et al. (2008). In this data set, there are 12,625 genes' expression of 17 normal tissues and 188 lung cancer tissues. We first filter the genes using the ratio of the sample standard deviation and sample mean of each gene and obtain 316 genes. Then, we standardize gene expression so that each gene has sample mean 0 and sample standard deviation 1. We randomly divide subjects into three groups of training, tuning, and testing sets with sample size 68, 68, and 69, and we build a model for each value of λ using the data in training set. Then λ is selected based on it's performance on tuning set by grid search. Using the model with the selected λ, misclassification rate on testing set is calculated. This whole procedure is repeated for 10 times.

The results are reported in Table 3.3. As shown in the table, the BSVM indeed performs better than the standard SVM, while the RSVM does not really improve the performance comparing to the SVM. Hence, we can conclude that, by using information of correctly classified data, the BSVM does obtain robustness which could not be achieved by bounding the effect of extreme outliers in this situation. This may be due to the nature of this data.

3.7 Remark and Possible Future Work

Our results indicate that the choice of v is indeed important for the performance of the BSVM. Although one may treat v as a tuning parameter, it will be more desirable to have a more efficient approach to select v. One possibility is to derive the GACV curve with respect to v and choose the value of v which minimizes the GACV.

The BCM has a nice interpretation and performs well in many situations. However, its linear loss function may emphasize too much on the correctly classified observations comparing to wrongly classified observations. Hence, one can consider to modify the loss function form of the BCM to reduce the loss imposed on correctly classified data. In particular, we consider a fimily of the BCM loss function

$$L(u) = 2[(1-u)/2]^a.$$

As shown in Figure 3.8, this loss function becomes equivalent to the original BCM loss function when $a = 1$, and as the value of a increases, the difference in loss between the correctly classified and wrongly classified observations becomes large. It will be interesting to investigate the following issues:

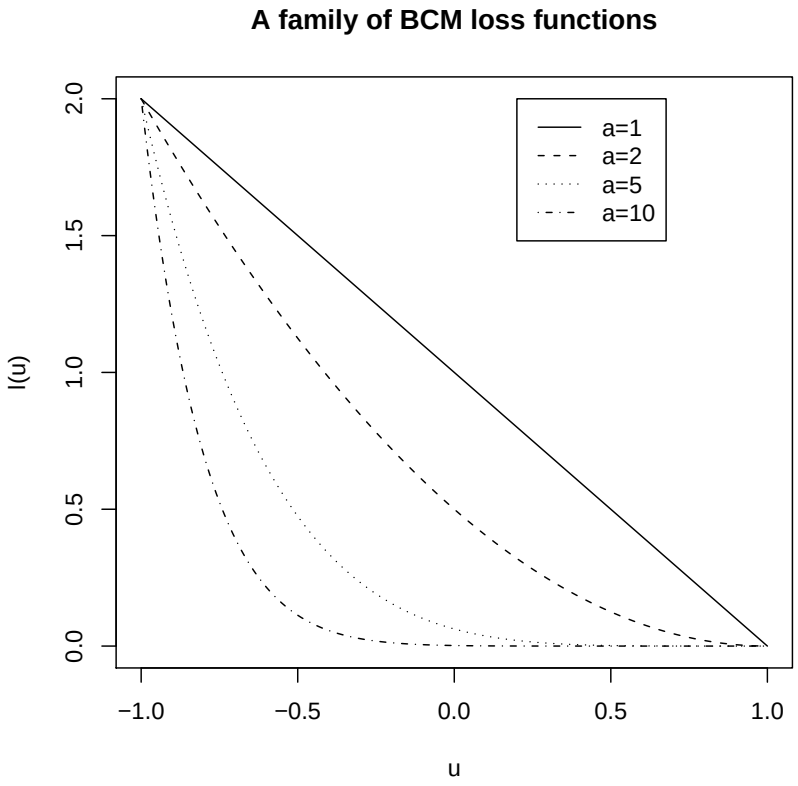

Figure 3.8: Plot of several BCM loss functions indexed by a.

- The effect of a on the classification performance;
- The choice of constraints;
- Fisher consistency behaviors.

3.8 Proofs

3.8.1 Proof of Theorem 3

Let $f = f(\boldsymbol{x})$, $p = P(\boldsymbol{x})$, and $A(f) = E[g(Yf(\boldsymbol{X})|\boldsymbol{X} = \boldsymbol{x}]$. First, we show that the minimizer f^* of $A(f)$ is on $[-1, 1]$. When $f > 1$, $A(f) = pv(f-1) + (1-p)(1+f) > 2(1-p) = A(1)$. Similarly, when $f < -1$, $A(f) = p(1-f) + (1-p)v(-f-1) > 2p = A(-1)$. Thus, $f^* \in [-1, 1]$. For $f \in [-1, 1]$, $A(f) = p(1-f) + (1-p)(1+f) = (1-2p)f + 1$. Hence $f = 1$ minimizes $A(f)$ if $p > 1/2$, and otherwise, $f = -1$ minimizes $A(f)$. Therefore, $\text{argmin}_f A(f) = \text{sign}[p - 1/2]$. This completes the proof. □

3.8.2 Proof of Theorem 4

It is easy to see that $f_l \leqslant k - 1$ for $l = 1, \cdots, k$. Thus, one can show that the problem reduces to

$$\begin{aligned} &\max_{\boldsymbol{f}} \sum_{l=1}^{k} P_l(\boldsymbol{x}) f_l(\boldsymbol{x}) \\ \text{s.t.} \quad &\sum_{l=1}^{k} f_l(\boldsymbol{x}) = 0; -1 \leqslant f_l(\boldsymbol{x}) \leqslant k-1, \forall l. \end{aligned} \tag{3.32}$$

Thus, the solution satisfies $f_j^*(\boldsymbol{x}) = k - 1$ if $j = \text{argmax}_j P_j(\boldsymbol{x})$ and -1 otherwise. □

3.8.3 Proof of Theorem 5 and Theorem 6

First we go over lemmas we need to prove the theorems. Lemma 2 guarantees that there is a finite minimizer of $Q(\boldsymbol{\beta})$.

Lemma 2. *Suppose that* ***A1*** *and* ***A2*** *are satisfied. Then* $Q(\boldsymbol{\beta}) \to \infty$ *as* $\|\boldsymbol{\beta}\| \to \infty$ *and the minimizer* $\boldsymbol{\beta}^*$ *exists.*

Proof. Without loss of generality, we can assume that $\boldsymbol{x}_0 = 0$ and $B(\boldsymbol{x}_0, r_0) \subset \mathcal{X}$. Then, for any $\epsilon > 0$, we have

$$\begin{aligned}
Q(\boldsymbol{\beta}) = \; & E[|g(Yf(\boldsymbol{X};\boldsymbol{\beta}))|] \\
& \geqslant \int_{\mathcal{X}} \min\{v,1\}|Yf(\boldsymbol{x};\boldsymbol{\beta}) - 1|(\pi_+ h_+(\boldsymbol{x}) + \pi_- h_-(\boldsymbol{x}))d\boldsymbol{x} \\
& \geqslant \min\{v,1\} \int_{\mathcal{X}} [|f(\boldsymbol{x};\boldsymbol{\beta})| - 1](\pi_+ h_+(\boldsymbol{x}) + \pi_- h_-(\boldsymbol{x}))d\boldsymbol{x} \\
& = \min\{v,1\} \left[\int_{\mathcal{X}} |f(\boldsymbol{x};\boldsymbol{\beta})|(\pi_+ h_+(\boldsymbol{x}) + \pi_- h_-(\boldsymbol{x}))d\boldsymbol{x} - 1 \right] \\
& = \min\{v,1\}\|\boldsymbol{\beta}\| \int_{\mathcal{X}} |f(\boldsymbol{x};\boldsymbol{\beta})|(\pi_+ h_+(\boldsymbol{x}) + \pi_- h_-(\boldsymbol{x}))d\boldsymbol{x} - \min\{v,1\} \\
& = C_1 \min\{v,1\}\|\boldsymbol{\beta}\| \int_B |f(\boldsymbol{x};\boldsymbol{\beta})|d\boldsymbol{x} - \min\{v,1\} \\
& = C_1 \min\{v,1\}\|\boldsymbol{\beta}\| \mathrm{vol}(\{|w_0 + \boldsymbol{x}^T\boldsymbol{w}_+| \geqslant \epsilon\} \cap \{B(0,r_0)\})\epsilon - \min\{v,1\},
\end{aligned}$$

where $\boldsymbol{w} = (w_0, \boldsymbol{w}_+)^T = \boldsymbol{\beta}/\|\boldsymbol{\beta}\|$ and $\mathrm{vol}(A)$ denotes the volume of a set A. Observe that $-1 \leqslant w_0 \leqslant 1$. For $0 \leqslant w_0 < 1$, if we take $\epsilon \in (0,1)$,

$$\begin{aligned}
& \mathrm{vol}(\{\boldsymbol{x} \in \mathcal{X} : |w_0 + \boldsymbol{x}^T\boldsymbol{w}_+| \geqslant \epsilon\} \cap \{B(0,r_0)\}) \\
& \geqslant \mathrm{vol}(\{\boldsymbol{x} \in \mathcal{X} : w_0 + \boldsymbol{x}^T\boldsymbol{w}_+ \geqslant \epsilon\} \cap \{B(0,r_0)\}) \\
& = \mathrm{vol}\left(\left\{ \boldsymbol{x} \in \mathcal{X} : \tfrac{\boldsymbol{x}^T\boldsymbol{w}_+}{\sqrt{1-w_0^2}} \geqslant \tfrac{\epsilon - w_0}{\sqrt{1-w_0^2}} \right\} \cap B(0,r_0) \right) \\
& \geqslant \mathrm{vol}\left(\left\{ \boldsymbol{x} \in \mathcal{X} : \tfrac{\boldsymbol{x}^T\boldsymbol{w}_+}{\sqrt{1-w_0^2}} \geqslant \epsilon \right\} \cap B(0,r_0) \right) \\
& \equiv V(r_0, \epsilon).
\end{aligned}$$

Similarly, we can show that $\mathrm{vol}(\{\boldsymbol{x} \in \mathcal{X} : |w_0 + \boldsymbol{x}^T\boldsymbol{w}_+| \geqslant \epsilon\} \cap \{B(0,r_0)\}) \geqslant V(r_0,\epsilon)$ when $-1 < w_0 < 0$. Since $V(r_0,\epsilon)$ is independent of $\boldsymbol{\beta}$ and $V(r_0,\epsilon) > 0$ for some $\epsilon < r_0$, we can conclude that $Q(\boldsymbol{\beta}) \to \infty$ as $\|\boldsymbol{\beta}\| \to \infty$

Furthermore, $Q(\boldsymbol{\beta})$ is convex because the loss function $g(yf(\boldsymbol{x};\boldsymbol{\beta}))$ is convex in $\boldsymbol{\beta}$. Using the fact that $Q(\boldsymbol{\beta}) \to \infty$ as $\|\boldsymbol{\beta}\| \to \infty$, the set of minimizers of $Q(\boldsymbol{\beta})$ is a bounded connected set. Thus, the minimizer $\boldsymbol{\beta}^*$ exists. □

Lemma 3 and 4 establishes $s(\boldsymbol{\beta})$ and $H(\boldsymbol{\beta})$, which are considered first and second derivatives of $Q(\boldsymbol{\beta})$, repectively.

Lemma 3. *Suppose that* ***A1*** *is satisfied. If* $\boldsymbol{\beta}_+ \neq \mathbf{0}$*, then*

$$\frac{\partial Q(\boldsymbol{\beta})}{\partial \beta_j} = S(\boldsymbol{\beta})_j$$

for $j = 0, \cdots, d$.

Proof. Define $\Delta(t) = g(f(\boldsymbol{x};\boldsymbol{\beta}) + t\tilde{\boldsymbol{x}}_j) - g(f(\boldsymbol{x};\boldsymbol{\beta}))$ for $t > 0$. When $\tilde{\boldsymbol{x}}_j > 0$, we have

$$\Delta(t) = \begin{cases} vt\tilde{x}_j & \text{if } f(\boldsymbol{x};\boldsymbol{\beta}) > 1 \\ (1+v)(f(\boldsymbol{x};\boldsymbol{\beta}) - 1) + vt\tilde{x}_j & \text{if } 1 - t\tilde{x}_j < f(\boldsymbol{x};\boldsymbol{\beta}) \leqslant 1 \\ -t\tilde{x}_j & \text{otherwise.} \end{cases}$$

Thus,

$$\begin{aligned} \frac{1}{t}\int_{\mathcal{X}} \Delta(t) I_{\{\tilde{x}_j > 0\}} h_+(\boldsymbol{x}) d\boldsymbol{x} &= v \int_{\mathcal{X}} I_{\{f(\boldsymbol{x};\boldsymbol{\beta}) > 1, \tilde{x}_j > 0\}} \tilde{x}_j h_+(\boldsymbol{x}) d\boldsymbol{x} \\ &\quad + \left(\tfrac{1+v}{t}\right) \int_{\mathcal{X}} I_{\{1 - t\tilde{x}_j < f(\boldsymbol{x};\boldsymbol{\beta}) \leqslant 1, \tilde{x}_j > 0\}} (f(\boldsymbol{x};\boldsymbol{\beta}) - 1) h_+(\boldsymbol{x}) d\boldsymbol{x} \\ &\quad + v \int_{\mathcal{X}} I_{\{1 - t\tilde{x}_j < f(\boldsymbol{x};\boldsymbol{\beta}) \leqslant 1, \tilde{x}_j > 0\}} \tilde{x}_j h_+(\boldsymbol{x}) d\boldsymbol{x} \\ &\quad + \int_{\mathcal{X}} I_{\{f(\boldsymbol{x};\boldsymbol{\beta}) \leqslant 1 - t\tilde{x}_j, \tilde{x}_j > 0\}} (-\tilde{x}_j) h_+(\boldsymbol{x}) d\boldsymbol{x} \end{aligned}$$

Here, consider the second term of the right hand side of the equation. Since $1 - t\tilde{x}_j < f(\boldsymbol{x};\boldsymbol{\beta}) \leqslant 1$ implies $|f(\boldsymbol{x};\boldsymbol{\beta}) - 1| \leqslant t\tilde{x}_j$, we have

$$\begin{aligned} &\left(\tfrac{1+v}{t}\right) \int_{\mathcal{X}} I_{\{1 - t\tilde{x}_j < f(\boldsymbol{x};\boldsymbol{\beta}) \leqslant 1, \tilde{x}_j > 0\}} (f(\boldsymbol{x};\boldsymbol{\beta}) - 1) h_+(\boldsymbol{x}) d\boldsymbol{x} \\ &\leqslant \left(\tfrac{1+v}{t}\right) \int_{\mathcal{X}} I_{\{1 - t\tilde{x}_j < f(\boldsymbol{x};\boldsymbol{\beta}) \leqslant 1, \tilde{x}_j > 0\}} |f(\boldsymbol{x};\boldsymbol{\beta}) - 1| h_+(\boldsymbol{x}) d\boldsymbol{x} \\ &\leqslant (1+v) \int_{\mathcal{X}} I_{\{1 - t\tilde{x}_j < f(\boldsymbol{x};\boldsymbol{\beta}) \leqslant 1, \tilde{x}_j > 0\}} \tilde{x}_j h_+(\boldsymbol{x}) d\boldsymbol{x} \\ &\to (1+v) \int_{\mathcal{X}} I_{\{f(\boldsymbol{x};\boldsymbol{\beta}) = 1, \tilde{x}_j > 0\}} \tilde{x}_j h_+(\boldsymbol{x}) d\boldsymbol{x} = 0 \quad \text{as } t \downarrow 0 \end{aligned}$$

by Dominated Convergence Theorem. This gives,

$$\begin{aligned} &\lim_{t \downarrow 0} \frac{1}{t} \int_{\mathcal{X}} \Delta(t) I_{\{\tilde{x}_j > 0\}} h_+(\boldsymbol{x}) d\boldsymbol{x} \\ &= v \int_{\mathcal{X}} I_{\{f(\boldsymbol{x};\boldsymbol{\beta}) \geqslant 1, \tilde{x}_j > 0\}} \tilde{x}_j h_+(\boldsymbol{x}) d\boldsymbol{x} + \int_{\mathcal{X}} I_{\{f(\boldsymbol{x};\boldsymbol{\beta}) < 1, \tilde{x}_j > 0\}} (-\tilde{x}_j) h_+(\boldsymbol{x}) d\boldsymbol{x}, \end{aligned}$$

resulting

$$\frac{\partial}{\partial \boldsymbol{\beta}_j}\int_{\mathcal{X}} g(f(\boldsymbol{x};\boldsymbol{\beta}))h_+(\boldsymbol{x})d\boldsymbol{x}$$
$$= v\int_{\mathcal{X}} I_{\{f(\boldsymbol{x};\boldsymbol{\beta})\geqslant 1\}}\tilde{x}_j h_+(\boldsymbol{x})d\boldsymbol{x} + \int_{\mathcal{X}} I_{\{f(\boldsymbol{x};\boldsymbol{\beta})<1\}}(-\tilde{x}_j)h_+(\boldsymbol{x})d\boldsymbol{x}.$$

Now we consider the case when $\tilde{x}_j < 0$. Define

$$\Delta(t) = \begin{cases} vt\tilde{x}_j & \text{if } f(\boldsymbol{x};\boldsymbol{\beta}) > 1 - t\tilde{x}_j \\ (1+v)(1-f(\boldsymbol{x};\boldsymbol{\beta})) - t\tilde{x}_j & \text{if } 1 < f(\boldsymbol{x};\boldsymbol{\beta}) \leqslant 1 - t\tilde{x}_j \\ -t\tilde{x}_j & \text{otherwise.} \end{cases}$$

In a similar manner, we can show that

$$\lim_{t\downarrow 0}\frac{1}{t}\int_{\mathcal{X}} \Delta(t) I_{\{\tilde{x}_j<0\}}h_-(\boldsymbol{x})d\boldsymbol{x}$$
$$= v\int_{\mathcal{X}} I_{\{f(\boldsymbol{x};\boldsymbol{\beta})\geqslant 1,\tilde{x}_j<0\}}\tilde{x}_j h_-(\boldsymbol{x})d\boldsymbol{x} + \int_{\mathcal{X}} I_{\{f(\boldsymbol{x};\boldsymbol{\beta})<1,\tilde{x}_j<0\}}(-\tilde{x}_j)h_-(\boldsymbol{x})d\boldsymbol{x}.$$

Consequently, we have

$$\lim_{t\downarrow 0}\frac{1}{t}\int_{\mathcal{X}} \Delta(t) h_+(\boldsymbol{x})d\boldsymbol{x}$$
$$= v\int_{\mathcal{X}} I_{\{f(\boldsymbol{x};\boldsymbol{\beta})\geqslant 1\}}\tilde{x}_j h_+(\boldsymbol{x})d\boldsymbol{x} + \int_{\mathcal{X}} I_{\{f(\boldsymbol{x};\boldsymbol{\beta})<1\}}(-\tilde{x}_j)h_+(\boldsymbol{x})d\boldsymbol{x}.$$

Thus, we may write

$$\frac{\partial}{\partial \boldsymbol{\beta}_j}\int_{\mathcal{X}} g(-f(\boldsymbol{x};\boldsymbol{\beta}))h_-(\boldsymbol{x})d\boldsymbol{x}$$
$$= v\int_{\mathcal{X}} I_{\{f(\boldsymbol{x};\boldsymbol{\beta})\leqslant -1\}}(-\tilde{x}_j)h_-(\boldsymbol{x})d\boldsymbol{x} + \int_{\mathcal{X}} I_{\{f(\boldsymbol{x};\boldsymbol{\beta})>-1\}}\tilde{x}_j h_-(\boldsymbol{x})d\boldsymbol{x},$$

finally, giving

$$\frac{\partial Q(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}_j} = E[-\rho(1 - Yf(\boldsymbol{X};\boldsymbol{\beta}))Y\tilde{\boldsymbol{X}}_j + v\rho(Yf(\boldsymbol{X};\boldsymbol{\beta}) - 1)Y\tilde{\boldsymbol{X}}_j]$$

□

Lemma 4. *Suppose that **A1** is satisfied. If $\boldsymbol{\beta}_+ \neq \mathbf{0}$, then*

$$\frac{\partial^2 Q(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}_j \partial \boldsymbol{\beta}_k} = H(\boldsymbol{\beta})_{jk}$$

for $j, k = 0, \cdots, d$.

Proof. Let

$$\Phi(\boldsymbol{\beta}) = v \int_{\mathcal{X}} I_{\{f(\boldsymbol{x};\boldsymbol{\beta}) \geqslant 1\}} s(\boldsymbol{x}) d\boldsymbol{x} - \int_{\mathcal{X}} I_{\{f(\boldsymbol{x};\boldsymbol{\beta}) < 1\}} s(\boldsymbol{x}) d\boldsymbol{x}.$$

Then it suffices to show that

$$\begin{aligned} \frac{\partial \Phi(\boldsymbol{\beta})}{\partial \beta_0} &= (1+v) \int_{\mathcal{X}} \delta(1 - f(\boldsymbol{x};\boldsymbol{\beta})) s(\boldsymbol{x}) d\boldsymbol{x} \\ \frac{\partial \Phi(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}_j} &= (1+v) \int_{\mathcal{X}} \delta(1 - f(\boldsymbol{x};\boldsymbol{\beta})) \tilde{\boldsymbol{x}}_j s(\boldsymbol{x}) d\boldsymbol{x}. \end{aligned}$$

Define

$$\Psi(\boldsymbol{\beta}) = \int_{\mathcal{X}} I_{\{f(\boldsymbol{x};\boldsymbol{\beta}) < 1\}} s(\boldsymbol{x}) d\boldsymbol{x}.$$

Note that

$$\Phi(\boldsymbol{\beta}) = v \int_{\mathcal{X}} s(\boldsymbol{x}) d\boldsymbol{x} - (1+v)\Psi(\boldsymbol{\beta}).$$

Since the term $v \int_{\mathcal{X}} s(\boldsymbol{x}) d\boldsymbol{x}$ is independent of $\boldsymbol{\beta}$, we have

$$\frac{\partial \Phi(\boldsymbol{\beta})}{\partial \beta_j} = -(1+v) \frac{\partial \Psi(\boldsymbol{\beta})}{\partial \beta_j}.$$

From the Lemma 3 in Koo et al. (2008),

$$\frac{\partial \Psi(\boldsymbol{\beta})}{\partial \beta_0} = -\int_{\mathcal{X}} \delta(1 - f(\boldsymbol{x};\boldsymbol{\beta})) s(\boldsymbol{x}) d\boldsymbol{x}$$

and

$$\frac{\partial \Psi(\boldsymbol{\beta})}{\partial \beta_j} = -\int_{\mathcal{X}} \delta(1 - f(\boldsymbol{x};\boldsymbol{\beta})) \tilde{x}_j s(\boldsymbol{x}) d\boldsymbol{x}$$

for $j = 1, \cdots, d$. This completes the proof. □

Lemma 5. *Suppose that* ***A1*** *and* ***A3*** *are satisfied. Then* $\boldsymbol{\beta}_+^* \neq \mathbf{0}$.

Proof. Assume the first case of **A3**

$$\pi_+ \left\{ \int_{\mathcal{X}} (I_{\{\boldsymbol{x}_{i^*} \leqslant F_{i^*}^+\}} - v I_{\{\boldsymbol{x}_{i^*} > F_{i^*}^+\}}) x_{i^*} h_+(\boldsymbol{x}) d\boldsymbol{x} \right\} > \pi_- \left\{ \int_{\mathcal{X}} (I_{\{x_{i^*} \geqslant G_{i^*}^-\}} - v I_{\{x_{i^*} < G_{i^*}^-\}}) x_{i^*} h_-(\boldsymbol{x}) d\boldsymbol{x} \right\}.$$

It is sufficient to show that

$$\min_{\beta_0} Q(\beta_0, 0, \cdots, 0) > \min_{\beta_0, \beta_{i^*}} Q(\beta_0, 0, \cdots, 0, \beta_{i^*}, 0, \cdots, 0). \tag{3.33}$$

We may write

$$Q(\beta_0, \beta_{i^*}) = \pi_+ \int_{\mathcal{X}} g(\beta_0 + \beta_{i^*} x_{i^*}) h_+(\boldsymbol{x}) d\boldsymbol{x} + \pi_- \int_{\mathcal{X}} g(-\beta_0 - \beta_{i^*} x_{i^*}) h_-(\boldsymbol{x}) d\boldsymbol{x}$$

First, consider the case that $\beta_{i^*} = 0$. We can show that

$$Q(\beta_0) = \begin{cases} (\pi_- - v\pi_+) + \beta_0(v\pi_+ + \pi_-) & \text{if } \beta_0 > 1 \\ 1 + \beta_0(\pi_- - \pi_+) & \text{if } -1 \leqslant \beta_0 \leqslant 1 \\ (\pi_+ - v\pi_-) - \beta_0(v\pi_- + \pi_+) & \text{otherwise,} \end{cases}$$

resulting

$$\min_{\beta_0} Q(\beta_0) = 2 \min\{\pi_+, \pi_-\}$$

.

Now, consider the case that $\beta_{i^*} > 0$. Then we have

$$\begin{aligned} Q(\beta_0, \beta_{i^*}) &= \pi_+ \int_{\mathcal{X}} I_{\left\{x_{i^*} < \frac{1-\beta_0}{\beta_{i^*}}\right\}} (1 - \beta_0 - \beta_{i^*} x_{i^*}) h_+(\boldsymbol{x}) d\boldsymbol{x} \\ &\quad + \pi_+ \int_{\mathcal{X}} I_{\left\{x_{i^*} > \frac{1-\beta_0}{\beta_{i^*}}\right\}} v(\beta_0 + \beta_{i^*} x_{i^*} - 1) h_+(\boldsymbol{x}) d\boldsymbol{x} \\ &\quad + \pi_- \int_{\mathcal{X}} I_{\left\{x_{i^*} > \frac{-1-\beta_0}{\beta_{i^*}}\right\}} (1 + \beta_0 + \beta_{i^*} x_{i^*}) h_-(\boldsymbol{x}) d\boldsymbol{x} \\ &\quad + \pi_- \int_{\mathcal{X}} I_{\left\{x_{i^*} < \frac{-1-\beta_0}{\beta_{i^*}}\right\}} v(-\beta_0 - \beta_{i^*} x_{i^*} - 1) h_-(\boldsymbol{x}) d\boldsymbol{x}, \end{aligned}$$

which gives,

$$\begin{aligned} \frac{\partial Q(\beta_0, \beta_{i^*})}{\partial \beta_0} &= -\pi_+ \int_{\mathcal{X}} I_{\left\{x_{i^*} < \frac{1-\beta_0}{\beta_{i^*}}\right\}} h_+(\boldsymbol{x}) d\boldsymbol{x} + v\pi_+ \int_{\mathcal{X}} I_{\left\{x_{i^*} > \frac{1-\beta_0}{\beta_{i^*}}\right\}} h_+(\boldsymbol{x}) d\boldsymbol{x} \\ &\quad + \pi_- \int_{\mathcal{X}} I_{\left\{x_{i^*} > \frac{-1-\beta_0}{\beta_{i^*}}\right\}} h_-(\boldsymbol{x}) d\boldsymbol{x} - v\pi_- \int_{\mathcal{X}} I_{\left\{x_{i^*} < \frac{-1-\beta_0}{\beta_{i^*}}\right\}} h_-(\boldsymbol{x}) d\boldsymbol{x} \\ &= -\pi_+ \left[(1+v) \int_{\mathcal{X}} I_{\left\{x_{i^*} < \frac{1-\beta_0}{\beta_{i^*}}\right\}} h_+(\boldsymbol{x}) d\boldsymbol{x} - v \right] \\ &\quad + \pi_- \left[(1+v) \int_{\mathcal{X}} I_{\left\{x_{i^*} > \frac{-1-\beta_0}{\beta_{i^*}}\right\}} h_-(\boldsymbol{x}) d\boldsymbol{x} - v \right]. \end{aligned}$$

Note that $\frac{\partial Q(\beta_0,\beta_{i^*})}{\partial \beta_0}$ increases in β_0, $\frac{\partial Q(\beta_0,\beta_{i^*})}{\partial \beta_0} \to -\pi_+ - v\pi_- < 0$ as $\beta_0 \to -\infty$, and $\frac{\partial Q(\beta_0,\beta_{i^*})}{\partial \beta_0} \to v\pi_+\pi_- > 0$ as $\beta_0 \to \infty$. Therefore the minimizer $\tilde{\beta}_0$ of $Q(\beta_0, \beta_{i^*})$ for a given β_{i^*} exists. Using $\frac{\partial Q(\beta_0,\beta_{i^*})}{\partial \beta_0}|_{\beta_0=\tilde{\beta}_0} = 0$, we have

$$\begin{aligned} Q(\tilde{\beta}_0, \beta_i^*) &= 2\pi_- \int_{\mathcal{X}} I_{\left\{x_{i^*} > \frac{-1-\tilde{\beta}_0}{\beta_{i^*}}\right\}} h_-(\boldsymbol{x})d\boldsymbol{x} - 2\pi_- v \int_{\mathcal{X}} I_{\left\{x_{i^*} < \frac{-1-\tilde{\beta}_0}{\beta_{i^*}}\right\}} h_-(\boldsymbol{x})d\boldsymbol{x} \\ &+\beta_{i^*}\Bigg[\pi_- \int_{\mathcal{X}} I_{\left\{x_{i^*} > \frac{-1-\tilde{\beta}_0}{\beta_{i^*}}\right\}} x_{i^*} h_-(\boldsymbol{x})d\boldsymbol{x} - v\pi_- \int_{\mathcal{X}} I_{\left\{x_{i^*} < \frac{-1-\tilde{\beta}_0}{\beta_{i^*}}\right\}} x_{i^*} h_-(\boldsymbol{x})d\boldsymbol{x} \\ &-\pi_+ \textstyle\int_{\mathcal{X}} I_{\left\{x_{i^*} < \frac{1-\tilde{\beta}_0}{\beta_{i^*}}\right\}} x_{i^*} h_+(\boldsymbol{x})d\boldsymbol{x} + v\pi_+ \int_{\mathcal{X}} I_{\left\{x_{i^*} > \frac{1-\tilde{\beta}_0}{\beta_{i^*}}\right\}} x_{i^*} h_+(\boldsymbol{x})d\boldsymbol{x}\Bigg]. \end{aligned} \tag{3.34}$$

Now assume $\pi_+ > \pi_-$. Then $F_{i^*}^+ < \infty$ and $G_{i^*}^- = -\infty$. These may not be uniquely determined if there are intervals with probability zero, but the proof is essentially the same even if we assume uniqueness of those. Since $\frac{\partial Q(\beta_0,\beta_{i^*})}{\partial \beta_0}$ has zero at $\beta_0 = \tilde{\beta}_0$, we have

$$\begin{aligned} (1+v)\int_{\mathcal{X}} I_{\left\{x_{i^*} < \frac{1-\tilde{\beta}_0}{\beta_{i^*}}\right\}} h_+(\boldsymbol{x})d\boldsymbol{x} - v &= \frac{\pi_-}{\pi_+}\left[(1+v)\int_{\mathcal{X}} I_{\left\{x_{i^*} > \frac{-1-\tilde{\beta}_0}{\beta_{i^*}}\right\}} h_-(\boldsymbol{x})d\boldsymbol{x} - v\right] \\ &< \frac{\pi_-}{\pi_+}[(1+v)\cdot 1 - v], \end{aligned}$$

resulting

$$\int_{\mathcal{X}} I_{\left\{x_{i^*} < \frac{1-\tilde{\beta}_0}{\beta_{i^*}}\right\}} h_+(\boldsymbol{x})d\boldsymbol{x} < \frac{\frac{\pi_-}{\pi_+} + v}{1+v}.$$

Hence, we have

$$\begin{aligned} & \tfrac{1-\tilde{\beta}_0}{\beta_{i^*}} < F_{i^*}^+ < \infty \\ \Rightarrow\quad & \tfrac{-1-\tilde{\beta}_0}{\beta_{i^*}} < F_{i^*}^+ - \tfrac{2}{\beta_{i^*}} \to -\infty \quad \text{as } \beta_{i^*} \to 0 \\ \Rightarrow\quad & \textstyle\int_{\mathcal{X}} I_{\left\{x_{i^*} > \frac{-1-\tilde{\beta}_0}{\beta_{i^*}}\right\}} h_-(\boldsymbol{x})d\boldsymbol{x} \to 1 \\ \Rightarrow\quad & \textstyle\int_{\mathcal{X}} I_{\left\{x_{i^*} < \frac{1-\tilde{\beta}_0}{\beta_{i^*}}\right\}} h_+(\boldsymbol{x})d\boldsymbol{x} \to \frac{\frac{\pi_-}{\pi_+}+v}{1+v} \\ \Rightarrow\quad & \tfrac{1-\tilde{\beta}_0}{\beta_{i^*}} \to F_{i^*}^+ \quad \text{as } \beta_{i^*} \to 0 \end{aligned}$$

Since $\int_{\mathcal{X}} I_{\left\{x_{i^*} > \frac{-1-\tilde{\beta}_0}{\beta_{i^*}}\right\}} h_-(\boldsymbol{x})d\boldsymbol{x} \to 1$ and $v\int_{\mathcal{X}} I_{\left\{x_{i^*} < \frac{-1-\tilde{\beta}_0}{\beta_{i^*}}\right\}} h_-(\boldsymbol{x})d\boldsymbol{x} \to 0$, from (3.34) we obtain

$$Q(\tilde{\beta}_0, \beta_i^*) < 2\pi_- = \min_{\beta_0} Q(\beta_0) \quad \text{for some } \beta_{i^*} > 0$$

Hence, we proved that $\beta_i^* \neq 0$ for the case when $\pi_+ > \pi_-$. We can show the same result for the

case when $\pi_+ < \pi_-$ in the similar fashion. When $\pi_+ = \pi_-$, we can easily check that

$$\frac{1-\tilde{\beta}_0}{\beta_{i^*}} \to \infty \quad \text{as } \beta_{i^*} \to 0,$$

and

$$\frac{-1-\tilde{\beta}_0}{\beta_{i^*}} \to -\infty \quad \text{as } \beta_{i^*} \to 0.$$

Using these and (3.34), we have

$$Q(\tilde{\beta}_0, \beta_i^*) < 1 = \min_{\beta_0} Q(\beta_0) \quad \text{for some } \beta_{i^*} > 0$$

Hence, we have shown that $Q(\tilde{\beta}_0, \beta_i^*) < \min_{\beta_0} Q(\beta_0)$ for some $\beta_{i^*} > 0$ under the first condition of **A3**. In the similar manner, it can be shown that $Q(\tilde{\beta}_0, \beta_i^*) < \min_{\beta_0} Q(\beta_0)$ for some $\beta_{i^*} < 0$ under the second condtion of **A3**. Therefore, we have shown (3.33). □

The following lemma establishes the lower bound of $H(\boldsymbol{\beta}^*)$.

Lemma 6. *Suppose **A1**, **A3**, and **A4** are met. Then,*

$$\boldsymbol{\beta}^T H(\boldsymbol{\beta}^*)\boldsymbol{\beta} \geqslant (1+v)C_4\|\boldsymbol{\beta}\|^2,$$

where C_4 may depend on $\boldsymbol{\beta}^*$.

Proof. Using Lemma 5 in Koo et al. (2008),

$$\begin{aligned}\boldsymbol{\beta}^T H(\boldsymbol{\beta}^*)\boldsymbol{\beta} &= (1+v)\boldsymbol{\beta}^T E[\delta(1-Yf(\boldsymbol{X};\boldsymbol{\beta}))\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^T]\boldsymbol{\beta} \\ &\geqslant (1+v)C_4\|\boldsymbol{\beta}\|^2\end{aligned}$$

□

Lemma 7. *Assume **A1**-**A4** are satisfied. Then $Q(\boldsymbol{\beta})$ has a unique minimizer.*

Proof. By Lemma 2, we may choose any minimizer $\boldsymbol{\beta}^*$ from a bounded connected set of minimizers of $Q(\boldsymbol{\beta})$. Lemma 5 and 6 guarantees that $H(\boldsymbol{\beta})$ is positive definite at $\boldsymbol{\beta}^*$. Then $Q(\boldsymbol{\beta})$ is locally strictly convex at $\boldsymbol{\beta}^*$, implying that $Q(\boldsymbol{\beta})$ has a local minimum at $\boldsymbol{\beta}^*$. Therefore the minimizer of $Q(\boldsymbol{\beta})$ is unique. □

Now we prove Theorem 5 and 6. For $\boldsymbol{\theta} = (\theta_0, \boldsymbol{\theta}_+)^T \in \mathbb{R}^{d+1}$, define

$$\Lambda_n(\boldsymbol{\theta}) = n\left(q_{\lambda,n}(\boldsymbol{\beta}^* + \frac{\boldsymbol{\theta}}{\sqrt{n}}) - q_{\lambda,n}(\boldsymbol{\beta}^*)\right)$$

and

$$\Gamma_n(\boldsymbol{\theta}) = E\Lambda_n(\boldsymbol{\theta}).$$

By Taylor series expansion,

$$\begin{aligned}\Gamma_n(\boldsymbol{\theta}) &= n\left(Q(\boldsymbol{\beta}^* + \tfrac{\boldsymbol{\theta}}{\sqrt{n}}) - Q(\boldsymbol{\beta}^*)\right) + \tfrac{\lambda}{2}\left(\|\boldsymbol{\theta}_+\|^2 + 2\sqrt{n}\boldsymbol{\beta}_+^{*T}\boldsymbol{\theta}_+\right) \\ &= \tfrac{1}{2}\boldsymbol{\theta}^T H(\tilde{\boldsymbol{\beta}})\boldsymbol{\theta} + \tfrac{\lambda}{2}\left(\|\boldsymbol{\theta}_+\|^2 + 2\sqrt{n}\boldsymbol{\beta}_+^{*T}\boldsymbol{\theta}_+\right),\end{aligned}$$

where $\tilde{\boldsymbol{\beta}} = \boldsymbol{\beta}^* + (t/\sqrt{n})\boldsymbol{\theta}$ for some $0 < t < 1$. Define $D_{jk}(\boldsymbol{\alpha}) = H(\boldsymbol{\beta}^* + \boldsymbol{\alpha})_{jk} + H(\boldsymbol{\beta}^*)_{jk}$ for $0 \leqslant j, k \leqslant d$. Because $H(\boldsymbol{\beta})$ is continuous in $\boldsymbol{\beta}$, there exists $\delta_1 > 0$ such that $\|\boldsymbol{\alpha}\| < \delta_1$ implies $|D_{jk}(\boldsymbol{\alpha})| < \epsilon_1$ for any $\epsilon_1 > 0$ and $0 \leqslant j, k \leqslant d$. Then, for sufficiently large n such that $\|(t/\sqrt{n})\boldsymbol{\theta}\| < \delta_1$, we have

$$\begin{aligned}\left|\boldsymbol{\theta}^T\left(H(\tilde{\boldsymbol{\beta}}) - H(\boldsymbol{\beta}^*)\right)\boldsymbol{\theta}\right| &\leqslant \textstyle\sum_{j,k}|\theta_j||\theta_k|\left|D_{j,k}\left(\frac{t}{\sqrt{n}}\theta\right)\right| \\ &\leqslant \epsilon_1 \textstyle\sum_{j,k}|\theta_j||\theta_k| \\ &\leqslant 2\epsilon_1\|\boldsymbol{\theta}\|^2,\end{aligned}$$

resulting

$$\frac{1}{2}\boldsymbol{\theta}^T H(\tilde{\boldsymbol{\beta}})\boldsymbol{\theta} = \frac{1}{2}\boldsymbol{\theta}^T H(\boldsymbol{\beta}^*)\boldsymbol{\theta} + o(1).$$

Considering $\lambda = o(n^{-1/2})$, we have

$$\Gamma_n(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{\theta}^T H(\boldsymbol{\beta}^*)\boldsymbol{\theta} + o(1).$$

Now, let $\boldsymbol{W}_n = \sum_{i=1}^n\left(-\rho(1 - Y_i f(\boldsymbol{X}_i;\boldsymbol{\beta}^*))Y_i\tilde{\boldsymbol{X}}_i + v\rho(Y f(\boldsymbol{X};\boldsymbol{\beta}) - 1)Y_i\tilde{\boldsymbol{X}}_i\right)$. Observe that $E(\boldsymbol{W}_n) = S(\boldsymbol{\beta}^*) = 0$ and $E(\boldsymbol{W}_n\boldsymbol{W}_n^T) = \sum_{i=1}^n E[(\rho(1 - Y_i f(\boldsymbol{X}_i;\boldsymbol{\beta}^*)) + v^2\rho(Y_i f(\boldsymbol{X}_i;\boldsymbol{\beta}^*) - 1))\tilde{\boldsymbol{X}}_i\tilde{\boldsymbol{X}}_i^T]$. Hence, by central limit theorem, we have

$$\frac{1}{\sqrt{n}}\boldsymbol{W}_n \to N(0, nG(\boldsymbol{\beta}^*))$$

in distribution.

Now, we define

$$\begin{aligned} R_{i,n}(\boldsymbol{\theta}) &= g(Y_i f(\boldsymbol{X}_i;\boldsymbol{\beta}^* + \boldsymbol{\theta}/\sqrt{n})) - g(Y_i f(\boldsymbol{X}_i;\boldsymbol{\beta}^*)) \\ &\quad +\rho(1 - Y_i f(\boldsymbol{X}_i;\boldsymbol{\beta}^*))Y_i f(\boldsymbol{X}_i;\boldsymbol{\theta}/\sqrt{n}) - v\rho(Y_i f(\boldsymbol{X}_i;\boldsymbol{\beta}^*) - 1)Y_i f(\boldsymbol{X}_i;\boldsymbol{\theta}/\sqrt{n}), \end{aligned}$$

which gives

$$\Lambda_n(\boldsymbol{\theta}) = \Gamma_n(\boldsymbol{\theta}) + \boldsymbol{W}_n^T\boldsymbol{\theta}/\sqrt{n} + \sum_{i=1}^{n}(R_{i,n}(\boldsymbol{\theta}) - ER_{i,n}(\boldsymbol{\theta})).$$

If we let $z = Y_i f(\boldsymbol{X}_i;\boldsymbol{\beta}^* + \boldsymbol{\theta}/\sqrt{n})$ and $a = Y_i f(\boldsymbol{X}_i;\boldsymbol{\beta}^*)$, we can write

$$\begin{aligned} R_{i,n}(\boldsymbol{\theta}) &= g(z) - g(a) + I\{a \leqslant 1\}(z-a) - vI\{a > 1\}(z-a) \\ &= I\{z \leqslant 1\}(1-z) + I\{z > 1\}v(z-1) \\ &\quad -I\{a \leqslant 1\}(1-a) - I\{a > 1\}v(a-1) + I\{a \leqslant 1\}(z-a) - I\{a > 1\}v(z-a) \\ &= I\{z \leqslant 1\}(1-z) + I\{z > 1\}v(z-1) \\ &\quad +I\{a \leqslant 1\}(z-1) - I\{a > 1\}v(z-1) \\ &= [I\{z \leqslant 1\} - I\{a \leqslant 1\}](1-z) + [I\{z > 1\} - I\{a > 1\}]v(z-1) \\ &\leqslant (a-z)I\{z \leqslant 1, a > 1\} + v(z-a)I\{z > 1, a \leqslant 1\} \\ &\leqslant \max\{1, v\}|z-a|I\{|1-a| \leqslant |z-a|\}. \end{aligned}$$

Thus, we have

$$|R_{i,n}(\boldsymbol{\theta})| \leqslant \max\{1, v\}(|f(\boldsymbol{X}_i;\boldsymbol{\theta})|/\sqrt{n})I_{\{|1-Y_i f(\boldsymbol{X}_i;\boldsymbol{\beta}^*)| \leqslant |f(\boldsymbol{X};\boldsymbol{\theta})|/\sqrt{n}\}},$$

resulting

$$\begin{aligned} \sum_{i=1}^{n} E|R_{i,n}(\boldsymbol{\theta}) - ER_{i,n}(\boldsymbol{\theta})|^2 &= \sum_{i=1}^{n}[E(R_{i,n}(\boldsymbol{\theta}))^2 - (ER_{i,n}(\boldsymbol{\theta}))^2] \\ &\leqslant \sum_{i=1}^{n} E(R_{i,n}(\boldsymbol{\theta}))^2 \\ &\leqslant \sum_{i=1}^{n} E\left[\max\{1, v^2\}\left|\frac{f(\boldsymbol{X}_i;\boldsymbol{\theta})}{\sqrt{n}}\right|^2 I_{\{|1-Y_i f(\boldsymbol{X}_i;\boldsymbol{\beta}^*)| \leqslant |f(\boldsymbol{X};\boldsymbol{\theta})|/\sqrt{n}\}}\right] \\ &\leqslant \max\{1, v^2\}\|\boldsymbol{\theta}\|^2 E\left[(1 + \|\boldsymbol{X}\|^2)I_{\{|1-Y_i f(\boldsymbol{X}_i;\boldsymbol{\beta}^*)| \leqslant \sqrt{1+\|\boldsymbol{X}\|^2}\|\boldsymbol{\theta}\|/\sqrt{n}\}}\right]. \end{aligned}$$

Note that **A1** implies that $E(\|\boldsymbol{X}\|^2) < \infty$. Thus, for any $\epsilon > 0$, there exists C_5 such that

$E[(1+\|\boldsymbol{X}\|^2)I_{\{\|\boldsymbol{X}\|>C_5\}}] < \epsilon/2$. Observe

$$\begin{aligned} & E\left[(1+\|\boldsymbol{X}\|^2)I_{\{|1-Y_i f(\boldsymbol{X}_i;\boldsymbol{\beta}^*)|\leqslant\sqrt{1+\|\boldsymbol{X}\|^2}\|\boldsymbol{\theta}\|/\sqrt{n}\}}\right] \\ \leqslant\ & E\left[(1+\|\boldsymbol{X}\|^2)I_{\{\|\boldsymbol{X}\|>C_5\}}\right] + (1+c_5^2)P\left(|1-Y_i f(\boldsymbol{X}_i;\boldsymbol{\beta}^*)| \leqslant \sqrt{1+C_5^2}\|\boldsymbol{\theta}\|/\sqrt{n}\right). \end{aligned}$$

The second term $(1+c_5^2)P\left(|1-Y_i f(\boldsymbol{X}_i;\boldsymbol{\beta}^*)| \leqslant \sqrt{1+C_5^2}\|\boldsymbol{\theta}\|/\sqrt{n}\right)$ goes to zero as $n \to \infty$ because of **A1**. Thus, we have $\sum_{i=1}^n E|R_{i,n}(\boldsymbol{\theta}) - ER_{i,n}(\boldsymbol{\theta})|^2 \to 0$ as $n \to \infty$. Hence, we can write

$$\Lambda_n(\boldsymbol{\theta}) = \Gamma_n(\boldsymbol{\theta}) + \boldsymbol{W}_n^T\boldsymbol{\theta}/\sqrt{n} + o_P(1).$$

Now, we define $\boldsymbol{\eta}_n(\boldsymbol{\theta}) = -H(\boldsymbol{\beta}^*)^{-1}\boldsymbol{W}_n/\sqrt{n}$. Using Convexity Lemma in Pollard (1991), we have

$$\Lambda_n(\boldsymbol{\theta}) = \frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\eta}_n)^T H(\boldsymbol{\beta}^*)(\boldsymbol{\theta}-\boldsymbol{\eta}_n) - \frac{1}{2}\boldsymbol{\eta}^T H(\boldsymbol{\beta}^*)\boldsymbol{\eta} + r_n(\boldsymbol{\theta}),$$

where, for each compact set $K \in \mathbb{R}$,

$$\sup_{\theta\in K}|r_n(\boldsymbol{\theta})| \to 0$$

in probability. Since $\boldsymbol{\eta}_n$ converges in distribution, there exists a compact set K which contains B_ϵ, where B_ϵ is a closed ball with center $\boldsymbol{\eta}_n$ and radius ϵ with probability arbitrarily close to one. This gives

$$\Delta_n = \sup_{\theta\in B_\epsilon}|r_n(\boldsymbol{\theta})| \to 0 \tag{3.35}$$

in probability. Now consider the outside of the ball B_ϵ. Writing $\boldsymbol{\theta} = \boldsymbol{\eta}_n + \gamma\boldsymbol{u}$ and $\boldsymbol{\theta}^* = \boldsymbol{\eta}_n + \epsilon\boldsymbol{u}$ with $\gamma > \epsilon$ and a unit vector $\boldsymbol{u}$, Lemma 6 and convexity of Λ_n gives

$$\begin{aligned} \tfrac{\epsilon}{\gamma}\Lambda_n(\boldsymbol{\theta}) + \left(1-\tfrac{\epsilon}{\gamma}\right)\Lambda_n(\boldsymbol{\eta}_n) &\geqslant \Lambda_n(\boldsymbol{\theta}^*) \\ &\geqslant \tfrac{1}{2}(\boldsymbol{\theta}^*-\boldsymbol{\eta}_n)^T H(\boldsymbol{\beta}^*)(\boldsymbol{\theta}^*-\boldsymbol{\eta}_n) - \tfrac{1}{2}\boldsymbol{\eta}^T H(\boldsymbol{\beta}^*)\boldsymbol{\eta} - \Delta_n \\ &\geqslant \tfrac{C_4}{2}\epsilon^2 + \Lambda_n(\boldsymbol{\eta}_n) - 2\Delta_n. \end{aligned}$$

Thus, we have

$$\frac{\epsilon}{\gamma}(\Lambda_n(\boldsymbol{\theta}) - \Lambda_n(\boldsymbol{\eta}_n)) \geqslant \frac{C_4}{2}\epsilon^2 - 2\Delta_n,$$

finally giving

$$\inf_{\|\boldsymbol{\theta}-\boldsymbol{\eta}_n\|>\epsilon} \Lambda_n(\boldsymbol{\theta}) \geqslant \Lambda_n(\boldsymbol{\eta}_n) + \left(\frac{C_4}{2}\epsilon^2 - 2\Delta_n\right).$$

By (3.35), we can take Δ_n so that $\frac{C_4}{2}\epsilon^2 - 2\Delta_n > 0$ with probability tending to one. Therefore, the minimum of Λ_n cannot occur at any $\boldsymbol{\theta}$ with $\|\boldsymbol{\theta} - \boldsymbol{\eta}_n\| > \epsilon$. Note that the minimizer of Λ_n is $\sqrt{n}(\hat{\boldsymbol{\beta}}_{\lambda,n} - \boldsymbol{\beta}^*)$. Hence we have

$$P(\|\sqrt{n}(\hat{\boldsymbol{\beta}}_{\lambda,n} - \boldsymbol{\beta}^*) - \boldsymbol{\eta}_n\| > \epsilon) \to 0$$

resulting

$$\sqrt{n}(\hat{\boldsymbol{\beta}}_{\lambda,n} - \boldsymbol{\beta}^*) \to \boldsymbol{\eta}_n$$

in probability. This completes the proof. □

Chapter 4

Multicategory Classification

4.1 Introduction

Binary classification problems are heavily studied, while in contrast, the attentions on multicategory problems are much less so. To solve a multicategory problem, there are two major groups of approaches. One is to employ multiple binary classifiers and then combine the results. The one-versus-rest and one-versus-one approaches are common examples of this type. Although the extension is simple to implement, there are drawbacks with these approaches. The one-versus-one approach may not work well when the numbers of observations in some classes are small. For the one-versus-rest approach, a serious drawback is that the approach may not be consistent when there is no dominating class, i.e., when the maximum class conditional probability is less than 0.5. The other group of multicategory approaches is to use simultaneous multicategory formulations. For example, Vapnik (1998); Crammer and Singer (2001); Lee et al. (2004) proposed various SVM techniques for simultaneous multicategory classification. One difficulty of these approaches is that the corresponding computational complexity grows very rapidly when the number of classes gets large.

Despite progress in multicategory classification, many challenges are yet to be solved. In particular, with the abundance of complex data with large volume, it is desirable to have multicategory classification techniques that are

- based on simultaneous formulation with sound theoretical properties;
- able to handle high dimensional data;
- efficient to compute even when the class number is large;

- able to estimate conditional class probabilities.

In this chapter, we propose a novel simultaneous multicategory technique, namely the Multicategory Composite Least Squares (CLS) Classifier. The proposed CLS classifier possesses all four aforementioned properties. Motivated from multicategory SVMs, the CLS classifier is based on a simultaneous formulation by using all data at once to produce a multicategory classifier. It has the desirable consistency. Similar to the SVM, it has the ability to handle high dimensional data. In contrast to the challenging optimization of multicategory SVM, the CLS classifier is very efficient to compute. Surprisingly, although it makes use of a simultaneous formulation, its computation can be decoupled as multiple smaller optimization problems as in the one-versus-rest approach. Consequently, computation complexity of the CLS classifier grows with the class number *linearly*, thus it is feasible even for problems with very large number of classes.

The CLS classifier is closely related to the SVM. Instead of using the multicategory hinge loss function as in the SVM, it makes use of the proposed composite squared loss which yields very efficient computation. More specifically, the CLS classifier uses a nontrivial *convex combination* of two different types of squared loss functions, with one of the two being the loss of the Proximal SVM (PSVM) (Suykens and Vandewalle, 1999; Fung and Mangasarian, 2001; Tang and Zhang, 2006). The combination is shown to be necessary as the performance of the combined loss is much better than the uncombined ones. Another important advantage of the CLS classifier is its ability to produce class probability estimation, while in contrast, the SVM cannot. Due to the special form of the loss function for the CLS classifier, we are able to derive closed-form solutions and the formulae to predict class probability.

The rest of this chapter is organized as follows. In Section 4.2, we give a brief background on multicategory classification and review some existing multicategory SVM techniques. In Section 4.3, we propose the multicagegory CLS classifier, study its property, and provide probability estimation. Section 4.4 discusses the computational algorithm. Numerical results through simulated examples and real data applications are presented in Section 4.5. Some discussion is given in Section 4.6. Proofs of theoretical results are included in Section 4.7.

4.2 Background on Multicategory Classification

Suppose our training dataset is $\{\boldsymbol{x}_i, y_i\}_{i=1,\cdots,n}$. Here, similar as before, $\boldsymbol{x}_i$ is the d-dimensional covariate, and $y_i \in \{1, \cdots, k\}$ represents a k-class label with $k > 2$. Our goal is to build a classifier based on the training data so that we can predict the class membership of new observations.

4.2.1 Sequence of Binary Classifiers

To solve a multicategory problem, a natural and direct way is to implement multiple binary classifiers. For example, one can use the one-versus-rest or one-versus-one approach. The one-versus-rest approach relabels the training data in the class j as the positive class and data which are not in the class j as the negative class, for each $j = 1, \cdots, k$. Then, one can employ a sequence of k binary classifiers for the membership of each data point, which can possibly give contradictory results among the k binary classifiers. The one-versus-one approach applies a given binary classifier to a binary problem of the class j_1 and the class j_2 for each of all possible pairs $j_1, j_2 \in \{1, \cdots, k\}$. Overall, $\binom{k}{2}$ binary classifications are performed. For each binary problem, the dataset can be very small.

When there is no dominating class, in the SVM context, the one-versus-rest approach can be self-contradicted (Lee et al., 2004) and Fisher consistency is not guaranteed (Liu, 2007). Thus, it is necessary to generalize binary classification methods to multicategory versions which consider all classes simultaneously and retain good properties of the original methods.

4.2.2 Simultaneous methods

In contrast to the approach using a sequence of binary classifiers, one can obtain the decision functions for all classes simultaneously and compare them all at once to predict class membership. More specifically, let $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_k(\boldsymbol{x})) \in \prod_{j=1}^{k}(\{1\} + \mathcal{H}_K)$ be the decision function vector, where $\mathcal{H}_K$ denotes a reproducing kernel Hilbert space generated by the kernel K. Once the value of $\boldsymbol{f}$ is obtained, the class membership of any new data point $\boldsymbol{x}$ is estimated by $\hat{y} = \text{argmax}_{j=1,2,\cdots,k} f_j(\boldsymbol{x})$. To remove redundancy in solutions, we use the zero-sum constraint, $\sum_{j=1}^{k} f_i(\boldsymbol{x}) = 0$. This formulation becomes equivalent to the binary problem when $k = 2$. When

$k = 2$, we have $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}))$ as the decision function vector with $f_1(\boldsymbol{x}) = -f_2(\boldsymbol{x})$ because of the zero-sum constraint. Note that $\text{sign}[f_1(\boldsymbol{x}) - f_2(\boldsymbol{x})] > 0$ if $f_1(\boldsymbol{x}) > f_2(\boldsymbol{x})$ and $\text{sign}[f_1(\boldsymbol{x}) - f_2(\boldsymbol{x})] \leqslant 0$ otherwise. Thus, having $f = f_1 - f_2$ and using the class label $y = -1$ instead of $y = 2$ makes $\text{sign}(f(\boldsymbol{x}))$ and $\text{argmax}_{j=1,\cdots,k} f_j(\boldsymbol{x})$ with $k = 2$ equivalent estimators for the class membership of the entry $\boldsymbol{x}$.

With a sensible loss function L given, the multicategory large margin classifier solves

$$\begin{aligned} &\min_{f \in \mathcal{F}} \sum_{i=1}^{n} L(\boldsymbol{f}(\boldsymbol{x}_i), y_i) + \lambda \sum_{j=1}^{k} J(f_j), \\ &\text{subject to} \quad \sum_{j=1}^{k} f_i(\boldsymbol{x}) = 0 \quad \text{for } i = 1, \cdots, n. \end{aligned} \tag{4.1}$$

Since a point $\boldsymbol{x}$ is misclassified when $y \neq \text{argmax}_j f_j(\boldsymbol{x})$, a good loss function L should force f_k to be the maximum among $f_1, \cdots, f_k$.

Fisher consistency is an important issue for the classification. It requires that a classifier approximates the Bayes rule when the sample size is sufficiently large. Thus in our formulation, Fisher consistency requires that $\text{argmax}_j f_j^* = \text{argmax}_j P_j$, where $P_j(\boldsymbol{x}) = P(Y = j|\boldsymbol{x})$, and $\boldsymbol{f}^*(\boldsymbol{x}) = (f_1^*(\boldsymbol{x}), \ldots, f_k^*(\boldsymbol{x}))$ denotes the minimizer of $E[L(\boldsymbol{f}(\boldsymbol{x}), y)|\boldsymbol{X} = \boldsymbol{x}]$. When we select a loss function L, it is necessary to study its Fisher consistency.

4.2.3 Existing Multicategory SVMs

In this section, we focus on different versions of simultaneous multicategory SVMs. In the literature, there are several different ways to extend the binary hinge loss to the multicategory versions. Here we list several commonly used versions:

1. (Naive hinge loss) $[1 - f_y(\boldsymbol{x})]_+$;

2. (Vapnik, 1998; Weston and Watkins, 1999; Bredensteiner and Bennett, 1999) $\sum_{j \neq y}[1 - (f_y(\boldsymbol{x}) - f_j(\boldsymbol{x}))]_+$;

3. (Crammer and Singer, 2001; Liu and Shen, 2006) $\sum_{j \neq y}[1 - \min_j(f_y(\boldsymbol{x}) - f_j(\boldsymbol{x}))]_+$;

4. (Lee et al., 2004) $\sum_{j \neq y}[1 + f_j(\boldsymbol{x})]_+$.

Loss 1 is a simple extension of the binary hinge loss. We call this loss function as the naive hinge loss. Liu (2007) showed that the minimizer $\boldsymbol{f}^*$ of $E[[1 - f_y(\boldsymbol{x})]_+ | \boldsymbol{X} = \boldsymbol{x}]$ subject to $\sum_{j=1}^k f_j(\boldsymbol{x}) = 0$ is $f_j^*(\boldsymbol{x}) = -(k-1)$ if $j = \text{argmin}_j P_j(\boldsymbol{x})$ and 1 otherwise, which implies the naive hinge loss function is not Fisher consistent. He also showed the cases when Loss 2 and Loss 3 are not Fisher consistent. In contrast to these three loss functions, Loss 4 is Fisher consistent as the minimizer of $\boldsymbol{f}^*$ of $E[\sum_{j \neq y}[1 + f_j(\boldsymbol{x})]_+ | \boldsymbol{X} = \boldsymbol{x}]$ subject to $\sum_{j=1}^k f_j(\boldsymbol{x}) = 0$ is $f_j^*(\boldsymbol{x}) = k - 1$ if $j = \text{argmax}_j P_j(\boldsymbol{x})$ and -1 otherwise.

Liu and Yuan proposed a new group of the multicategory SVM loss functions called the reinforced hinge loss,

$$L(\boldsymbol{f}(\boldsymbol{x}), y) = \gamma[(k-1) - f_y(\boldsymbol{x})]_+ + (1-\gamma) \sum_{j \neq y} [1 + f_j(\boldsymbol{x})]_+. \tag{4.2}$$

It is the convex combination of the naive hinge loss and the Loss 4 by Lee et al. (2004) with weights $(\gamma, 1-\gamma)$. When $\gamma = 1/2$, if we replace $k-1$ in (4.2) by 1, it becomes $\sum_{j=1}^k [1 - c_j^y f_j(\boldsymbol{x})]_+$, where $c_j^y = 1$ if $j = y$ and -1 otherwise. Minimizing this loss function is equivalent to the one-versus-rest method (Weston, 1999), except the one-versus-rest approach does not enforce the zero-sum constraint. Thus, we can conclude that the reinforced hinge loss builds a connection between the one-versus-rest approach and the simultaneous classification approach. Interestingly, even though the naive hinge loss function is not Fisher consistent, the reinforced hinge loss function is Fisher consistent when $0 \leqslant \gamma \leqslant 1/2$. Liu and Yuan showed that the loss function (4.2) with $\gamma = 1/2$ gives the best classification performance.

One computational difficulty of the hinge loss is that it is not differentiable. This causes the minimizer of the hinge loss not to attain all information of the class probability. To improve this, the squared loss function can be employed instead of the hinge loss. Because the squared loss function is differentiable, the computation is easier and its minimizer yields the class probability, which enables us to estimate the class probability. Tang and Zhang (2006) proposed the multiclass proximal SVM, which employs the squared loss function for simultaneous multicategory classification frame work. More specifically, they used the loss function

$$L(\boldsymbol{f}(\boldsymbol{x}), y) = \sum_{j \neq y} \Big(1 + f_j(\boldsymbol{x})\Big)^2. \tag{4.3}$$

Note that this loss function is essentially the same extension with Loss 4, except it uses the squared loss instead of the hinge loss. Tang and Zhang (2006) showed the loss function of the multiclass proximal SVM is always Fisher consistent. Moreover, the minimizer $\boldsymbol{f}(\boldsymbol{x})$ of $E[\sum_{j\neq y}(1+f_j(\boldsymbol{x}))^2|\boldsymbol{X}=\boldsymbol{x}]$ satisfies

$$f_j(\boldsymbol{x}) = \frac{k}{1-P_j(\boldsymbol{x})}\Big/ \quad \sum_{l=1}^{k}\frac{1}{1-P_j(\boldsymbol{x})}\Bigg) - 1,$$

hence one can estimate the class probability $P_j(\boldsymbol{x})$ for $j = 1, \cdots, k$ using $\hat{f}_j(\boldsymbol{x})$.

4.3 Multicategory Composite Least Squares Classifier

The reinforced multicategory SVM establishes a bridge between two different versions of hinge loss functions with different values of weight γ and it shows the best choice of γ is around 0.5. This indicates the combination of those loss functions works better than the uncombined ones. This motivates us to combine two different versions of squared loss functions in the similar manner. More specifically, we propose to use the following family of composite squared loss functions

$$L(\boldsymbol{f}(\boldsymbol{x}), y) = \gamma[(k-1) - f_y(\boldsymbol{x})]^2 + (1-\gamma)\sum_{j\neq y}[1+f_j(\boldsymbol{x})]^2, \tag{4.4}$$

subject to $\sum_{j=1}^{k} f_j(\boldsymbol{x}) = 0$, where $\gamma \in [0,1]$. We call problem (4.1) with the composite squared loss as the Multicategory Composite Least Squares (CLS) Classifier. It can be easily shown that the multicategory CLS classifier with $\gamma = 0$ is equivalent to the multiclass proximal SVM of Tang and Zhang (2006) with all misclassification costs equal.

To further understand the proposed loss family (4.4), we express the composite squared loss function using the multiple comparison vector representation proposed by Liu and Shen (2006). In particular, they defined the comparison vector $\boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}), y) = (f_y(\boldsymbol{x}) - f_1(\boldsymbol{x}), \cdots, f_y(\boldsymbol{x}) - f_{y-1}(\boldsymbol{x}), f_y(\boldsymbol{x}) - f_{y+1}(\boldsymbol{x}), \cdots, f_y(\boldsymbol{x}) - f_k(\boldsymbol{x}))$. Then, $\min(\boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x})), y) \leqslant 0$ if and only if an observation $(\boldsymbol{x}, y)$ is misclassified. Let $\boldsymbol{u} = \boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}), y)$. Using this notation, the $0-1$ loss

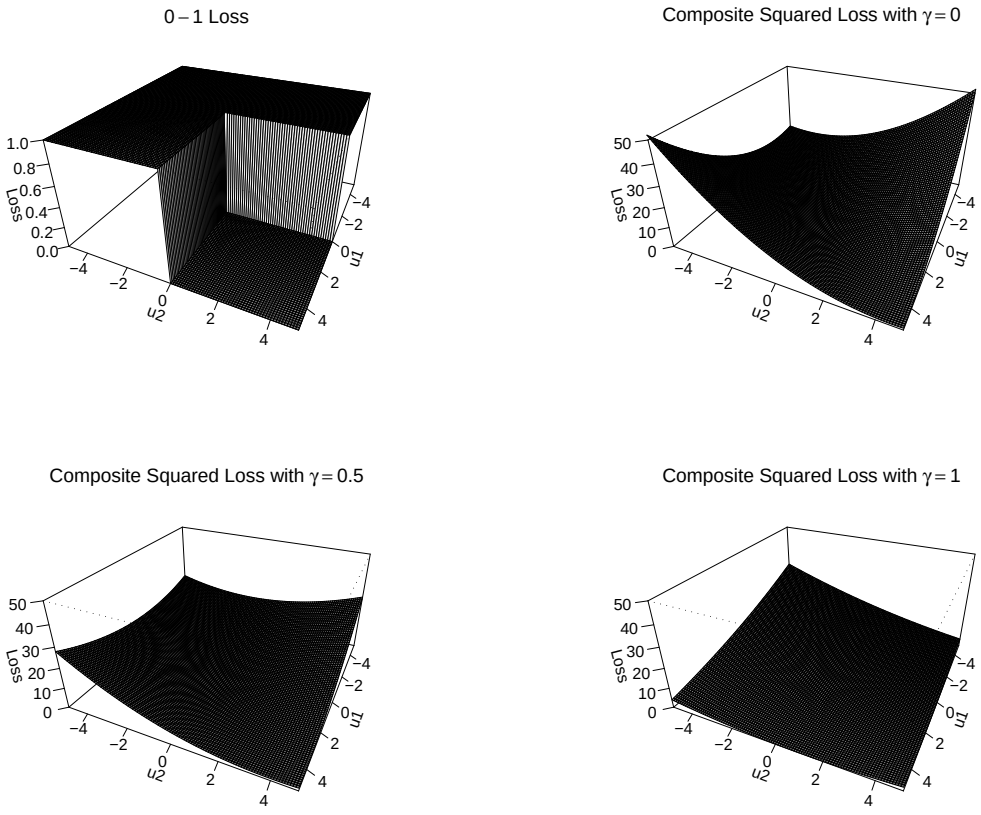

Figure 4.1: Plot of the $0-1$ loss function and the composite squared loss functions with $\gamma = 0, 0.5, 1$.

function becomes $I\{\min_j u_j \leqslant 0\}$. The composite squared loss function can be written as

$$\gamma[(k-1) - \sum_{l=1}^{k-1} u_l/k]^2 + (1-\gamma)\sum_{j=1}^{k-1}[1 + \sum_{l=1}^{k-1} u_l/k - u_j]^2, \tag{4.5}$$

and we plot (4.5) in Figure 4.1 with $\gamma = 0, 0.5, 1$, as well as $0-1$ loss function. We can see that as γ increases, the value of the loss function increases when both u_1 and u_2 are negative, while the loss decreases when only one of u_l's is negative.

The behavior of γ in the multicategory CLS classifier is very different from that of the RMSVM as shown in the numerical examples in Section 4.5. In particular, $\gamma = 0.5$ does not always show the best performance, thus the choice of γ should depend on the data. However, unlike the RMSVM, the multicategory CLS classifier offers class probability estimation which

enables one to better understand the nature of the data.

4.3.1 Properties of the multicategory CLS classifier

The following theorem establishes Fisher consistency of the composite loss function in a general form which includes the composite squared loss function as a special case.

Theorem 7. *Suppose a function $g(u)$ is twice differentiable, $g'(u) < 0$, and $g''(0) > 0$. Let*

$$L(\boldsymbol{f}(\boldsymbol{x}), y) = \gamma g(f_y(\boldsymbol{x})) + (1-\gamma)\sum_{j \neq y} g(-f_j(\boldsymbol{x})).$$

Then, the minimizer $\boldsymbol{f}^*$ of $E[L(\boldsymbol{f}(\boldsymbol{x}), y)]$, subject to $\sum_j^k f_j(\boldsymbol{x}) = 0$, satisfies the following: $\text{argmax}_j\, P_j(\boldsymbol{x}) = \text{argmax}_j\, f_j(\boldsymbol{x})$.

Remark 1 The conclusion of the Theorem 7 holds if the assumption on $g(u)$ is reduced to that $g'(u) < 0$ and $g'(u)$ is strictly increasing.

Remark 2 The reinforced hinge loss function is Fisher consistent only when $0 \leqslant \gamma \leqslant 1/2$. However, the composite squared loss function is always Fisher consistent for all values of $\gamma \in [0, 1]$.

4.3.2 Probability Estimation

It can be shown that the theoretical minimizer $\boldsymbol{f}^*(\boldsymbol{x})$ of $E[L(\boldsymbol{f}(\boldsymbol{X}), Y)|\boldsymbol{X} = \boldsymbol{x}]$ is a function of the conditional class probabilities. Hence, we can use $\hat{\boldsymbol{f}}^*(\boldsymbol{x})$ to predict class probabilities. The following theorem shows the exact form of $\boldsymbol{f}^*(\boldsymbol{x})$.

Theorem 8. *The minimizer $\boldsymbol{f}^*(\boldsymbol{x})$ of $E[L(\boldsymbol{f}(\boldsymbol{X}), Y)|\boldsymbol{X} = \boldsymbol{x}]$ with L in (4.4), subject to $\sum_j^k f_j(\boldsymbol{x}) = 0$, is $(f_1^*(\boldsymbol{x}), \cdots, f_k^*(\boldsymbol{x}))$, with*

$$f_j^*(\boldsymbol{x}) = (\frac{\sum_{l=1}^k a_l b_l}{\sum_{l=1}^k a_l} + b_j) a_j$$

where

$$a_l = 1/[2\gamma P_l(\boldsymbol{x}) + 2(1-\gamma)(1 - P_l(\boldsymbol{x}))]$$

and

$$b_l = 2\gamma(k-1)P_l(\boldsymbol{x}) - 2(1-\gamma)(1-P_l(\boldsymbol{x}))$$

for $l = 1, \cdots, k$.

Due to the complicated structure of $f_j^*(\boldsymbol{x})$, it is difficult to recover $(P_1(\boldsymbol{x}), \cdots, P_k(\boldsymbol{x}))$ from $(f_1(\boldsymbol{x}), \cdots, f_k(\boldsymbol{x}))$ for a general γ. However, for $\gamma = 0, 0.5$, and 1, we are able to make $a_l b_l$ simple to recover conditional class probabilities $(P_1(\boldsymbol{x}), \cdots, P_k(\boldsymbol{x}))$ from $(f_1(\boldsymbol{x}), \cdots, f_k(\boldsymbol{x}))$. The formulae to estimate class probabilities using $(f_1(\boldsymbol{x}), \cdots, f_k(\boldsymbol{x}))$ for $\gamma = 0, 0.5$, and 1 are given as follows:

$$\text{For } \gamma = 0, \quad \hat{P}_j(\boldsymbol{x}) = 1 - (k-1)\frac{1/[1+\hat{f}_j(\boldsymbol{x})]}{\sum_{l=1}^k 1/[1+\hat{f}_l(\boldsymbol{x})]}; \tag{4.6}$$

$$\text{For } \gamma = 0.5, \quad \hat{P}_j(\boldsymbol{x}) = \frac{1}{k}(1+\hat{f}_j(\boldsymbol{x})); \tag{4.7}$$

$$\text{For } \gamma = 1, \quad \hat{P}_j(\boldsymbol{x}) = \frac{1/[\hat{f}_j(\boldsymbol{x}) - (k-1)]}{\sum_{l=1}^k 1/[\hat{f}_l(\boldsymbol{x}) - (k-1)]}. \tag{4.8}$$

Notice that the proposed estimators of class probabilities $\hat{P}_j(\boldsymbol{x})$ sum up to 1, that is, $\sum_{j=1}^k \hat{P}_j(\boldsymbol{x}) = 1$. However, individual estimator $\hat{P}_j(\boldsymbol{x})$ in (4.6)-(4.8) may be outside of $[0, 1]$. To ensure the estimiated probabilities are proper, we propose to rescale the probability estimates using the following formulae,

$$\hat{P}_j^{scaled}(\boldsymbol{x}) = \frac{\hat{P}_j(\boldsymbol{x}) - \min_{l=1,\cdots,k} \hat{P}_l(\boldsymbol{x})}{\sum_{l=1}^k [\hat{P}_l(\boldsymbol{x}) - \min_{m=1,\cdots,k} \hat{P}_m(\boldsymbol{x})]}.$$

4.4 Computational Algorithm

Since $f_j \in (\{1\} + \mathcal{H}_K)$, we can write $f_j(\boldsymbol{x}) = \beta_{j0} + \sum_{i=1}^n \beta_{ji} K(\boldsymbol{x}_i, \boldsymbol{x})$ using the representative theorem (Kimeldorf and Wahba, 1971). The L_2 penalty commonly used is $J(\boldsymbol{f}_j) = \|f_j\|_{\mathcal{H}_K}^2 = \boldsymbol{\beta}_j^T \boldsymbol{K} \boldsymbol{\beta}_j$, where $\boldsymbol{\beta}_j = (\beta_{j1}, \cdots, \beta_{jn})^T$ and $\boldsymbol{K}$ is the $n \times n$ matrix with ij-th entry $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$. For simplicity of calculation, we use the penalty term $J(\boldsymbol{f}_j) = \|f_j\|_{\mathcal{H}_K}^2 + \beta_{j0}^2 = \boldsymbol{\beta}_j^T \boldsymbol{K} \boldsymbol{\beta}_j + \beta_{j0}^2$ as in Tang and Zhang (2006), which results in a closed form solution. This makes computation simpler and gives similar results with the same problem using the original L_2 penalty. In addition, we

use a new coding to replace y with z defined as

$$z_{ij} = \begin{cases} 1 & \text{if } y_i = j \\ -\frac{1}{k-1} & \text{if } y_i \neq j. \end{cases} \tag{4.9}$$

Let $w_{ij} = \gamma$ if $y_i = j$ and $1 - \gamma$ otherwise. Then minimizing $\sum_{i=1}^{n} L(\boldsymbol{f}(\boldsymbol{x}_i), y_i)$ with L in (4.4) becomes equivalent to minimizing

$$\begin{aligned} & \sum_{i=1}^{n} \left[\gamma \left(1 - f_{y_i}(\boldsymbol{x}_i)\right)^2 + (1-\gamma) \sum_{j \neq y_i} \left(-\frac{1}{k-1} - f_j(\boldsymbol{x}_i) \right)^2 \right] \\ = & \sum_{i=1}^{n} \sum_{j=1}^{k} w_{ij} (z_{ij} - f_j(\boldsymbol{x}_i))^2 \\ = & \sum_{j=1}^{k} [\boldsymbol{z}_j - \tilde{\boldsymbol{X}} \tilde{\boldsymbol{\beta}}_j]^T \boldsymbol{W}_j [\boldsymbol{z}_j - \tilde{\boldsymbol{X}} \tilde{\boldsymbol{\beta}}_j], \end{aligned} \tag{4.10}$$

where $\boldsymbol{z}_j = (z_{1j}, \cdots, z_{nj})^T$, $\mathbf{1}_n = (1, \cdots, 1)^T$, $\tilde{\boldsymbol{X}} = [\mathbf{1}_n, \boldsymbol{K}]$, $\tilde{\boldsymbol{\beta}}_j = (\beta_{j0}, \boldsymbol{\beta}_j^T)^T$ and $\boldsymbol{W}_j = \text{diag}\{w_{1j}, \cdots, w_{nj}\}$. Let $\mathbf{0}_n = (0, \cdots, 0)^T$,

$$\boldsymbol{G} = \begin{pmatrix} 1 & \mathbf{0}_n^T \\ \mathbf{0}_n & \boldsymbol{K} \end{pmatrix},$$

$\tilde{\boldsymbol{X}}_j^* = \boldsymbol{W}_j^{1/2} \tilde{\boldsymbol{X}} \boldsymbol{G}^{-1/2}$, $\boldsymbol{z}_j^* = \boldsymbol{W}_j^{1/2} \boldsymbol{z}_j$, and $\boldsymbol{\beta}_j^* = \boldsymbol{G}^{1/2} \tilde{\boldsymbol{\beta}}_j$. Then, (4.10) can be written as $\sum_{j=1}^{k} [\boldsymbol{z}_j^* - \tilde{\boldsymbol{X}}^* \boldsymbol{\beta}_j^*]^T [\boldsymbol{z}_j^* - \tilde{\boldsymbol{X}}^* \boldsymbol{\beta}_j^*]$. Hence, solving (4.1) with the loss function L in (4.4) and the penalty term $J(\boldsymbol{f}_j) = \boldsymbol{\beta}_j^T \boldsymbol{K} \boldsymbol{\beta}_j + \beta_{j0}^2$ is equivalent to minimizing

$$\sum_{j=1}^{k} [\boldsymbol{z}_j^* - \tilde{\boldsymbol{X}}_j^* \boldsymbol{\beta}_j^*]^T [\boldsymbol{z}_j^* - \tilde{\boldsymbol{X}}_j^* \boldsymbol{\beta}_j^*] + \lambda \boldsymbol{\beta}_j^{*T} \boldsymbol{\beta}_j^*$$

subject to $\sum_{j=1}^{k} \boldsymbol{\beta}_j^* = \mathbf{0}_{n+1}$. To solve this, we consider its dual problem with the Lagrange multiplier vector $2\boldsymbol{u} \in \mathbf{R}^{n+1}$,

$$\mathfrak{L}(\boldsymbol{\beta}, \boldsymbol{u}) = \sum_{j=1}^{k} \left[(\boldsymbol{z}_j^* - \tilde{\boldsymbol{X}}_j^* \boldsymbol{\beta}_j^*)^T (\boldsymbol{z}_j^* - \tilde{\boldsymbol{X}}_j^* \boldsymbol{\beta}_j^*) + \lambda \boldsymbol{\beta}_j^{*T} \boldsymbol{\beta}_j^* \right] - 2\boldsymbol{u}^T \sum_{j=1}^{k} \boldsymbol{\beta}_j^*. \tag{4.11}$$

Setting derivatives of (4.11) to zero gives

$$\frac{\partial \mathfrak{L}}{\partial \boldsymbol{\beta}_j^*} = 2\left[\tilde{\boldsymbol{X}}_j^{*T}(\tilde{\boldsymbol{X}}_j^*\boldsymbol{\beta}_j^* - \boldsymbol{z}_j^*) + \lambda\boldsymbol{\beta}_j^* - \boldsymbol{u}\right] = \mathbf{0}_{n+1}, \tag{4.12}$$

$$\frac{\partial \mathfrak{L}}{\partial \boldsymbol{u}} = \sum_{j=1}^{k}\boldsymbol{\beta}_j^* = \mathbf{0}_{n+1}. \tag{4.13}$$

From (4.12), we have $\boldsymbol{\beta}_j^* = (\tilde{\boldsymbol{X}}_j^{*T}\tilde{\boldsymbol{X}}_j^* + \lambda\boldsymbol{I})^{-1}(\tilde{\boldsymbol{X}}_j^{*T}\boldsymbol{z}_j^* + \boldsymbol{u})$. Combining with (4.13), we have $\boldsymbol{u} = -[\sum_{j=1}^k \boldsymbol{B}_j]^{-1}[\sum_{j=1}^k \boldsymbol{B}_j\tilde{\boldsymbol{X}}_j^*\boldsymbol{z}_j^*]$, where $\boldsymbol{B}_j = (\tilde{\boldsymbol{X}}_j^{*T}\tilde{\boldsymbol{X}}_j^* + \lambda\boldsymbol{I})^{-1}$. Plugging this into (4.12), together with the definition of $\boldsymbol{\beta}_j^*$, gives

$$\boldsymbol{\beta}_j = \boldsymbol{A}_j\left[\tilde{\boldsymbol{X}}^T\boldsymbol{W}_j\boldsymbol{z}_j - \left(\sum_{j=1}^{k}\boldsymbol{A}_j\right)^{-1}\left(\sum_{j=1}^{k}\boldsymbol{A}_j\tilde{\boldsymbol{X}}^T\boldsymbol{W}_j\boldsymbol{z}_j\right)\right], \tag{4.14}$$

where $\boldsymbol{A}_j = (\tilde{\boldsymbol{X}}^T\boldsymbol{W}_j\tilde{\boldsymbol{X}} + \lambda\boldsymbol{G})^{-1}$.

So far, we have focused on the standard case which treats all samples with equal weight. However, there could be situations that we want to give different weights on different subjects. For example, it could be more severe to misclassify an observation to a certain class than to other classes. Then it is natural to put a higher cost for that certain type of misclassification. This can be achieved by putting different weights on observations in different classes.

The multicategory weighted CLS classifier can be directly implemented with a simple modification. Let π_i be the weight we want to impose on the i-th observation. Then the loss function in (4.10) remains the same, except w_{ij} is replaced by $w_{ij}^* = \pi_i w_{ij}$. The rest of the algorithm remains the same.

4.5 Numerical Results

4.5.1 Simulation

To explore the performance of our proposed multicategory CLS classifier, we carry out some numerical analysis on the following multi-class examples. For Example 4.5.1.1, which is a 3-class problem used in Liu and Yuan, we generate 50 observations for training, 50 observations for tuning, and 10^6 observations for testing. For Examples 4.5.1.2 and 4.5.1.3, which have 6 and

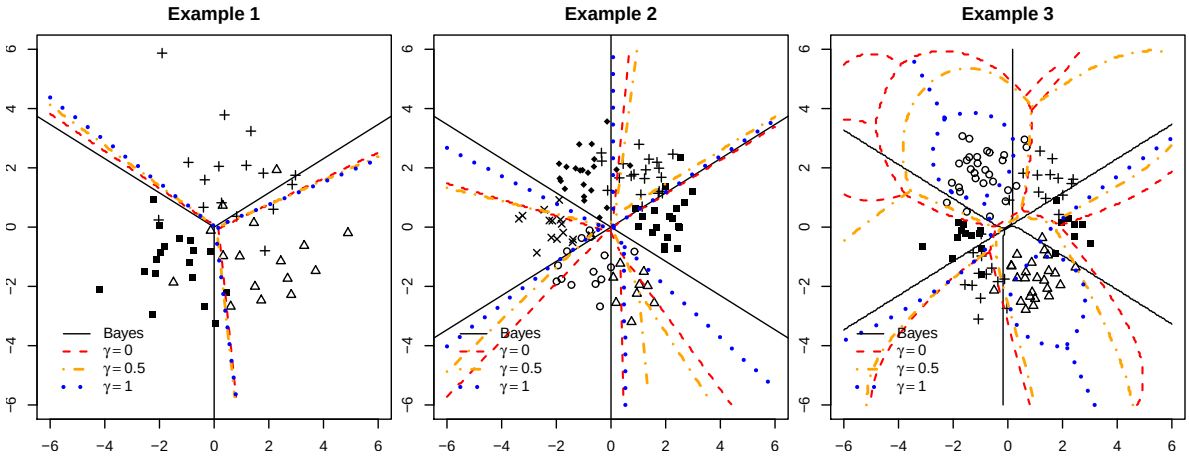

Figure 4.2: Scatter plots of typical datasets of Example 4.5.1, 4.5.2, and 4.5.3.

4 classes respectively, we generate 100 observations for each of training and tuning sets to ensure each training set has reasonable number of observations for every class. For testing, we generate 10^6 observations similarly to Example 4.5.1.1. A model is developed based on the training set, then the tuning set is used to choose λ among the set $\{2^{-16}, 2^{-15}, \cdots, 2^{15}\}$. With the selected model and λ, the misclassification rate is calculated based on the testing set. We repeat this procedure 100 times with $\gamma = 0.0, 0.1, 0.2, \cdots, 1.0$ to examine the effect of γ.

We also included the results of the case when γ is tuned among $\{0, 0.5, 1\}$ together with λ. For probability estimation, we train and tune the model in the same manner for $\gamma = 0, 0.5, 1$, then use the formulae (4.6)-(4.8) to estimate class probability. The probability estimation error, $\frac{1}{nk}\sum_{i=1}^{n}\sum_{j=1}^{k}|\hat{P}_j(\boldsymbol{x}) - P_j(\boldsymbol{x})|$, is calculated based on the testing set.

Example 4.5.1.1 We generate three-class data with

$$\begin{aligned} &P(Y=j) = 1/3, \text{ for } j = 1, \cdots, 3, \\ &P(\boldsymbol{X}|Y=j) \sim N(\boldsymbol{\mu}_j, 1.5^2\boldsymbol{I}_2), \text{ for } j = 1, \cdots, 3, \\ &\boldsymbol{\mu}_j = \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} -\sqrt{3} \\ -1 \end{pmatrix}, \begin{pmatrix} \sqrt{3} \\ -1 \end{pmatrix}, \text{ for } j = 1, \cdots, 3, \text{ respectively.} \end{aligned} \tag{4.15}$$

Since the Bayes boundary of this example is piecewise linear, linear learning can be sufficient. However, we added the results using the polynomial kernel of order 2 as well to further illustrate

the behavior of the multicategory CLS classifier. Moreover, we also report the performance of the RMSVM with the linear kernel for comparison.

Table 4.1: Estimated Test errors based on 100 replications for Example 4.5.1.1. The rows with tuned 1 and tuned 2 show the results when λ is tuned at the same time with γ among $\{0, 0.1, 0.2, \cdots, 1.0\}$, and among $\{0, 0.5, 1\}$, respectively. The Bayes error is 0.2043.

	Multicategory CLS		RMSVM	
γ	Linear	Poly	Linear	Poly
0.0	0.2275(0.0019)	0.2218(0.0016)	0.2821(0.0094)	0.2273 (0.0023)
0.1	0.2268(0.0017)	0.2226(0.0019)	0.2672(0.0080)	0.2286 (0.0024)
0.2	0.2248(0.0015)	0.2227(0.0018)	0.2527(0.0062)	0.2291 (0.0024)
0.3	0.2234(0.0015)	0.2230(0.0019)	0.2425(0.0051)	0.2315 (0.0028)
0.4	0.2211(0.0013)	0.2237(0.0019)	0.2370(0.0043)	0.2323 (0.0029)
0.5	0.2196(0.0012)	0.2248(0.0020)	0.2312(0.0036)	0.2337 (0.0031)
0.6	0.2182(0.0011)	0.2261(0.0021)	0.2282(0.0031)	0.2363 (0.0034)
0.7	0.2171(0.0013)	0.2270(0.0021)	0.2282(0.0032)	0.2390 (0.0034)
0.8	0.2162(0.0010)	0.2269(0.0018)	0.2285(0.0032)	0.2426 (0.0035)
0.9	0.2169(0.0011)	0.2296(0.0019)	0.2307(0.0037)	0.2546 (0.0051)
1.0	0.2180(0.0014)	0.2399(0.0033)	0.2493(0.0061)	0.3101 (0.0066)
tuned 1	0.2204(0.0013)	0.2255(0.0022)	0.2288(0.0029)	0.2344 (0.0032)
tuned 2	0.2215(0.0015)	0.2252(0.0022)	0.2340(0.0036)	0.2331 (0.0035)

The results are summarized in Table 4.1 and Figure 4.3. When the linear kernel is used, the test error decreases as γ increases. On the other hand, the polynomial kernel of order 2 gives the similar results except the test error becomes very high when $\gamma = 1.0$. Overall, the results are consistently good when γ is around 0.5. The probability estimation error has a similar pattern: it becomes the smallest when $\gamma = 0.5$. These results indicate that the multicategory CLS classifier gives the best result when γ is near 0.5, and the combination of two different versions of the squared loss functions is indeed better than both uncombined loss functions.

The performance of the RMSVM has a similar pattern to that of the multicategory CLS classifier in the sense that the classification error decreases as the value of γ goes bigger, but after some point it increases a bit so that γ in the middle works the best overall. Furthermore, the multicategory CLS classifier outperforms the RMSVM on this example.

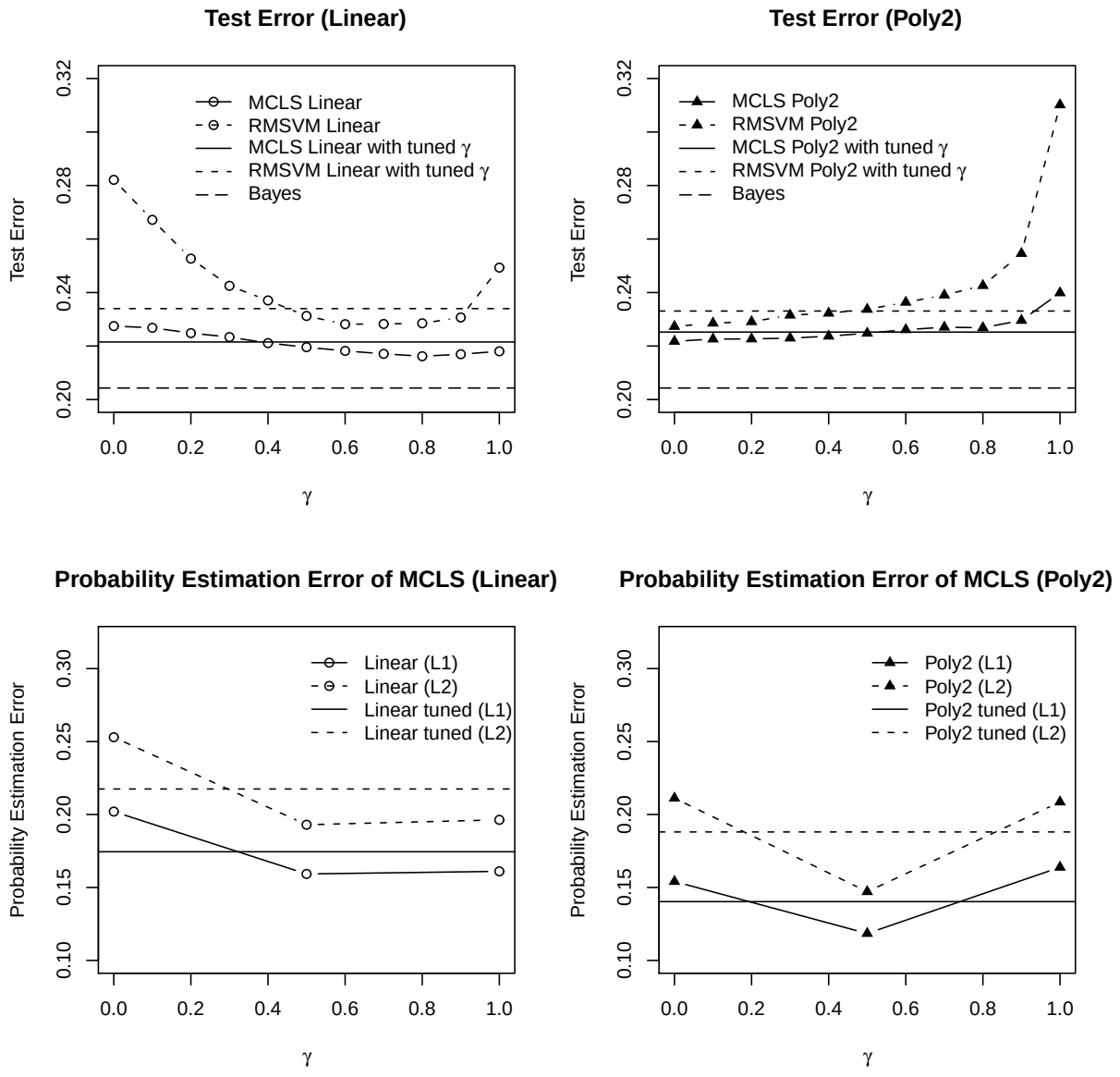

Figure 4.3: Left: Plot of the average test errors of the multicategory CLS classifier based on 100 replications with $\gamma = 0.0, 0.1, 0.2, \cdots, 1.0$ for Example 4.5.1.1. Right: Plot of the average probability estimation errors of the multicategory CLS classifier based on 100 replications with $\gamma = 0.0, 0.5$, and 1.0 for Example 4.5.1.1.

Table 4.2: Estimated Test errors based on 100 replications for Example 4.5.1.2. The rows with tuned 1 and tuned 2 show the results when λ is tuned at the same time with γ among $\{0, 0.1, 0.2, \cdots, 1.0\}$, and among $\{0, 0.5, 1\}$, respectively. The Bayes error is 0.0459 and 0.1538 when $\sigma = 0.5$ and $\sigma = 0.7$, respectively.

	$\sigma = 0.5$		$\sigma = 0.7$	
γ	Linear	Poly	Linear	Poly
0.0	0.2280 (0.0078)	0.1137 (0.0046)	0.3074 (0.0062)	0.2304 (0.0045)
0.1	0.2276 (0.0078)	0.1035 (0.0043)	0.3072 (0.0063)	0.2229 (0.0041)
0.2	0.2274 (0.0078)	0.0927 (0.0038)	0.3078 (0.0063)	0.2139 (0.0036)
0.3	0.2269 (0.0079)	0.0848 (0.0034)	0.3075 (0.0063)	0.2046 (0.0033)
0.4	0.2244 (0.0079)	0.0772 (0.0029)	0.3061 (0.0063)	0.1961 (0.0029)
0.5	0.2177 (0.0078)	0.0712 (0.0025)	0.3009 (0.0062)	0.1898 (0.0027)
0.6	0.1984 (0.0075)	0.0668 (0.0020)	0.2873 (0.0059)	0.1841 (0.0022)
0.7	0.1595 (0.0064)	0.0633 (0.0015)	0.2594 (0.0051)	0.1793 (0.0018)
0.8	0.1172 (0.0050)	0.0612 (0.0012)	0.2255 (0.0041)	0.1761 (0.0015)
0.9	0.0822 (0.0034)	0.0604 (0.0010)	0.1957 (0.0030)	0.1743 (0.0014)
1.0	0.0660 (0.0012)	0.0590 (0.0011)	0.1839 (0.0018)	0.1737 (0.0014)
tuned 1	0.0670 (0.0012)	0.0626 (0.0010)	0.1871 (0.0020)	0.1775 (0.0016)
tuned 2	0.0660 (0.0012)	0.0619 (0.0011)	0.1849 (0.0019)	0.1775 (0.0016)

Example 4.5.1.2 In this example, we generate six-class data with

$$
\begin{aligned}
&P(Y = j) = 1/6, \text{ for } j = 1, \cdots, 6, \\
&P(\boldsymbol{X}|Y = j) \sim N(\boldsymbol{\mu}_j, \sigma^2 \boldsymbol{I}_2), \text{ for } j = 1, \cdots, 6, \\
&\boldsymbol{\mu}_j = \begin{pmatrix} 1 \\ \sqrt{3} \end{pmatrix}, \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ -\sqrt{3} \end{pmatrix}, \begin{pmatrix} -1 \\ -\sqrt{3} \end{pmatrix}, \begin{pmatrix} -2 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ \sqrt{3} \end{pmatrix}, \text{ for } j = 1, \cdots, 6, \text{ respectively,}
\end{aligned} \tag{4.16}
$$

for $\sigma = 0.5$ and $\sigma = 0.7$. Similar to Example 4.5.1.1, the Bayes boundary is piecewise linear, but we report the results of the multicategory CLS classifier with both of linear and polynomial kernel of order 2 in Table 4.2 and Figure 4.4. Since the RMSVM runs into numerical problems on this example due to too high number of classes, we do not report the results of the RMSVM and the Two-step MCLS classifier.

Clearly, higher γ works better in this example regardless of the kernel choice. This indicates that the multicategory CLS classifier indeed improves the existing multi-class proximal SVM. The tuned methods give very reasonable performance.

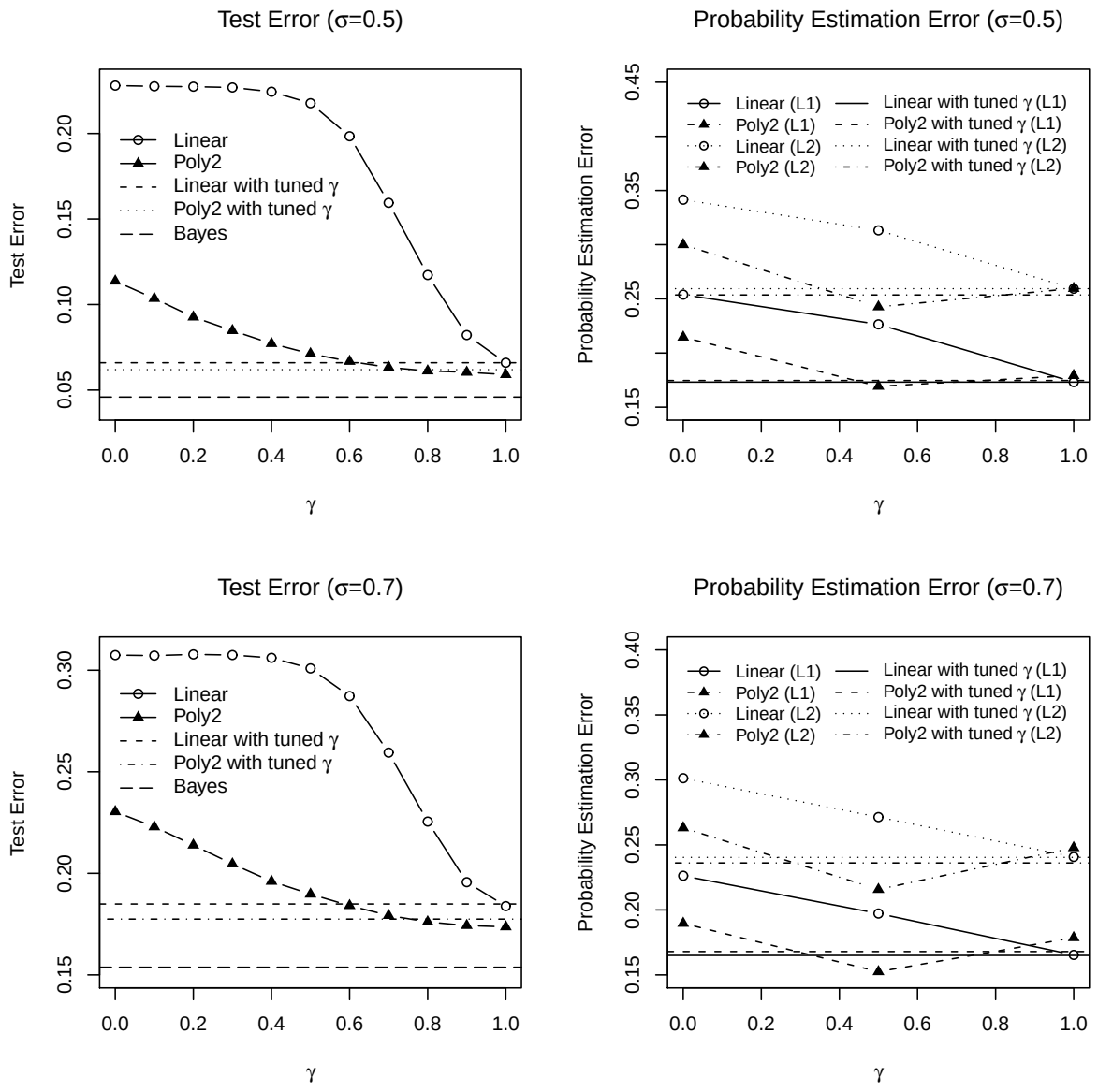

Figure 4.4: Left: Plot of the average test errors of the multicategory CLS classifier based on 100 replications with $\gamma = 0.0, 0.1, 0.2, \cdots, 1.0$ for Example 4.5.1.2. Here, the results with 'tuned γ' are the results when γ is tuned among $\{0, 0.5, 1\}$ along with λ. Right: Plot of the average probability estimation errors of the multicategory CLS classifier based on 100 replications with $\gamma = 0.0, 0.5$, and 1.0 for Example 4.5.1.2.

Table 4.3: Estimated Test errors based on 100 replications for Example 4.5.1.3. The rows with tuned 1 and tuned 2 show the results when λ is tuned at the same time with γ among $\{0, 0.1, 0.2, \cdots, 1.0\}$, and among $\{0, 0.5, 1\}$, respectively. The Bayes error is 0.0434 and 0.1450 when $\sigma = 0.5$ and $\sigma = 0.7$, respectively.

	$\sigma = 0.5$		$\sigma = 0.7$	
γ	Linear	Poly	Linear	Poly
0.0	0.2056 (0.0029)	0.2065 (0.0038)	0.0916 (0.0031)	0.0963 (0.0038)
0.1	0.2011 (0.0028)	0.2010 (0.0035)	0.0839 (0.0027)	0.0882 (0.0032)
0.2	0.1951 (0.0027)	0.1955 (0.0032)	0.0779 (0.0023)	0.0821 (0.0027)
0.3	0.1899 (0.0024)	0.1891 (0.0028)	0.0732 (0.0020)	0.0771 (0.0023)
0.4	0.1855 (0.0022)	0.1850 (0.0025)	0.0700 (0.0018)	0.0724 (0.0020)
0.5	0.1822 (0.0021)	0.1793 (0.0021)	0.0673 (0.0016)	0.0686 (0.0016)
0.6	0.1788 (0.0020)	0.1761 (0.0019)	0.0655 (0.0015)	0.0656 (0.0015)
0.7	0.1763 (0.0016)	0.1721 (0.0016)	0.0628 (0.0013)	0.0633 (0.0013)
0.8	0.1742 (0.0016)	0.1695 (0.0014)	0.0614 (0.0012)	0.0611 (0.0012)
0.9	0.1772 (0.0016)	0.1689 (0.0015)	0.0606 (0.0012)	0.0597 (0.0010)
1.0	0.2571 (0.0035)	0.1662 (0.0013)	0.1476 (0.0031)	0.0599 (0.0010)
tuned 1	0.1795 (0.0022)	0.1715 (0.0015)	0.0634 (0.0014)	0.0619 (0.0012)
tuned 2	0.1851 (0.0021)	0.1714 (0.0014)	0.0699 (0.0017)	0.0615 (0.0011)

Example 4.5.1.3 we generate four-class data with

$$
\begin{aligned}
&P(Y = j) = 1/4, \text{ for } j = 1, \cdots, 4, \\
&P(\boldsymbol{X}|Y = 1) \sim 0.5N((1, \sqrt{3})^T, \sigma^2 \boldsymbol{I}_2) + 0.5N((-1, -\sqrt{3})^T, \sigma^2 \boldsymbol{I}_2) \\
&P(\boldsymbol{X}|Y = 2) \sim 0.5N((2, 0)^T, \sigma^2 \boldsymbol{I}_2) + 0.5N((-2, 0)^T, \sigma^2 \boldsymbol{I}_2) \\
&P(\boldsymbol{X}|Y = j) \sim N(\boldsymbol{\mu}_j, \sigma^2 \boldsymbol{I}_2), \text{ for } j = 3, 4, \\
&\boldsymbol{\mu}_3 = \begin{pmatrix} 1 \\ -\sqrt{3} \end{pmatrix}, \boldsymbol{\mu}_4 = \begin{pmatrix} -1 \\ \sqrt{3} \end{pmatrix},
\end{aligned}
\tag{4.17}
$$

for $\sigma = 0.5$ and $\sigma = 0.7$. Due to the structure of this dataset, linear learning will not be suitable, thus we consider the Gaussian kernel and the polynomial kernel of order 2.

The results are summarized in Table 4.3 and Figure 4.6. For each kernel, higher γ generally works better, but the highest $\gamma = 1$ gives the worst performance. Hence, we can conclude that it is safe to use γ that is somewhere in the middle. Furthermore, tuning γ can be a reasonable way to choose γ.

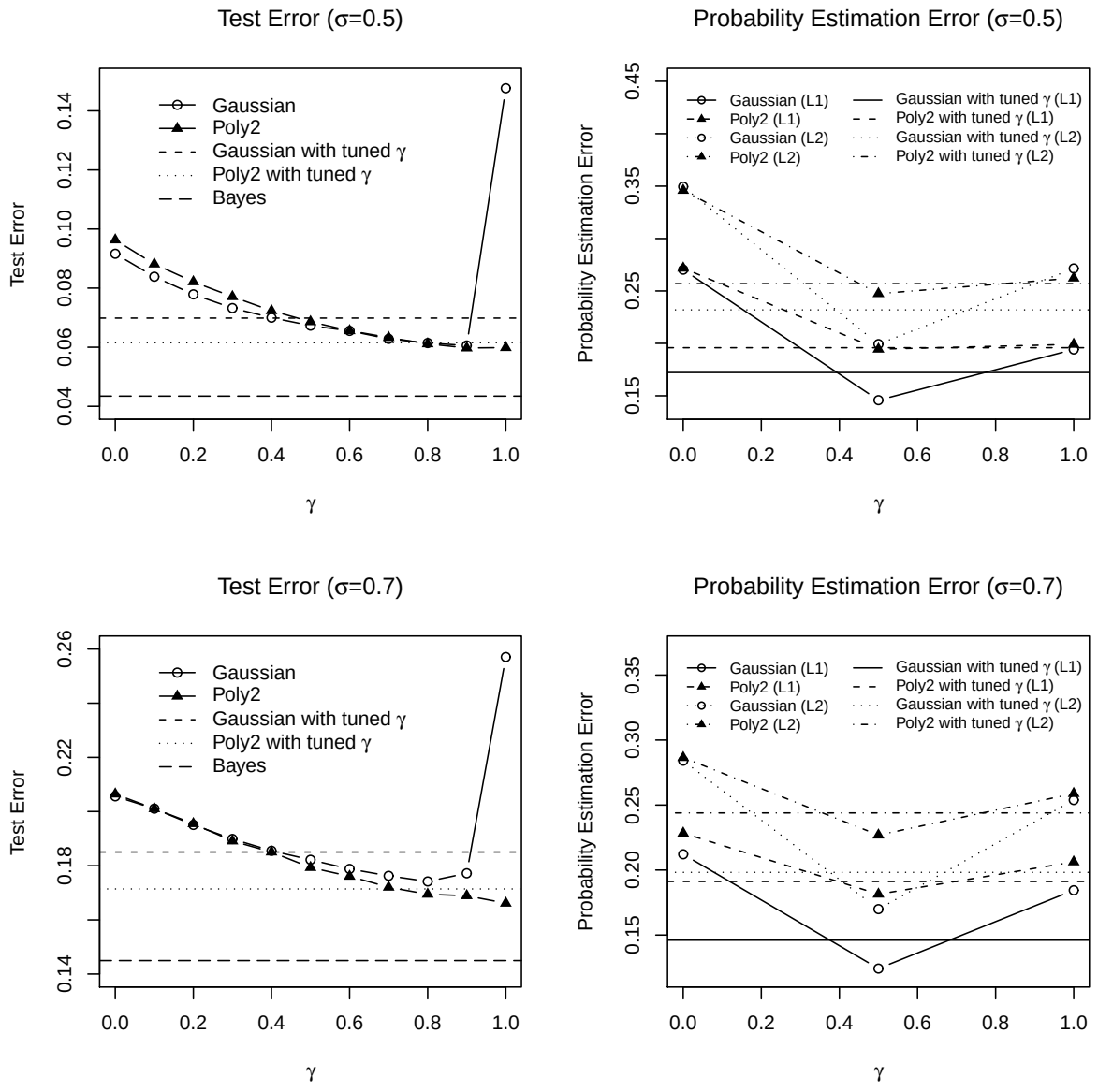

Figure 4.5: Left: Plot of the average test errors of the multicategory CLS classifier based on 100 replications with $\gamma = 0.0, 0.1, 0.2, \cdots, 1.0$ for Example 4.5.1.3. Right: Plot of the average probability estimation errors of the multicategory CLS classifier based on 100 replications with $\gamma = 0.0, 0.5$, and 1.0 for Example 4.5.1.3.

4.5.2 Real Application

To further demonstrate the performance of the multicategory CLS classifiers, we use the Leukemia data set described in Golub et al. (1999). The Leukemia data set consists of 7129 gene expression values of 38 examples in the training set and 34 examples in the testing set. In the original paper, they use gene expression data to classify subjects into two types of leukemias, ALL (acute lymphoblastic leukemia) and AML (acute myeloid leukemia). Since ALL type can be further divided into B-cell and T-cell ALLs (ALLB and ALLT), we can perform three-category classification on this data set. The training set contains 19, 8, 11 subjects of ALLB, ALLT, and AML types, and the testing set has 19, 1, 14 subjects of ALLB, ALLT, and AML types. Out of 7129 genes, we choose 40 genes by prescreening procedure using the ratios of between-groups to within-groups sum of squares of the genes. We choose the tuning parameter λ by 5-fold cross validation on the training set.

Similar to the results of the original binary problem, only one or no observation is misclassified for any value of γ. Hence rather than looking at the misclassification rate, it might be more interesting to look at the performance of the class probability estimation. As shown in Figure 4.7, the estimated class probabilities agree with the true memberships of the observations, except when $\gamma = 1$, two patients are misclassified (ALLT patient and the first AML patient in the graph). According to heatmap of this data set with selected 40 genes in Figure 4.6, these are the ones that are hard to classify and the estimated probability reflects the ambiguity of the gene expressions of those two patients. In real application, estimated class probabilities such as the one shown here could be more helpful than just class membership prediction.

4.6 Summary and Discussion

In this section, we propose the multicategory CLS classifier which makes use of a convex combination of two square loss functions. The CLS classifier is shown to have Fisher consistency, conditional probability estimation, efficient computation, ability to handle data with high dimension and large number of classes. Through simulated examples, the CLS classifier is shown to outperform proximal SVMs and multicategory SVMs.

The current CLS classifier makes use of the L_2 penalty as in the ridge regression. Although L_2

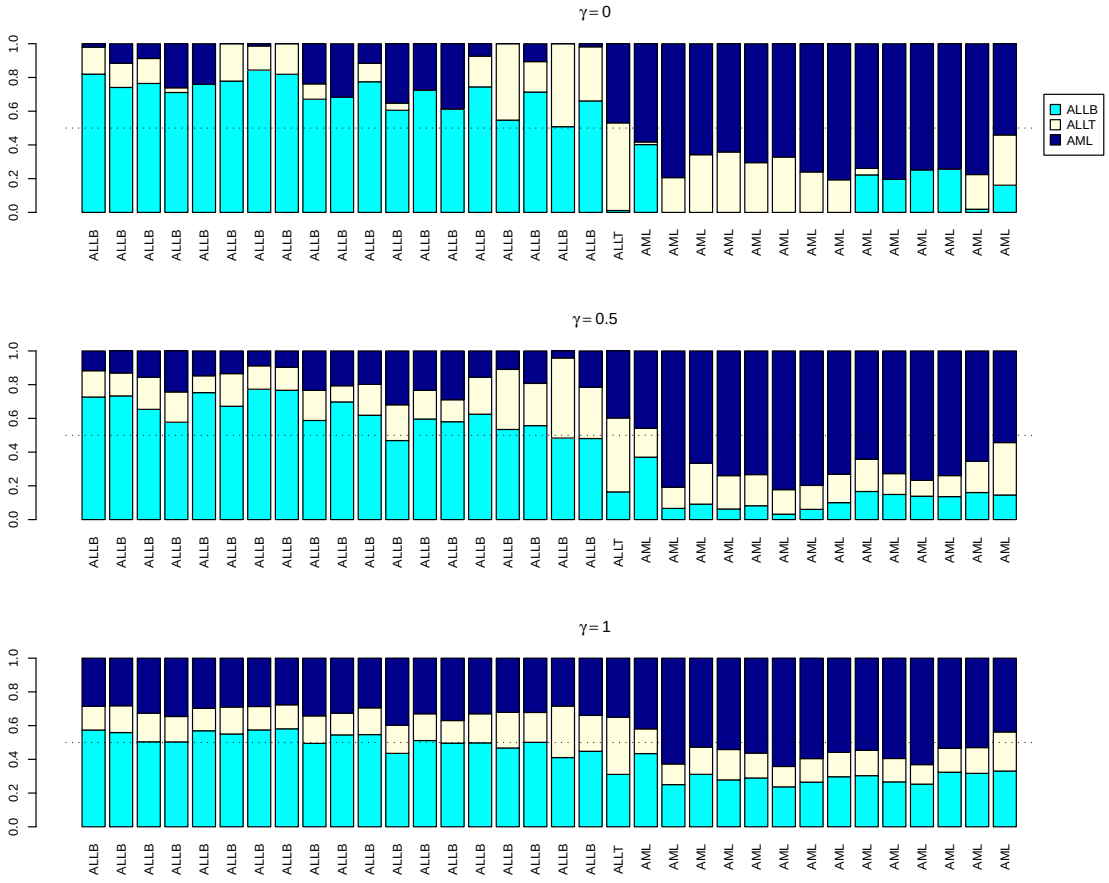

Figure 4.6: Plot of the estimated class probabilities for subjects in the testing set of the Leukemia data. The heights of cyan, bright yellow, and dark green bars stand for the estimated probability of ALLB, ALLT, and AML, respectively.

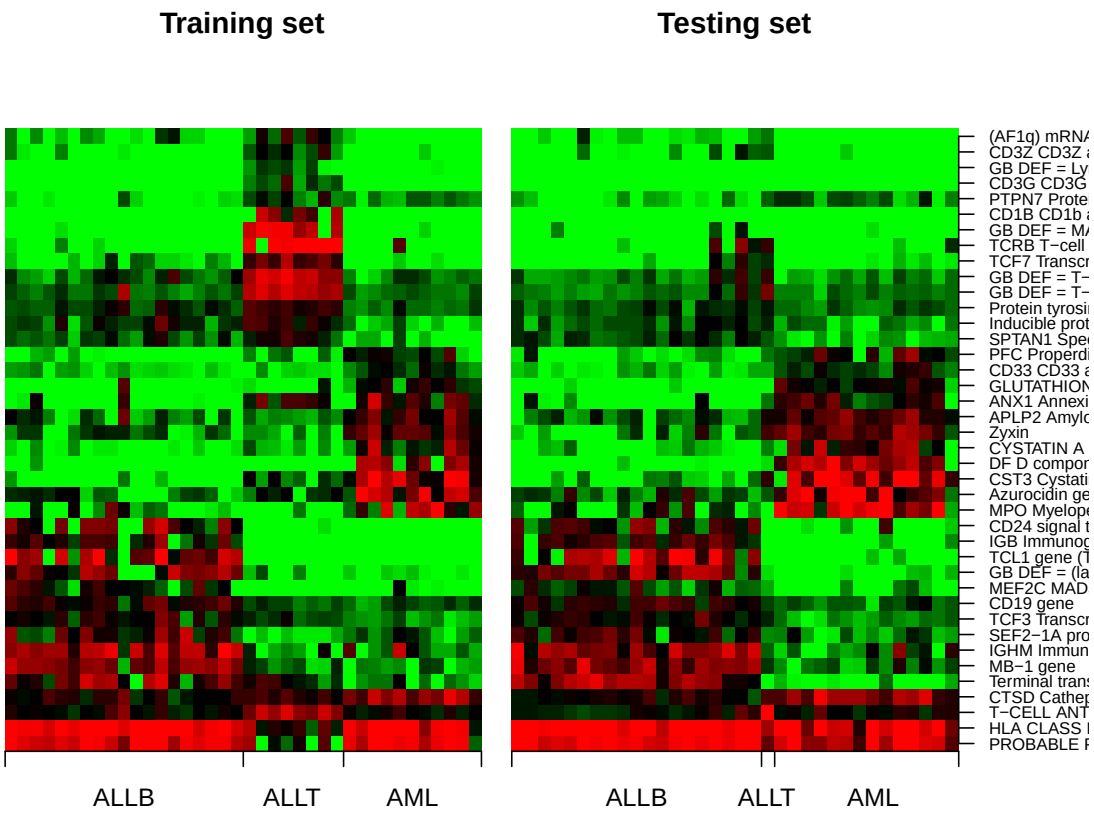

Figure 4.7: Heat maps of the Leukemia data. The left panel is for the training set and the right panel is for the testing set. The red and green colors represent high and low expression values respectively. The subjects are displayed in the same order as the Figure 4.

penalty works well overall, it does not perform automatic variable selection. For high dimensional data with many noise variables, other penalties such as L_1 penalty (Zhu et al., 2004; Wang and Shen, 2007) can be more useful to deliver sparse models. It will be interesting to explore sparse CLS classifiers for high dimensional data analysis.

4.7 Proofs

Proof of Theorem 7

First, observe that

$$\begin{aligned} E[L(\boldsymbol{f}(\boldsymbol{x}), y)|\boldsymbol{X} = \boldsymbol{x}] &= \gamma \textstyle\sum_{j=1}^k g(f_j(\boldsymbol{x}))P_j(\boldsymbol{x}) + (1-\gamma)\sum_{l=1}^k [(\sum_{j=1}^k g(-f_j(\boldsymbol{x}))) - g(-f_l(\boldsymbol{x}))]P_l(\boldsymbol{x}) \\ &= \gamma \textstyle\sum_{j=1}^k g(f_j(\boldsymbol{x}))P_j(\boldsymbol{x}) + (1-\gamma)\sum_{j=1}^k g(-f_j(\boldsymbol{x})) - (1-\gamma)\sum_{l=1}^k g(-f_l(\boldsymbol{x}))P_l(\boldsymbol{x}) \\ &= \gamma \textstyle\sum_{j=1}^k g(f_j(\boldsymbol{x}))P_j(\boldsymbol{x}) + (1-\gamma)\sum_{j=1}^k (1-P_j(\boldsymbol{x}))g(-f_j(\boldsymbol{x})). \end{aligned} \tag{4.18}$$

To obtain the minimizer of (4.18) subject to $\sum_j^k f_j(\boldsymbol{x}) = 0$, we use the Lagrange multiplier method. For convenience, let $f_j = f_j(\boldsymbol{x})$. The corresponding Lagrange primal is

$$\mathfrak{L}(\boldsymbol{f}, \alpha) = \gamma \sum_{j=1}^k g(f_j)P_j(\boldsymbol{x}) + (1-\gamma)\sum_{j=1}^k (1-P_j(\boldsymbol{x}))g(-f_j) - \alpha \sum_{j=1}^k f_j.$$

Setting the first derivatives to zero gives,

$$\begin{aligned} \tfrac{\partial \mathfrak{L}}{\partial f_1} &= \gamma P_1(\boldsymbol{x})g'(f_1) - (1-\gamma)(1-P_1(\boldsymbol{x}))g'(-f_1) - \alpha = 0 \\ &\vdots \\ \tfrac{\partial \mathfrak{L}}{\partial f_k} &= \gamma P_k(\boldsymbol{x})g'(f_k) - (1-\gamma)(1-P_k(\boldsymbol{x}))g'(-f_k) - \alpha = 0 \\ \tfrac{\partial \mathfrak{L}}{\partial \alpha} &= \textstyle\sum_{j=1}^k = 0. \end{aligned}$$

Therefore, we have

$$-\gamma P_j(\boldsymbol{x})g'(f_j) + (1-\gamma)(1-P_j(\boldsymbol{x}))g'(-f_j) = -\gamma P_l(\boldsymbol{x})g'(f_l) + (1-\gamma)(1-P_l(\boldsymbol{x}))g'(-f_l) \text{ for any } j \neq l \tag{4.19}$$

Without loss of generality, we can assume that $\operatorname{argmax}_j P_j(\boldsymbol{x}) = 1$. Suppose $f_j \geqslant f_1$ for some $j \neq 1$. Since $g'(u) < 0$ and $g''(u) > 0$, we have $g'(f_1) \leqslant g'(f_j) < 0$ and $0 > g'(-f_1) \geqslant$

$g'(-f_j)$. This implies $\gamma P_1(\boldsymbol{x})g'(f_1) < \gamma P_j(\boldsymbol{x})g'(f_j)$ and $(1-\gamma)(1-P_1(\boldsymbol{x}))g'(-f_1) > (1-\gamma)(1-P_j(\boldsymbol{x}))g'(-f_j)$, which gives $-\gamma P_1(\boldsymbol{x})g'(f_1) + (1-\gamma)(1-P_1(\boldsymbol{x}))g'(-f_1) > -\gamma P_j(\boldsymbol{x})g'(f_j) + (1-\gamma)(1-P_j(\boldsymbol{x}))g'(-f_j)$. This contradicts to (4.19). Thus we conclude $f_1 > f_j$ for any $j \neq 1$ when $\operatorname{argmax}_j P_j(\boldsymbol{x}) = 1$. This completes the proof. □

Bibliography

L. T. H. An and P. D. Tao. Solving a class of linearly constrained indefinite quadratic problems by d.c. algorithms. *Journal of Global Optimization*, 11:253–285, 1997.

R. Bahadur. A note on quantiles in large samples. *Annals of Mathematical Statistics*, 37:577–580, 1966.

P. Bartlett, M. Jordan, and J. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101:138–156, 2006.

A. Bhattacharjee, W. G. Richards, J. Staunton, C. Li, S. Monti, P. Vasa, C. Ladd, J. Beheshti, R. Bueno, M. Gillette, M. Loda, G. Weber, E. J. Mark, E. S. Lander, W. Wong, B. E. Johnson, T. R. Golub, D. J. Sugarbaker, and M. Meyerson. Classification of human lung carcinomas by mrna expression profiling reveals distinct adenocarcinoma subclasses. *Proceedings of the National Academy of Sciences of the United States of America*, 98(24):13790–13795, 2001.

A. M. Bianco and V. J. Yohai. Robust estimation in the logistic regression model. In H Rieder, editor, *Robust Statistics, Data Analysis, and Computer Intensive Methods*, volume 109 of *Lecture Notes in Statistics*. Springer-Verlag, New York, 1996.

H. Bondell. Minimum distance estimation for the logistic regression model. *Biometrika*, 92: 724–731, 2005.

E. Bredensteiner and K. Bennett. Multicategory classification by support vector machines. *Computational Optimizations and Applications*, 12:53–79, 1999.

E. Cantoni and E. Ronchetti. Robust inference for generalized linear models. *Journal of the American Statistical Association*, 96(455):1022–1030, 2001.

R. J. Carroll and S. Pederson. On robustness in the logistic regression model. *Journal of the Royal Statistical Society. Series B (Methodological)*, 55:693–706, 1993.

P. Chaudhuri. Nonparametric estimates of regression quantiles and their local bahadur representation. *The Annals of Statistics*, 19:760–777, 1991.

R. Collobert, F. Sinz, J. Weston, and L. Bottou. Large scale transductive svms. *Journal of Machine Learning Research*, 7:1687–1712, September 2006.

J. B. Copas. Binary regression models for contaminated data (with discussion). *Journal of the Royal Statistical Society. Series B (Methodological)*, 50:225–265, 1988.

K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001.

P. Craven and G. Wahba. Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross-validation. *Numerische Mathematik*, 31(4):377–403, 1979.

N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

C. Croux and G. Haesbroeck. Implementing the bianco and yohai estimator for logistic regression. *Computational Statistics and Data Analysis*, 44:273–295, 2003.

S. Dudoit, J. Fridly, and T. P. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 97: 77–87, 2002.

J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and it oracle properties. *Journal of American Statistical Association*, 96:1348–1360, 2001.

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

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

G. Fung and O. L. Mangasarian. Proximal support vector machine classifiers. In *Proceedings KDD-2001: Knowledge Discovery and Data Mining*, pages 77–86, 2001.

T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, and M. Caligiuri. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5(Oct):1391–1415, 2004.

T. J. Hastie and R. J. Tibshirani. *Generalized Additive Models*. Chapman and Hall, 1990.

R. Horst and N. V. Thoai. Dc programming: overview. *Journal of Optimization Theory and Applications*, 103:1–41, 1999.

D. Hunter and R. Li. Variable selection using mm algorithms. *The Annals of Statistics*, 33: 1617–1642, 2005.

G. Kimeldorf and G. Wahba. Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33:82–95, 1971.

J.-Y. Koo, Y. Lee, Y. Kim, and C. Park. A bahadur representation of the linear support vector machine. *Journal of Machine Learning Research*, 9:1343–1368, 2008.

W. S. Krasker and R. E. Welsch. Efficient bounded-influence regression estimation. *Journal of the American Statistical Association*, 77(379):595–604, 1982.

H. R. Künsch, L. A. Stefanski, and R. J. Carroll. Conditionally unbiased bounded-influence estimation in general regression models, with applications to generalized linear models. *Journal of the American Statistical Association*, 84(406):460–466, 1989.

le Cessie and van Houwelingen. Ridge estimators in logistic regression. *Applied Statistics*, 41: 191–201, 1992.

Y. Lee, Y. Lin, and G. Wahba. Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association*, 99:67–81, 2004.

X. Lin, G. Wahba, D. Xiang, F. Gao, R. Klein, and B. Klein. Smoothing spline ANOVA models for large data sets with Bernoulli observations and the randomized GACV. *The Annals of Statistics*, 28(6):1570–1600, 2000.

Y. Lin. Support vector machines and the bayes rule in classification. *Data Mining and Knowledge Discovery*, 6:259–275, 2002.

Y. Lin. A note on margin-based loss functions in classification. *Stat. and Prob. Letters*, 68: 73–82, 2004.

Y. Lin and H. H. Zhang. Component selection and smoothing in smoothing spline analysis of variance models – cosso. *Annals of Statistics*, 34:2272–2297, 2006.

Y. Lin, Y. Lee, and G. Wahba. Support vector machines for classification in nonstandard situations. *Machine Learning*, 46:191–202, 2002.

Y. Liu. Unbiased estimate of generalization error and model selection in neural network. *Neural Networks*, 8:215–219(5), 1995.

Y. Liu. Fisher consistency of multicategory support vector machines. *Eleventh International Conference on Artificial Intelligence and Statistics*, pages 289–296, 2007.

Y. Liu and X. Shen. Multicategory psi-learning. *Journal of the American Statistical Association*, 101:500–509, 2006.

Y. Liu and M Yuan. Reinforced multicategory support vector machines. Under review.

Y. Liu, X. Shen, and H. Doss. Multicategory psi-learning and support vector machine: computational tools. *Journal of Comput. and Graphical Statistics*, 14:219–236, 2005.

Y. Liu, D. N. Hayes, A. Nobel, and J. S. Marron. Statistical significance of clustering for high dimension low sample size data. *Journal of the American Statistical Association*, 103: 1281–1293, 2008.

J. S. Marron, M. Todd, and J. Ahn. Distance-weighted discrimination. *Journal of the American Statistical Association*, 102(480):1267–1271, 2007.

P. McCullagh and J. Nelder. *Generalized Linear Models*. CAHPMAN & HALL/CRC, 1989.

S. Morgenthaler. Least-absolute-deviations fits for generalized linear models. *Biometrika*, 79(4): 747–754, 1992.

M. Y. Park and T. J. Hastie. Penalized logistic regression for detecting gene interactions. *Biostatistics*, page kxm010, 2007. doi: 10.1093/biostatistics/kxm010. URL `http://biostatistics.oxfordjournals.org/cgi/content/abstract/kxm010v1`.

D. Pollard. Asymptotics for least absolute deviation regression estimators. *Econometric Theory*, 7(2):186–199, 1991.

D. Pregibon. Resistant fits for some commonly used logistic models with medical applications. *Biometrics*, 38(2):485–498, 1982.

X. Shen, G.C. Tseng, X. Zhang, and W.H. Wong. On psi-learning. *Journal of the American Statistical Association*, 98:724–734, 2003.

L. A. Stefanski, R. J. Carroll, and D. Ruppert. Optimally hounded score functions for generalized linear models with applications to logistic regression. *Biometrika*, 73(2):413–424, 1986.

J. A. K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Process. Lett.*, 9(3):293–300, 1999.

Y. Tang and H. H. Zhang. Multiclass proximal support vector machines. *Journal of Computational and Graphical Statistics*, 15:339–355(17), 2006.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996.

V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

G. Wahba. Support vector machines, reproducing kernel hilbert spaces and the randomized GACV. In *Advances in Kernel Methods Support Vector Learning*, pages 69–88. MIT Press, 1999.

J. Wang, X. Shen, and Y. Liu. Probability estimation for large margin classifiers. *Biometrika*, 95(1):149–167, 2007.

L. Wang and X. Shen. On L_1-norm multiclass support vector machines. *Journal of the American Statistical Association*, 102(478):583–594, 2007.

J. Weston. Extensions to the support vector method. Technical report, Royal Holloway, University of London, 1999.

J. Weston and C. Watkins. Support vector machines for multi-class pattern recognition. In M. Verleysen, editor, *Proceedings of the 7th European Symposium on Artificial Neural Networks (ESANN-99)*, pages 219–224. Bruges, Belgium, 1999.

Y. Wu and Y. Liu. Robust truncated hinge loss support vector machines. *Journal of the American Statistical Association*, 102:974–983, 2007.

D. Xiang and G. Wahba. A generalized approximate cross validation for smoothing splines with non-gaussian data. *Statistica Sinica*, 6:675–692, 1996.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68:49–67, 2006.

H. H. Zhang. Variable selection for support vector machines via smoothing spline ANOVA. *Statistica Sinica*, 16:659–674, 2006.

H. H. Zhang and W. Lu. Adaptive lasso for cox's proportional hazards model. *Biometrika*, 94: 691–703, 2007.

H. H. Zhang, J. Ahn, X. Lin, and C. Park. Gene selection using support vector machines with nonconvex penalty. *Bioinformatics*, 22:88–95, 2006.

J. Zhu and T. Hastie. Kernel logistic regression and the import vector machine. *Journal of Computational and Graphical Statistics*, 14:185–205, 2005.

J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani. 1-norm support vector machines. *Neural Information Processing Systems*, 16, 2004.

H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101:1418–1429, 2006.