Available online at www.sciencedirect.com
ScienceDirect
Computational Statistics & Data Analysis 52 (2007) 452–470
COMPUTATIONAL STATISTICS & DATA ANALYSIS
www.elsevier.com/locate/csda

High-dimensional pseudo-logistic regression and classification with applications to gene expression data

Chunming Zhang*, Haoda Fu, Yuan Jiang, Tao Yu

Department of Statistics, 1300 University Avenue, University of Wisconsin, Madison, WI 53706, USA

Available online 28 December 2006

Abstract

High dimension low sample size data, like the microarray gene expression levels, pose numerous challenges to conventional statistical methods. In the particular case of binary classification, some classification methods, such as the support vector machine (SVM), can efficiently deal with high-dimensional predictors, but lacks the accuracy in estimating the probability of membership of a class. In contrast, the traditional logistic regression (TLR) effectively estimates the probability of class membership for data with low-dimensional inputs, but does not handle high-dimensional cases. The study bridges the gap between SVM and TLR by their loss functions. Based on the proposed new loss function, a pseudo-logistic regression and classification approach which simultaneously combines the strengths of both SVM and TLR is also proposed. Simulation evaluations and real data applications demonstrate that for low-dimensional data, the proposed method produces regression estimates comparable to those of TLR and penalized logistic regression, and that for high-dimensional data, the new method possesses higher classification accuracy than SVM and, in the meanwhile, enjoys enhanced computational convergence and stability.

Keywords: Bayes optimal rule; Large p and small n data; Logistic regression; Loss function; Support vector machine

1. Introduction

Technological invention and information advancement have revolutionized scientific research and technological development. Many sophisticated large-scale data sets have recently been collected. These new data sets and streams pose numerous challenges to conventional statistical or data mining methods due to not only the massive size, but also the high dimensionality.

In this paper, we focus on high dimension low sample size data, the so-called large p small n data, with binary class label responses. Notable examples include clinical assessment of tumor types for microarray gene expression data, in which the number of variables (genes) far exceeds the number of samples (arrays). The traditional logistic regression (TLR) method effectively estimates the probability of class membership for large n small p data, but does not handle data sets with high-dimensional predictors. Besides, a monotone likelihood problem will occur when the predictors are fully separable (Firth, 1993). In that case, logistic regression will give unreliable estimates. See Albert and Anderson (1984) and Santner and Duffy (1986) for details.

* Corresponding author. Tel.: +1 608 262 0084; fax: +1 608 262 0032.
E-mail addresses: cmzhang@stat.wisc.edu (C. Zhang), fuhaoda@stat.wisc.edu (H. Fu), jiangy@stat.wisc.edu (Y. Jiang), yutao@stat.wisc.edu (T. Yu).

0167-9473/$ - see front matter © 2007 Published by Elsevier B.V.
doi:10.1016/j.csda.2006.12.033

On the other hand, the support vector machine (SVM) has emerged as a powerful pattern classification tool for high-dimensional data. By means of the dual representation, SVM translates an optimization problem of p-variables into the counterpart of n-variables. This characteristic enables SVM to efficiently deal with high-dimensional predictors. Refer to Vapnik (1996) and Cristianini and Shawe-Taylor (2000), among many others, for details. Nonetheless, unlike the logistic regression, SVM lacks the accuracy in estimating the probability of membership for each class. Therefore, SVM is less appropriate to estimate the class probability, which is of significant importance in various scientific disciplines.

In this paper, we aim to develop a high-dimensional regression and classification method which simultaneously combines the strengths of both SVM and TLR. To achieve this goal, we bridge the gap between SVM and TLR by their loss functions. Based on our proposed new loss function, we further propose a pseudo-logistic regression (PsLR) and classification approach which integrates the classification ability of SVM and the regression capability of TLR. Simulation evaluations and real data applications demonstrate that for low-dimensional data, the proposed method produces regression estimates comparable to those of TLR and penalized logistic regression (PeLR) (Eilers et al., 2001), and that for high-dimensional data, the new method possesses higher classification accuracy than SVM and, in the meanwhile, enjoys enhanced computational convergence and stability. As will be discussed in Section 3.2, the PeLR when applied to high-dimensional data, reduces the size of the estimating equations, but could not genuinely resolve the problems of computational instability and solution non-uniqueness. In contrast, our proposed method effectively overcomes these problems.

This paper is organized as follows. In Section 2, we review TLR and SVM, and connect them by their loss functions. In Section 3, we propose the PsLR method. In Section 4, we present some property of PsLR and propose a bias correction procedure for PsLR estimates. We apply our method to simulated data in Section 5 and real data sets in Section 6. Section 7 concludes this paper by a discussion. All detailed derivations are postponed to the Appendix.

2. Logistic regression and SVM

In this section, we start by reviewing TLR and SVM. After that, we will connect these two methods by their loss functions, which motivate the proposed PsLR method.

2.1. Logistic regression

Let $Y \in \{0, 1\}$ indicate the class label of a sample and $\mathrm{X} = (X_1, \ldots, X_p)^{\mathrm{T}}$ be the vector of explanatory variables. Define the conditional mean response function by $m(\mathrm{x}) = P(Y = 1|\mathrm{X} = \mathrm{x})$ and the canonical parameter by $\theta(\mathrm{x}) = \ln[m(\mathrm{x})/\{1 - m(\mathrm{x})\}]$. In TLR, it is assumed that

$$\theta(\mathrm{x}) = \beta_0 + \mathrm{x}^{\mathrm{T}}\boldsymbol{\beta}, \tag{2.1}$$

where β_0 and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^{\mathrm{T}}$ are unknown parameters.

For independent samples $\{(\mathrm{x}_i, y_i)\}_{i=1}^n$ drawn from (X, Y), the maximum likelihood estimates of β_0 and $\boldsymbol{\beta}$ are obtained from minimizing the negative conditional log-likelihood function

$$\begin{aligned}\ell(\widetilde{\boldsymbol{\beta}}) &= -\sum_{i=1}^{n}[y_i \ln\{m(\mathrm{x}_i)\} + (1 - y_i)\ln\{1 - m(\mathrm{x}_i)\}] \\ &= -\sum_{i=1}^{n}[y_i\widetilde{\mathrm{x}}_i^{\mathrm{T}}\widetilde{\boldsymbol{\beta}} - \ln\{1 + \exp(\widetilde{\mathrm{x}}_i^{\mathrm{T}}\widetilde{\boldsymbol{\beta}})\}],\end{aligned} \tag{2.2}$$

where $\widetilde{\mathrm{x}}_i = (1, \mathrm{x}_i^{\mathrm{T}})^{\mathrm{T}}$ and $\widetilde{\boldsymbol{\beta}} = (\beta_0, \boldsymbol{\beta}^{\mathrm{T}})^{\mathrm{T}}$. For computational implementation, it is customary to use the Newton–Raphson algorithm which requires the score vector

$$\frac{\partial\ell(\widetilde{\boldsymbol{\beta}})}{\partial\widetilde{\boldsymbol{\beta}}} = -\sum_{i=1}^{n}\widetilde{\mathrm{x}}_i\{y_i - m(\mathrm{x}_i)\},$$

and the Hessian matrix

$$\frac{\partial^2 \ell(\widetilde{\boldsymbol{\beta}})}{\partial \widetilde{\boldsymbol{\beta}} \partial \widetilde{\boldsymbol{\beta}}^{\mathrm{T}}} = \sum_{i=1}^{n} \widetilde{\mathrm{x}}_i \widetilde{\mathrm{x}}_i^{\mathrm{T}} m(\mathrm{x}_i)\{1 - m(\mathrm{x}_i)\}.$$

Starting from an initial value $\widetilde{\boldsymbol{\beta}}^{(0)}$, the estimate of $\widetilde{\boldsymbol{\beta}}$ can be obtained from the iterations

$$\widetilde{\boldsymbol{\beta}}^{(k+1)} = \widetilde{\boldsymbol{\beta}}^{(k)} - \left\{ \frac{\partial^2 \ell(\widetilde{\boldsymbol{\beta}})}{\partial \widetilde{\boldsymbol{\beta}} \partial \widetilde{\boldsymbol{\beta}}^{\mathrm{T}}} \bigg|_{\widetilde{\boldsymbol{\beta}} = \widetilde{\boldsymbol{\beta}}^{(k)}} \right\}^{-1} \frac{\partial \ell(\widetilde{\boldsymbol{\beta}})}{\partial \widetilde{\boldsymbol{\beta}}} \bigg|_{\widetilde{\boldsymbol{\beta}} = \widetilde{\boldsymbol{\beta}}^{(k)}}, \quad k = 0, 1, \ldots .$$

Denote the resulting estimates by $\widehat{\beta}_0$ and $\widehat{\boldsymbol{\beta}}$. The mean regression function can then be estimated by $\widehat{m}(\mathrm{x}) = 1/\{1 + \exp(-\widehat{\beta}_0 - \mathrm{x}^{\mathrm{T}}\widehat{\boldsymbol{\beta}})\}$. More details of logistic regression can be found in McCullagh and Nelder (1989).

Logistic regression offers the advantage of simultaneously estimating the probabilities $m(\mathrm{x})$ and $1 - m(\mathrm{x})$ of class membership and classifying data sets with low-dimensional predictors. That is, for a sample with input x, the predicted class label is given by

$$I\{\widehat{m}(\mathrm{x}) > \tfrac{1}{2}\},$$

where $I(\cdot)$ is an indicator function. On the other hand, we notice that the Hessian matrix in the preceding Newton–Raphson algorithm is a $(p+1) \times (p+1)$ matrix. For high-dimensional data with dimension $p+1$ greater than the sample size n, the Hessian matrix will not have full rank. In that case, logistic regression will fail to produce reliable regression estimates and classification outputs.

2.2. Support vector machine

We briefly review the standard (linear) SVM for binary classification. Suppose that we have a training set of samples $(\mathrm{x}_1, y_1^*), \ldots, (\mathrm{x}_n, y_n^*)$, where $y_i^* = -1$ or 1 depends on whether its class label is 0 or 1. Set

$$f(\mathrm{x}) = b_0 + \mathrm{x}^{\mathrm{T}}\mathbf{b},$$

where $\mathbf{b} = (b_1, \ldots, b_p)^{\mathrm{T}}$, and define a hyperplane by $\{\mathrm{x} : f(\mathrm{x}) = 0\}$. A classification rule induced by $f(\mathrm{x})$ is

$$\mathrm{sign}\{f(\mathrm{x})\} \quad \text{or} \quad \mathrm{sign}(b_0 + \mathrm{x}^{\mathrm{T}}\mathbf{b}).$$

The SVM determines the classification rule by finding an optimal separating hyperplane. There are two cases, called separable and non-separable, respectively, that need to be addressed for the estimation of b_0 and $\mathbf{b}$.

First, consider the separable case. Naively, the idea of maximizing the margin can be captured in the optimization problem

$$\max_{b_0, \mathbf{b}} \quad C \tag{2.3}$$

$$\text{s.t.} \quad y_i^*(b_0 + \mathrm{x}_i^{\mathrm{T}}\mathbf{b}) \geqslant C, \quad i = 1, \ldots, n. \tag{2.4}$$

However, there is no finite solution for (2.3). Without any constraint, any $af(\mathrm{x})$, where $a > 0$, will be a classification rule and a will push C in (2.3) toward infinity.

One way to fix this problem is to add a normalizing constraint, $\|\mathbf{b}\|_2 = 1$ where $\|\mathbf{b}\|_2 = \{\sum_{j=1}^{p} |b_j|^2\}^{1/2}$, and to consider

$$\max_{b_0, \mathbf{b}} \quad C \tag{2.5}$$

$$\text{s.t.} \quad \begin{cases} \|\mathbf{b}\|_2 = 1, \\ y_i^*(b_0 + \mathrm{x}_i^{\mathrm{T}}\mathbf{b}) \geqslant C, \quad i = 1, \ldots, n. \end{cases} \tag{2.6}$$

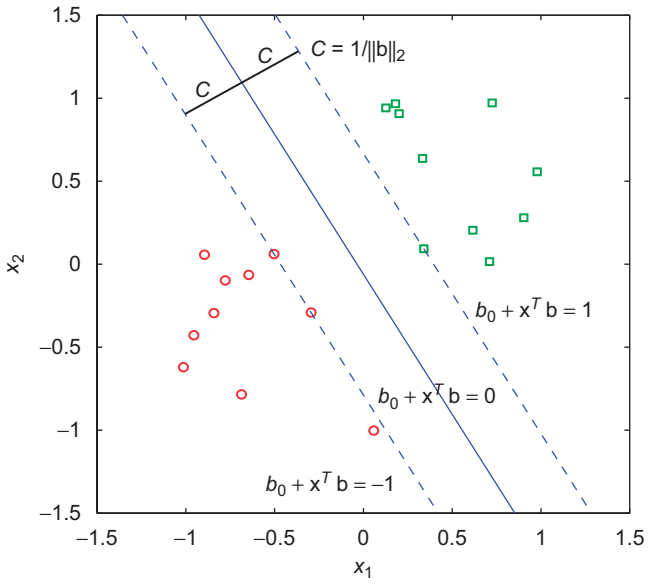

Fig. 1. Illustration of support vector machine for binary classification in the separable case.

More conveniently, the constraints in (2.6) can be substituted by $(1/\|\mathbf{b}\|_2)y_i^*(b_0 + \mathrm{x}_i^{\mathrm{T}}\mathbf{b}) \geqslant C$, $i = 1, \ldots, n$. By setting $\|\mathbf{b}\|_2 = 1/C$, (2.5)–(2.6) are equivalent to

$$\min_{b_0,\,\mathbf{b}} \quad \tfrac{1}{2}\|\mathbf{b}\|_2^2 \tag{2.7}$$

$$\text{s.t.} \quad y_i^*(b_0 + \mathrm{x}_i^{\mathrm{T}}\mathbf{b}) \geqslant 1, \quad i = 1, \ldots, n. \tag{2.8}$$

An illustration of the SVM for classifying a data set with two-dimensional predictors is displayed in Fig. 1.

Second, consider the non-separable case. It is noticed that if the data are not separable, there is no feasible solution for (2.5) since it is not possible to find a positive margin C to separate the data. An alternative is to introduce the slack variables $\{\xi_i\}_{i=1}^n$ to (2.7)–(2.8) to penalize the violation of the constraints, and the problem could be formulated as follows:

$$\min_{b_0,\mathbf{b},\{\xi_i\}} \quad \tfrac{1}{2}\|\mathbf{b}\|_2^2 + \frac{1}{\lambda}\sum_{i=1}^{n}\xi_i \tag{2.9}$$

$$\text{s.t.} \quad \begin{cases} y_i^*(b_0 + \mathrm{x}_i^{\mathrm{T}}\mathbf{b}) \geqslant 1 - \xi_i, & i = 1, \ldots, n, \\ \xi_i \geqslant 0, & i = 1, \ldots, n, \end{cases} \tag{2.10}$$

where $\lambda > 0$ is a tuning parameter. Using the dual theorem (see e.g. Fletcher, 1987, p. 219; Burges, 1998, p. 130; Hastie et al., 2001, p. 374), the minimization problem (2.9)–(2.10) is equivalent to maximizing the dual problem, i.e.,

$$\max_{\boldsymbol{\alpha}} \quad -\tfrac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{H}_{\mathrm{L}}\boldsymbol{\alpha} + \mathbf{1}^{\mathrm{T}}\boldsymbol{\alpha} \tag{2.11}$$

$$\text{s.t.} \begin{cases} \mathbf{0} \leqslant \boldsymbol{\alpha} \leqslant (1/\lambda)\mathbf{1}, \\ \mathbf{y}^{*\mathrm{T}}\boldsymbol{\alpha} = 0, \end{cases} \tag{2.12}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^{\mathrm{T}}$, $\mathbf{1} = (1, \ldots, 1)^{\mathrm{T}}$, $\mathbf{y}^* = (y_1^*, \ldots, y_n^*)^{\mathrm{T}}$, $\mathbf{X} = (\mathrm{x}_1, \ldots, \mathrm{x}_n)^{\mathrm{T}}$ and $\mathbf{H}_{\mathrm{L}} = \mathrm{diag}\,(\mathbf{y}^*)\mathbf{X}\mathbf{X}^{\mathrm{T}}\,\mathrm{diag}(\mathbf{y}^*)$. Denote by $\widehat{\boldsymbol{\alpha}} = (\widehat{\alpha}_1, \ldots, \widehat{\alpha}_n)^{\mathrm{T}}$ the resulting minimizer. The minimizers $\widehat{b}_0$ and $\widehat{\mathbf{b}}$ for (2.9)–(2.10) can be obtained by

$$\widehat{\mathbf{b}} = \sum_{i=1}^{n}\widehat{\alpha}_i y_i^* \mathrm{x}_i,$$

$$\widehat{b}_0 = -\langle \mathrm{x}_t + \mathrm{x}_s, \widehat{\mathbf{b}}\rangle/2,$$

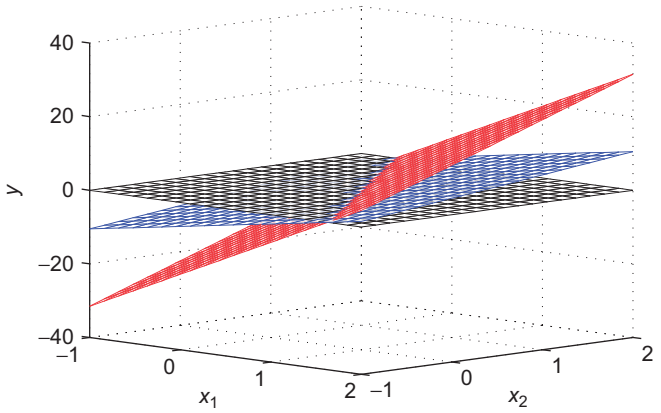

Fig. 2. Plots of $\theta(\mathrm{x}) = -3.5 + 4x_1 + 3x_2$ in TLR, $f(\mathrm{x}) = -10.5 + 12x_1 + 9x_2$ in SVM, and the zero-plane $y = 0$.

where $\langle \cdot, \cdot \rangle$ denotes the inner product and x_t and x_s could be any support points such that

$$0 < \widehat{\alpha}_t < 1/\lambda, \quad y_t^* = 1, \quad 0 < \widehat{\alpha}_s < 1/\lambda, \quad y_s^* = -1.$$

Detailed derivations of the above transformations can be found in Hastie et al. (2001), among others. Observe that (2.11) is a quadratic programming problem with linear constraints, and thus can be solved by standard software. Furthermore, the constraints in (2.12) could handle both separable and non-separable cases, in which the separable case corresponds to $\lambda = 0$, namely, the elements of $\boldsymbol{\alpha}$ have no upper bounds. Fuller details on SVM can be found in Vapnik (1996).

One of the major advantages of SVM is its ability to classify high-dimensional data. By maximizing the dual function in (2.11), only n maximizers $\{\widehat{\alpha}_i\}_{i=1}^n$ need to be solved. After that, the solutions $\widehat{b}_0$ and $\widehat{\mathbf{b}}$ for SVM would be the linear combinations of these n solutions. In addition, SVM is known for its good performance in binary classification. Nonetheless, it is noticed that SVM does not directly estimate $m(\mathrm{x})$, despite that $m(\mathrm{x})$ itself is of substantial interest in a wide array of scientific disciplines. A naive way of estimating $m(\mathrm{x})$ via SVM is

$$\frac{1}{1 + \exp\{-(\widehat{b}_0 + \mathrm{x}^{\mathrm{T}}\widehat{\mathbf{b}})\}}, \tag{2.13}$$

which suffers from severe loss of accuracy. See simulation evaluations in Section 5.

2.3. Connection between TLR and SVM

At first sight, the difference between SVM and TLR may arise from the functions $f(\mathrm{x})$ used in SVM and $\theta(\mathrm{x})$ used in TLR. Fig. 2 illustrates the situation where

$$f(\mathrm{x}) = -10.5 + 12x_1 + 9x_2 \quad \text{and} \quad \theta(\mathrm{x}) = -3.5 + 4x_1 + 3x_2.$$

In that case, $f(\mathrm{x})$ and $\theta(\mathrm{x})$ are apparently distinct since $\theta(\mathrm{x}) = f(\mathrm{x})/3$. However, since $\{\mathrm{x} : f(\mathrm{x}) = 0\} = \{\mathrm{x} : f(\mathrm{x})/3 = 0\}$, the hyperplane, $\{\mathrm{x} : f(\mathrm{x}) = 0\}$, formed by SVM and the hyperplane, $\{\mathrm{x} : \theta(\mathrm{x}) = 0\}$, formed by TLR are exactly the same, thus the two approaches yield identical classification outputs. This example indicates that the difference between f and θ is not the essential cause for the difference between SVM and TLR in classification.

To gain further insight into the difference, we examine the loss functions of SVM and TLR. Define the hinge loss by

$$\mathscr{L}_{\mathrm{H}}(z) = (1 - z)_+, \quad z \in \mathbb{R}, \tag{2.14}$$

where $x_+ = \max\{x, 0\}$. In view of SVM, it can be shown that (2.9)–(2.10) is equivalent to the unconstrained optimization problem

$$(\widehat{b}_0, \widehat{\mathbf{b}}) = \arg\min_{b_0, \mathbf{b}} \left[\sum_{i=1}^{n} \mathscr{L}_{\mathrm{H}}\{y_i^*(b_0 + \mathrm{x}_i^{\mathrm{T}}\mathbf{b})\} + \frac{\lambda}{2}\|\mathbf{b}\|_2^2 \right], \tag{2.15}$$

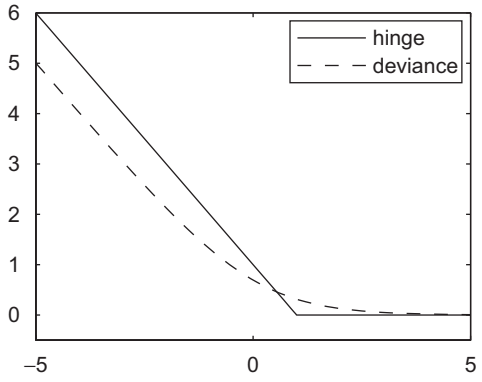

Fig. 3. Comparison of the deviance loss for TLR and the hinge loss for SVM.

in which the criterion function consists of a loss term plus a penalty term. An overview can be found in Burges (1998), Evgeniou et al. (2000), Wahba (1999) and Hastie et al. (2001).

For TLR, recall that the responses are $y_i \in \{0, 1\}$ instead of $y_i^* \in \{-1, 1\}$ in the SVM context. By using the transformation,

$$y_i^* = 2y_i - 1, \quad i = 1, \ldots, n,$$

it is readily seen from (2.2) that TLR estimates solve the unconstrained optimization problem

$$(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}) = \arg \min_{\beta_0, \boldsymbol{\beta}} \left[\sum_{i=1}^{n} \mathscr{L}_{\mathrm{D}}\{y_i^*(\beta_0 + \mathrm{x}_i^{\mathrm{T}}\boldsymbol{\beta})\} + \frac{\gamma}{2}\|\boldsymbol{\beta}\|_2^2 \right], \quad \gamma = 0, \tag{2.16}$$

which is formed by the deviance loss,

$$\mathscr{L}_{\mathrm{D}}(z) = \ln\{1 + \exp(-z)\}, \quad z \in \mathbb{R}, \tag{2.17}$$

and a tuning parameter $\gamma = 0$. The relationship between the hinge loss and deviance loss can clearly be seen from Fig. 3. Indeed, the idea of our proposed PsLR is motivated by the difference arising from the loss functions of SVM and TLR.

3. Pseudo-logistic regression

In this section, we propose a new method called PsLR. The main idea is to replace the deviance loss in the criterion function (2.16) for TLR by a new loss function, called the "pseudo-quadratic loss". This loss function is constructed in a way that intends to better resemble the deviance loss and meanwhile, preserve the piecewise nature of the hinge loss. Similar to TLR, PsLR will give regression estimates $\widehat{\beta}_0$, $\widehat{\boldsymbol{\beta}}$ and $\widehat{\theta}(\mathrm{x}) = \widehat{\beta}_0 + \mathrm{x}^{\mathrm{T}}\widehat{\boldsymbol{\beta}}$. We expect that $\widehat{\theta}(\mathrm{x})$ will mimic the estimate for $\theta(\mathrm{x})$ and can therefore estimate the probabilities of class membership by

$$\widehat{m}(\mathrm{x}) = \frac{1}{1 + \exp\{-\widehat{\theta}(\mathrm{x})\}} \tag{3.1}$$

and $1 - \widehat{m}(\mathrm{x})$. For the purpose of classification, the PsLR classifier is given by $\mathrm{sign}\{\widehat{\theta}(\mathrm{x})\}$ or equivalently, $\mathrm{sign}\{\widehat{m}(\mathrm{x}) - 1/2\}$.

3.1. Pseudo-quadratic logistic regression

We now describe how to approximate the deviance loss function by a piecewise quadratic function whose first-order derivative is continuous. By using the Taylor's expansion, we expand the deviance loss function (2.17) at the point 0 and get the approximation

$$\ln\{1 + \exp(-z)\} \doteq \ln(2) - z/2 + z^2/8.$$

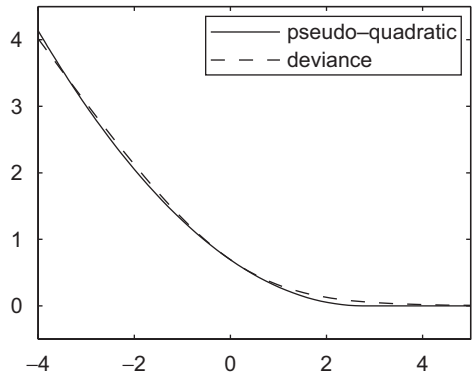

Fig. 4. Comparison of the deviance loss for TLR and the pseudo-quadratic loss for PsLR.

We replace the third term on the right side by $z^2/\{16\ln(2)\}$, so that the resulting function can smoothly touch the zero line. The approximate function leads to the pseudo-quadratic loss function

$$\mathscr{L}_{\mathrm{Q}}(z) = \{(d_1 - d_2 z)_+\}^2, \quad z \in \mathbb{R}, \tag{3.2}$$

where $d_1 = \sqrt{\ln(2)} = .8326$ and $d_2 = 1/\{4\sqrt{\ln(2)}\} = .3003$. (Unless otherwise specified, these canonical values of (d_1, d_2) will be used in the paper.) As illustrated in Fig. 4, this loss function, on the one hand, resembles the deviance loss of TLR very well and, on the other hand, mimics the margin-based hinge loss of SVM. Indeed, the PsLR method will share advantages of the margin-based loss functions for classification.

We propose the PsLR estimates to minimize the criterion function

$$(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}) = \arg\min_{\beta_0, \boldsymbol{\beta}} \left[\sum_{i=1}^{n} \mathscr{L}_{\mathrm{Q}}\{y_i^*(\beta_0 + \mathrm{x}_i^{\mathrm{T}}\boldsymbol{\beta})\} + \frac{\gamma_2}{2}\|\boldsymbol{\beta}\|_2^2 \right], \tag{3.3}$$

where γ_2 is a positive tuning parameter.

The algorithm for obtaining minimizers in (3.3) is described as follows. We first notice that (3.3) is equivalent to the following problem:

$$\min_{\beta_0, \boldsymbol{\beta}, \{\xi_i\}} \quad \tfrac{1}{2}\|\boldsymbol{\beta}\|_2^2 + \frac{1}{\gamma_2}\sum_{i=1}^{n}\xi_i^2 \tag{3.4}$$

$$\text{s.t.} \quad \begin{cases} d_2 y_i^*(\beta_0 + \mathrm{x}_i^{\mathrm{T}}\boldsymbol{\beta}) \geqslant d_1 - \xi_i, & i = 1, \ldots, n, \\ \xi_i \geqslant 0, & i = 1, \ldots, n. \end{cases} \tag{3.5}$$

See the proof in Appendix A. In this way, we can transform the piecewise quadratic function in (3.3) to two sets of inequality constraints; handling the two constraints is comparatively easier than handling the piecewise function itself. By further studying the constraints, we find that the second set of constraints, $\{\xi_i \geqslant 0, i = 1, \ldots, n\}$, is actually not needed. See the proof in Appendix B. Hence we further simplify the problem (3.4)–(3.5) to the following:

$$\min_{\beta_0, \boldsymbol{\beta}, \{\xi_i\}} \quad \tfrac{1}{2}\|\boldsymbol{\beta}\|_2^2 + \frac{1}{\gamma_2}\sum_{i=1}^{n}\xi_i^2 \tag{3.6}$$

$$\text{s.t.} \quad d_2 y_i^*(\beta_0 + \mathrm{x}_i^{\mathrm{T}}\boldsymbol{\beta}) \geqslant d_1 - \xi_i, \quad i = 1, \ldots, n. \tag{3.7}$$

By derivations in Appendix C, the above problem is equivalent to the following dual problem:

$$\max_{\boldsymbol{\alpha}} \quad -\tfrac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{H}_{\mathrm{Q}}\boldsymbol{\alpha} + d_1\mathbf{1}^{\mathrm{T}}\boldsymbol{\alpha} \tag{3.8}$$

$$\text{s.t.} \quad \begin{cases} \mathbf{0} \leqslant \boldsymbol{\alpha}, \\ \mathbf{y}^{*\mathrm{T}}\boldsymbol{\alpha} = 0, \end{cases} \tag{3.9}$$

Table 1
Comparison of SVM and PsLR methods

SVM	PsLR
$(\widehat{b}_0, \widehat{\mathbf{b}}) = \arg\min_{b_0,\mathbf{b}} \left[\sum_{i=1}^{n} \mathscr{L}_{\mathrm{H}}\{y_i^*(b_0 + \mathrm{x}_i^{\mathrm{T}}\mathbf{b})\} + \frac{\lambda}{2}\|\mathbf{b}\|_2^2\right]$	$(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}) = \arg\min_{\beta_0,\boldsymbol{\beta}} \left[\sum_{i=1}^{n} \mathscr{L}_{\mathrm{Q}}\{y_i^*(\beta_0 + \mathrm{x}_i^{\mathrm{T}}\boldsymbol{\beta})\} + \frac{\gamma_2}{2}\|\boldsymbol{\beta}\|_2^2\right]$
$\mathrm{sign}(\widehat{b}_0 + \mathrm{x}^{\mathrm{T}}\widehat{\mathbf{b}})$	$\mathrm{sign}(\widehat{\beta}_0 + \mathrm{x}^{\mathrm{T}}\widehat{\boldsymbol{\beta}})$
$\min_{b_0,\mathbf{b},\{\xi_i\}} \left[\frac{1}{2}\|\mathbf{b}\|_2^2 + \frac{1}{\lambda}\sum_{i=1}^{n}\xi_i\right]$, s.t. $\begin{cases} y_i^*(b_0 + \mathrm{x}_i^{\mathrm{T}}\mathbf{b}) \geqslant 1 - \xi_i, & i = 1,\ldots,n, \\ \xi_i \geqslant 0, & i = 1,\ldots,n. \end{cases}$	$\min_{\beta_0,\boldsymbol{\beta},\{\xi_i\}} \left[\frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + \frac{1}{\gamma_2}\sum_{i=1}^{n}\xi_i^2\right]$, s.t. $\{d_2 y_i^*(\beta_0 + \mathrm{x}_i^{\mathrm{T}}\boldsymbol{\beta}) \geqslant d_1 - \xi_i, \quad i = 1,\ldots,n.$
$\max_{\boldsymbol{\alpha}}\{-\frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{H}_{\mathrm{L}}\boldsymbol{\alpha} + \mathbf{1}^{\mathrm{T}}\boldsymbol{\alpha}\}$, s.t. $\begin{cases} \mathbf{0} \leqslant \boldsymbol{\alpha} \leqslant (1/\lambda)\mathbf{1}, \\ \mathbf{y}^{*\mathrm{T}}\boldsymbol{\alpha} = 0. \end{cases}$	$\max_{\boldsymbol{\alpha}}\{-\frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{H}_{\mathrm{Q}}\boldsymbol{\alpha} + d_1\mathbf{1}^{\mathrm{T}}\boldsymbol{\alpha}\}$, s.t. $\begin{cases} \mathbf{0} \leqslant \boldsymbol{\alpha}, \\ \mathbf{y}^{*\mathrm{T}}\boldsymbol{\alpha} = 0. \end{cases}$
$\mathbf{H}_{\mathrm{L}} = \mathrm{diag}(\mathbf{y}^*)\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathrm{diag}(\mathbf{y}^*)$	$\mathbf{H}_{\mathrm{Q}} = d_2^2\,\mathrm{diag}(\mathbf{y}^*)\mathbf{X}\mathbf{X}^{\mathrm{T}}\,\mathrm{diag}(\mathbf{y}^*) + (\gamma_2/2)\mathbf{I}$

where $\mathbf{H}_{\mathrm{Q}} = d_2^2\,\mathrm{diag}(\mathbf{y}^*)\mathbf{X}\mathbf{X}^{\mathrm{T}}\,\mathrm{diag}(\mathbf{y}^*) + (\gamma_2/2)\mathbf{I}$, with $\mathbf{I}$ an identity matrix. After obtaining $\widehat{\boldsymbol{\alpha}}$, the solutions $\widehat{\boldsymbol{\beta}}$ and $\widehat{\beta}_0$ in (3.3) can be calculated as follows:

$$\widehat{\boldsymbol{\beta}} = d_2 \sum_{i=1}^{n} \widehat{\alpha}_i y_i^* \mathrm{x}_i, \tag{3.10}$$

$$\widehat{\beta}_0 = y_j^*(d_1 - \widehat{\alpha}_j\gamma_2/2)/d_2 - \langle \mathrm{x}_j, \widehat{\boldsymbol{\beta}}\rangle, \tag{3.11}$$

where the index j is chosen so that $\widehat{\alpha}_j > 0$. By utilizing dual representation, the algorithm for (3.3) is significantly simplified. More importantly, for high-dimensional data, we only need to optimize an objective function with respect to an $n \times 1$ vector $\boldsymbol{\alpha}$, other than a $p \times 1$ vector $\boldsymbol{\beta}$. Therefore, this algorithm enables the PsLR method to efficiently handle large p small n data.

In the above formulation, there is a tuning parameter γ_2. It can be selected by cross-validation. Recall that in TLR, the corresponding tuning parameter in the criterion function (2.16) equals zero. This observation gives us a hint that for PsLR, a small value of γ_2 may suffice. From the simulation studies, for ordinary sample sizes, we find that the classification performance can hardly be influenced by the choice of this parameter, since we have already set a "constraint" that the fit for $\theta(\mathrm{x})$ be linear. The tuning parameter will play a more significant role if we extend the space for $\theta(\mathrm{x})$ to a much wider space like a kernel space and employ this tuning parameter to control the modelling complexity. The advantage of allowing a non-zero γ_2 in the penalty term here is to prevent the so-called monotone likelihood problem (Firth, 1993). It is well known that, when the binary data are separable, the likelihood function for TLR will be a monotone function and the fit of the parameters will diverge to infinity. By adding this penalized term, we can avoid such a problem and still produce a solution to PsLR.

3.2. Comparison between PsLR and other algorithms

We first compare PsLR with SVM. In SVM, a Hessian matrix, $\mathbf{H}_{\mathrm{L}}$, appears in its dual problem (2.11). When $n > p$, $\mathbf{H}_{\mathrm{L}}$ is semi-positive definite, i.e., a matrix not having full rank. As pointed out by Burges and Crisp (2000), the solutions of the corresponding quadratic programming algorithms are not unique. A numerical way to ameliorate this problem is to substitute $\mathbf{H}_{\mathrm{L}}$ by $\mathbf{H}_{\mathrm{L}} + 10^{-10}\mathbf{I}$. Alternatively, a decomposition technique can be used (see Osuna et al., 1997 for details).

Interestingly, for PsLR, the associated Hessian matrix, $\mathbf{H}_{\mathrm{Q}}$ in (3.8), is always positive definite, and thus a unique solution will automatically be ensured. From our simulation, when n is large and p is small, the PsLR solution converges faster and is computationally stabler than that of SVM. For the sake of clarity, Table 1 compares the major features of

Table 2
Comparison of TLR, PsLR and PeLR methods

Criterion (2.16)	TLR	PsLR	PeLR
Loss	$\mathscr{L}_{\mathrm{D}}$	$\mathscr{L}_{\mathrm{Q}}$	$\mathscr{L}_{\mathrm{D}}$
Tuning parameter	0	>0	>0
Solution when $n \ll p$	N.A.	Exact; using dual form	Approximate, non-unique; using transformation

SVM and PsLR in terms of the corresponding un-constrained optimization problem, classifier, constrained optimization problem, dual problem, and Hessian matrix.

Compared with the TLR method, the proposed PsLR method can very easily be applied to both regression and classification for high-dimensional data. In addition, the new method is computationally efficient and stable. When the data are separable, TLR will suffer from the monotone likelihood problem. In contrast, our method overcomes this problem.

We notice that PeLR for high-dimensional classification was considered by Eilers et al. (2001), but with quite different motivations. PeLR uses the criterion function identical to (2.16) of TLR and sets $\gamma > 0$, whereas TLR sets $\gamma = 0$. The solutions to the nonlinear penalized likelihood equations of PeLR were obtained by first approximating $m(\mathrm{x})$ by its first-order Taylor expansion, then forming systems of linear estimating equations, and finally developing transformation methods to reduce the size of the systems. Our simulation experiments indicate that for $n \gg p$, the PeLR and PsLR deliver very similar regression estimates, whereas for $n \ll p$, the PsLR method is computationally more stable than the PeLR method. This is mainly due to the fact that when $n \ll p$, the size-reduced coefficient matrix in PeLR is semi-positive definite (but not positive definite), thus no unique solution is ensured. For a fair comparison, numerical results in Sections 5–6 using PeLR will be reported only for the case $n \gg p$. As a summary, Table 2 compares TLR, PsLR and PeLR methods.

3.3. Why scaling in PsLR? Difference between regression and classification

We make some remarks on the choices of d_1 and d_2 in the pseudo-quadratic loss. First, it is true that $\mathscr{L}_{\mathrm{Q}}(z)$ is proportional to $\{(1-d_2/d_1 z)_+\}^2$. Second, by reparametrizations, the solutions in (3.3) can be rewritten as $\widehat{\beta}_0 = d_1/d_2\widehat{\beta}_0^*$ and $\widehat{\boldsymbol{\beta}} = d_1/d_2\widehat{\boldsymbol{\beta}}^*$, where

$$(\widehat{\beta}_0^*, \widehat{\boldsymbol{\beta}}^*) = \arg\min_{\beta_0^*, \boldsymbol{\beta}^*} \left[\sum_{i=1}^{n} [\{1 - y_i^*(\beta_0^* + \mathrm{x}_i^{\mathrm{T}}\boldsymbol{\beta}^*)\}_+]^2 + \frac{\gamma_2 d_1}{2d_2^2} \|\boldsymbol{\beta}^*\|_2^2 \right].$$

It should be emphasized that different values of the scaling multiplier, d_1/d_2, do not affect much the classification outcomes, since from the "*classification perspective*",

$$\mathrm{sign}(\widehat{\beta}_0 + \mathrm{x}^{\mathrm{T}}\widehat{\boldsymbol{\beta}}) = \mathrm{sign}(\widehat{\beta}_0^* + \mathrm{x}^{\mathrm{T}}\widehat{\boldsymbol{\beta}}^*).$$

Nonetheless, different values of d_1/d_2 indeed cause an essential difference in regression estimates of the class membership probabilities, since from the "*regression viewpoint*",

$$\frac{1}{1+\exp\{-(\widehat{\beta}_0 + \mathrm{x}^{\mathrm{T}}\widehat{\boldsymbol{\beta}})\}} \neq \frac{1}{1+\exp\{-(\widehat{\beta}_0^* + \mathrm{x}^{\mathrm{T}}\widehat{\boldsymbol{\beta}}^*)\}}.$$

See Theorem 1 for some additional insight and Section 5 for numerical evidence.

Third, because the main focus of our paper is to develop the PsLR approach which "simultaneously" carries out regression and classification for high-dimensional data, special cares need to be taken for d_1 and d_2. The method we propose in Section 3.1 for determining canonical values of (d_1, d_2) has a more natural interpretation: they are chosen to better approximate the deviance loss, namely, to enhance accuracy of regression estimates, a goal that is absent from the SVM classifier.

4. Bias correction for PsLR estimates

In this section, we study a bias correction procedure for PsLR in finite-sample situations to achieve more efficient estimates of the class membership probabilities $m(\mathrm{x})$ and $1 - m(\mathrm{x})$.

We first present a nice property of the pseudo-quadratic loss function. See Appendix D for the proof. The result will be helpful for finding the bias expression more explicitly.

Theorem 1. *Assume that conditional on* $\mathrm{X} = \mathrm{x}$, $Y \sim \text{Bernoulli}(m(\mathrm{x}))$. *Define* $Y^* = 2Y - 1$. *For constants* $d_1 > 0$ *and* $d_2 > 0$ *in* (3.2) *for the pseudo-quadratic loss function* $\mathscr{L}_{\mathrm{Q}}$, *the minimizer of* $E[\mathscr{L}_{\mathrm{Q}}\{Y^* g(\mathrm{X})\}]$ *with respect to a measurable function* g *such that* $E\{g^2(\mathrm{X})\} < \infty$ *is given by*

$$g_{\mathrm{B}}(\mathrm{x}) = d_1/d_2\{2m(\mathrm{x}) - 1\}$$

and thus g_{B} *is a Bayes optimal rule*.

Remark 1. In Theorem 1, if the pseudo-quadratic loss $\mathscr{L}_{\mathrm{Q}}$ of PsLR is replaced by the hinge loss $\mathscr{L}_{\mathrm{H}}$ of SVM, then similar arguments will show that the corresponding minimizer is $\operatorname{sign}\{2m(\mathrm{x}) - 1\}$.

We now investigate the bias of $\widehat{m}(\mathrm{x})$ arising from the PsLR estimation. Let $\widehat{\theta}(\mathrm{x}) = \widehat{\beta}_0 + \mathrm{x}^{\mathrm{T}}\widehat{\boldsymbol{\beta}}$, where $\widehat{\beta}_0$ and $\widehat{\boldsymbol{\beta}}$ are the minimizers of the criterion function (3.3), which is equivalent to

$$\frac{1}{n}\sum_{i=1}^{n} \mathscr{L}_{\mathrm{Q}}\{y_i^*(\beta_0 + \mathrm{x}_i^{\mathrm{T}}\boldsymbol{\beta})\} + \frac{\gamma_2}{2n}\|\boldsymbol{\beta}\|_2^2.$$

The first term above is asymptotically consistent to $E[\mathscr{L}_{\mathrm{Q}}\{Y^*(\beta_0 + \mathrm{X}^{\mathrm{T}}\boldsymbol{\beta})\}]$ as the sample size tends to infinity, whereas the second term in our implementation has a negligible effect as compared to the first term. From Theorem 1, we anticipate that under the assumption (2.1),

- $\widehat{\theta}(\mathrm{x})$ will mimic the estimate for $g_{\mathrm{B}}(\mathrm{x})$, and consequently,
- $\widehat{m}(\mathrm{x})$, as defined in (3.1), will mimic the estimate for $1/[1 + \exp\{-g_{\mathrm{B}}(\mathrm{x})\}]$.

This leads to the approximate asymptotic bias of $\widehat{m}(\mathrm{x})$ in the form

$$\operatorname{Abias}\{m(\mathrm{x})\} = \frac{1}{1 + \exp[-d_1/d_2\{2m(\mathrm{x}) - 1\}]} - m(\mathrm{x}),$$

which is plotted in Fig. 5. The magnitude of the approximate asymptotic bias is conceivably small.

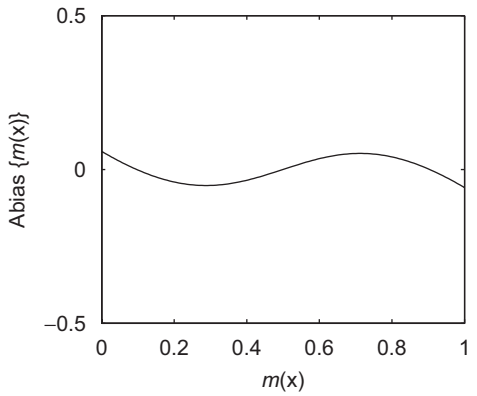

Fig. 5. The approximate asymptotic bias of $\widehat{m}(\mathrm{x})$ in PsLR versus $m(\mathrm{x})$.

Table 3
Regression estimates of $(\beta_0, \beta_1, \beta_2)$ for the simulated data

EX	True	PsLR $d_1 = .8326$ $d_2 = .3003$	TLR	SVM	PsLR $d_1 = 1$ $d_2 = 1$	PeLR
I	−3.5	−3.5520 (.5105)	−3.6103 (.7070)	−2.9649 (.4774)	−1.3555 (.2069)	−3.3332 (.6012)
	4.0	4.1099 (.6862)	4.1215 (.8538)	3.5085 (.6022)	1.5654 (.2707)	3.8145 (.7407)
	3.0	3.0086 (.7597)	3.1139 (.8639)	2.4312 (.7515)	1.1510 (.2972)	2.8653 (.7619)
II	−3.5	−3.1293 (.5779)	−3.7753 (.8712)	−2.6066 (.6333)	−1.2124 (.2647)	−3.4268 (.6328)
	4.0	3.5434 (.5676)	4.2817 (.9125)	2.9404 (.6535)	1.3731 (.2712)	3.8850 (.6319)
	3.0	2.6625 (.4545)	3.2126 (.7025)	2.2136 (.5000)	1.0311 (.2104)	2.9169 (.5059)

Motivated from Theorem 1, we propose the finite-sample bias corrected estimate

$$\widehat{m}_c(\mathrm{x}) = \begin{cases} 0 & \text{if } \widehat{\theta}(\mathrm{x}) < -d_1/d_2, \\ \widehat{m}(\mathrm{x}) - \mathrm{Abias}\{\widehat{m}(\mathrm{x})\} & \text{if } |\widehat{\theta}(\mathrm{x})| \leqslant d_1/d_2, \\ 1 & \text{if } \widehat{\theta}(\mathrm{x}) > d_1/d_2. \end{cases}$$

The proposed bias correction method will be illustrated with the Leukemia data in Section 6.2.1.

5. Simulation study

To evaluate the performance of PsLR (using canonical values $d_1 = .8326$ and $d_2 = .3003$) in estimating the regression coefficients, we conduct the simulation study.

We consider the Bernoulli response variable Y which, conditional on $\mathrm{X} = (X_1, X_2)^{\mathrm{T}}$, has the canonical parameter given in (2.1), where the true values of the parameters are

$$\beta_0 = -3.5, \quad \beta_1 = 4, \quad \beta_2 = 3.$$

Two types of design variables are considered:

Example I: $X_1 \sim U(0, 1)$, $X_2 \sim U(0, 1)$, X_1 is independent of X_2;
Example II: $X_1 \sim N(.5, 1)$, $X_2 \sim N(.5, 1)$, X_1 is independent of X_2.

To facilitate comparison, the SVM estimates, the PsLR estimates using $d_1 = 1$ and $d_2 = 1$, and the PeLR estimates are included. For PsLR, as we have explained before, γ_2 in the criterion function (3.3) can be selected by cross-validation, but for a linear form of $\theta(\mathrm{x})$, the choice of γ_2 will be less important. Thus, we adopt $\gamma_2 = .1$ throughout simulation studies in the paper. For SVM, as in Section 6.1.2, $\lambda = .01$ is used throughout. For PeLR, the tuning parameter is the same as γ_2.

We generate 200 random samples $\{(\mathrm{x}_i, y_i)_{i=1}^n\}$ of size $n = 150$ from the distribution of (X, Y). The estimates of $(\beta_0, \beta_1, \beta_2)$ are summarized in Table 3, where the numbers in the brackets are standard deviations of the estimates from 200 runs. The boxplots of $\widehat{\beta}_k - \beta_k$, $k = 0, 1, 2$, are also displayed in Fig. 6.

The simulation results demonstrate the following features. First, the estimates from PsLR (using canonical values of (d_1, d_2)), TLR and PeLR are comparable, whereas the estimates via SVM and PsLR using alternative scaling of d_1/d_2 are much more biased. This lends support to the suitable choice of the scaling multiplier. Second, the PsLR and PeLR estimates have smaller variances than those of TLR estimates. This is mainly due to the presence of L_2-penalty terms on regression coefficients in PsLR and PeLR, and the absence of such penalization in TLR. Indeed, for high-dimensional data, suitably incorporating the penalty makes the PsLR estimates stabler and algorithm convergence faster than those of TLR.

6. Real data analysis

In this section, we apply PsLR to six sets of real data to assess the regression estimates (in two sets) and the classification performance (in five sets). We adopt the following procedure to evaluate the classification performance

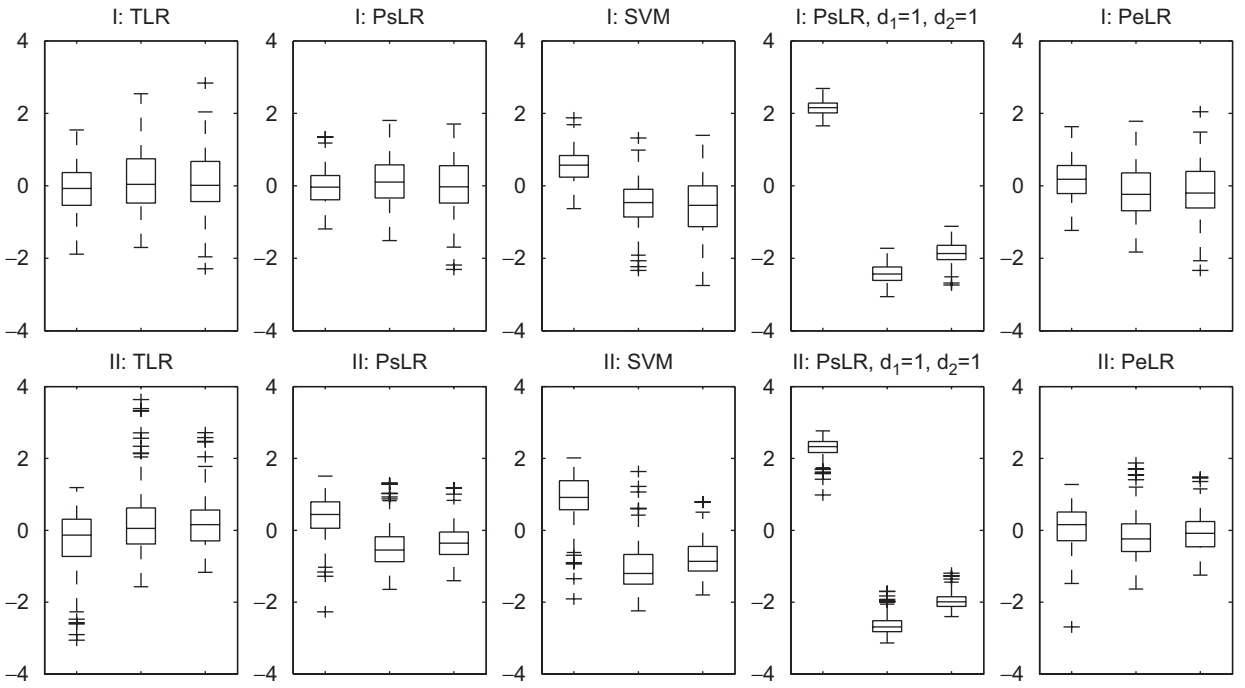

Fig. 6. Boxplots of $\widehat{\beta}_0 - \beta_0$, $\widehat{\beta}_1 - \beta_1$, and $\widehat{\beta}_2 - \beta_2$ (from left to right in each panel) by TLR, PsLR with canonical values $(d_1, d_2) = (.8326, .3003)$, SVM, PsLR with alternative values $(d_1, d_2) = (1, 1)$, and PeLR. The top panel is for the uniform design and the bottom panel is for the normal design.

for data according to their diversity and size.

- (Type 1) If the data set has already been splitted into training and test sets, we simply train the classifier on the training set and evaluate it on the test set. This type of data includes the Leukemia data. For other types, we use the following ways of random splitting.
- (Type 2) If the sample size of the data set is medium or large, i.e., at least 100, we use the 10-fold cross-validation method to evaluate. Lim et al. (2000) used the same procedure to compare 33 data sets. We describe the procedure as follows which are extracted from their paper:
  - The data set is randomly divided into 10 disjoint subsets, each containing approximately the same number of records. Sampling is stratified by the class labels to ensure that subset class proportions are roughly the same as those in the whole data set.
  - For each subset, a classifier is constructed using the records not in it. The classifier is then tested on the withheld subset to obtain a cross-validation estimate of its error rate.
  - The 10 cross-validation estimates of error rate are averaged to provide an estimate for the classifier constructed from all data.

  This type of data includes the Pima Indians Diabetes data and Wisconsin Breast Cancer data.
- (Type 3) If the sample size of the data set is small, i.e., fewer than 100, we randomly split the data and use one part for training and the other part for testing. This type of data includes the Breast Cancer Gene Expression data and Colon data. We replicate this random splitting 100 times and calculate the average misclassification number from these 100 runs. This method is widely used in gene classification (see Ding and Gentleman, 2005) and other classification contexts (see Shen et al., 2003). Shen et al. (2003) chose the ratio 1:1 for splitting the training and test sets. Ding and Gentleman (2005) chose the ratio 2:1 for training and test sets. Since the Breast Cancer data here has fewer samples than they used, we choose the ratio 7:3 to guarantee a reasonable number of samples from each group.

Table 4
Regression estimates for Minnesota Storm data

Variable	PsLR	TLR	PeLR
Intercept	−9.2188	−9.5621 (.7499)	−9.4559
$\log_2(D)$	2.1352	2.2164 (.2079)	2.1989
S	4.2714	4.5086 (.5159)	4.3889

Table 5
Regression estimates for Pima Indians Diabetes data

Variable	PsLR	TLR	PeLR
Intercept	−8.4955	−8.4047 (.7166)	−8.4003
Pregnant	.1248	.1232 (.0321)	.1231
Plasma glucose	.0361	.0352 (.0037)	.0352
Diastolic blood pressure	−.0132	−.0133 (.0052)	−.0133
Triceps skin fold thickness	.0006	.0006 (.0069)	.0006
2-h serum insulin	−.0012	−.0012 (.0009)	−.0012
Body mass index	.0880	.0897 (.0151)	.0897
Diabetes pedigree function	.8944	.9452 (.2991)	.9368
Age	.0151	.0149 (.0093)	.0149

6.1. Small p large n data

6.1.1. Minnesota Storm data

We first consider the data set described and analyzed in Weisberg (2005, p. 251) by TLR. (The website http://www.stat.umn.edu/alr/data.html links the data set, blowBF.txt.) A storm on July 4, 1999 with winds exceeding 90 miles per hour hit the boundary waters canoe area wilderness in northeastern Minnesota, causing serious damage to the forest. Roy Rich studied the effects of this storm using a very extensive ground survey of the area. One goal of this study is to determine the dependence of survival on species, size of the tree and the local severity.

We use PsLR to analyze this data set. We use 659 Balsam Fir trees, with the response variable Y coded as 1 for trees that were blown down and died and 0 for trees that survived, versus the predictor $\log_2(D)$, the base-two logarithm of the diameter of the tree, and a severity variable S. Table 4 compares the PsLR estimates with TLR and PeLR estimates. The numbers in the brackets are standard errors from TLR. From Table 4, we can see that estimates using three methods are comparable. The estimates from PsLR seem to be more shrinking in magnitude.

6.1.2. Pima indians diabetes data

This data file, donated by Vincent Sigillito, Johns Hopkins University, is downloadable from ftp://ftp.ics.uci.edu/pub/machine-learning-databases/pima-indians-diabetes/. The data set has 768 samples and eight covariates. The goal of the study was to establish a relationship between the eight measurements collected and whether or not the woman has diabetes (see Smith et al., 1988). Both regression and classification approaches can be used to analyze this data.

First, we estimate the regression coefficients. Table 5 details the estimates from TLR, PsLR and PeLR. The results lend further support for the closeness between the PsLR, TLR and PeLR estimates.

Second, we compare the classification performance of our method with SVM and PeLR. Since the sample size of this data set is medium, the 10-fold cross-validation is used to evaluate the performance. We randomly divide 768 samples into 10 disjoint sets, in which nine sets have 77 samples and one set has 75 samples. It is noticed that directly implementing the SVM algorithm will get a non-full rank Hessian matrix. To fix this problem, we download the Matlab code for SVM from http://www.isis.ecs.soton.ac.uk/resources/svminfo/. The version is SVM7-22 provided by Steve Gunn. Throughout the paper, we use the tuning parameter $\lambda = .01$ for SVM, which is selected from $[10^{-3}, 10]$ to achieve the best result for SVM. The results are summarized in Table 6. For this data set, the PsLR and PeLR methods provide more accurate classification results than SVM. Interestingly, Lim et al. (2000, p. 215) analyzed the same data set using the same 10-fold cross-validation to evaluate 33 classification methods. They reported that those 33 classification

Table 6
Classification of Pima Indians Diabetes data

	PsLR	SVM	PeLR
Average misclassification rate	.22937	.23063	.22677

Table 7
Classification of Wisconsin Breast Cancer data

	PsLR	SVM	PeLR
Average misclassification rate	.0325	.0369	.0371

Table 8
Classification of Leukemia data

	PsLR	SVM
Experiment 1	1	3
Experiment 2	0	0

methods have error rates ranging from .22 to .31 on this data set. Therefore, the PsLR method can be regarded as one of the most accurate classifiers for this data set based on the 10-fold cross-validation evaluation.

6.1.3. Wisconsin breast cancer data

The Wisconsin Breast Cancer data set, collected at University of Wisconsin Hospitals, concerns visually assessed nuclear features of fine-needle aspirates taken from patients' breasts. There are 699 samples in which 16 samples contain missing values. This leads to 683 samples in our data analysis. For each sample, there are nine diagnostic characteristics with each component being in the interval 1–10 where 1 corresponds to a normal state and 10 corresponds to a most abnormal state. The task is to decide whether a sample is benign or malignant. The data set was described in Wolberg and Mangasarian (1990) and can be downloaded from http://www.ics.uci.edu/∼mlearn/databases/breast-cancer-wisconsin/. Wolberg and Mangasarian's original work applied the multisurface method to a 369-case subset of the database, resulting in testing error rates more than 6%.

We compare the classification performance of PsLR with SVM and PeLR using the 10-fold cross-validation, where nine subsets have 69 samples and one has 62 samples. From the results listed in Table 7, we observe that PsLR method has lower misclassification rates than SVM and PeLR. Lim et al. (2000, p. 215) reported that for this data set, those 33 classification methods have error rates from .03 to .09. Based on their results, our method belongs to the best classifiers for this data set based on the 10-fold cross-validation evaluation.

6.2. Large p small n data

6.2.1. Leukemia data

The data set, downloadable from http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi, is composed of training and test sets. The training set consists of 38 leukemia patients of which 11 suffer from acute myeloid leukemia (AML) and 27 from acute lymphoblastic leukemia (ALL). The test set consists of 34 patients of which 14 suffer from AML and 20 from ALL. The number of gene expression levels is 7129. A primary issue is to separate the AML samples from the ALL samples. Golub et al. (1999) used "neighborhood analysis" method to select the genes and "weighted vote" method to do the classification. They found that predictors based on between 10 and 200 genes were all found to be 100% accurate.

We conduct two experiments for this data set. In the first experiment, we use the training set to evaluate the performance on the test set. The results are shown in Table 8. Particularly, the PsLR method has one misclassified sample, whereas SVM has three misclassified samples. Zhu and Hastie (2004, p. 434) analyzed the same data set by using SVM with

Table 9
Classification of Breast Cancer Gene Expression data

	PsLR	SVM
Average misclassification number	2.19	2.21

Table 10
Classification of Colon data

	PsLR	SVM
Average misclassification number	4.08	4.16

two different gene selection methods. They reported that SVM had one and three misclassified samples by different gene selection methods. Without screening out noisy genes, our method on this data set possesses the same accuracy as those methods using gene screening. The second experiment is to combine the training set and test set for classification. Table 8 reveals that all methods have zero misclassification on the combined data set.

In the two experiments above, we use the entire 7129 genes for classification. We would like to remark here that screening methods are often used before the classification. Properly eliminating some noisy genes can increase the prediction accuracy, but eliminating genes before building a model may overlook some important predictors and the relationship between predictors. Some of the traditional methods need to do the screening first, because those methods either could not directly handle high-dimensional variables or those methods will become computationally intensive. See Dudoit et al. (2002) for details. Using all of the genes will allow us to obtain a "global" view of all genes and help direct further study. Hence an algorithm which can handle all genes simultaneously is desirable.

Moreover, we use the 38 training samples to obtain the PsLR estimate of the probability that the misclassified sample (in Table 8) in the test set belongs to ALL. The estimated probability is .4660. For this data set, neither TLR nor SVM can estimate the probability. The PsLR method seems to be more flexible than both SVM and TLR.

6.2.2. Breast cancer gene expression data

The data set can be retrieved from http://data.cgt.duke.edu/west.php. In this data set, there are 49 samples with 7129 genes, in which 25 samples are with ER+ and 24 samples are with ER−. West et al. (2001) developed Bayesian regression models to analyze this data set. They found five misclassified samples based on the ER status.

We randomly split this data set and use 70% of data (34 out of 49) for the training set and the other 30% of data (15 out of 49) for the test set. We apply SVM and PsLR to the training set and calculate the number of misclassification on the test set. We repeat the random splitting 100 times and calculate the average misclassification number. The results are summarized in Table 9. For this data set, the PsLR method has higher classification accuracy than SVM.

6.2.3. Colon data

The classification of colon cancer is discussed in Alon et al. (1999) and can be downloaded from http://microarray.princeton.edu/oncology/affydata/index.html. In this data set, there are 62 samples and 2000 genes, in which 22 samples are from normal colon tissues and 40 samples are from tumor tissues. Alon et al. (1999) used two-way clustering and were able to cluster 19 normal and five tumor samples into one group and 35 tumor and three normal tissues into the other group.

We compare SVM and PsLR for this data set. We randomly split this data set and use 70% of data (43 out of 62) for the training set and the other 30% of data (19 out of 62) for the test set. We calculate the misclassification number on the test data set. We repeat the random splitting 100 times and calculate the average misclassification number. The results summarized in Table 10 continues to provide evidence that the PsLR classifier outperforms the SVM counterpart.

7. Discussion

There is a diverse and extensive literature addressing classification methods. See Zhang and Fu (2006) for discussion of some issues. In this paper, we aim to develop a new method which could simultaneously perform reliable regression and conduct effective classification for high-dimensional binary data. In contrast, neither SVM alone nor TLR alone could achieve both goals simultaneously. Our research thus will benefit the statistical analysis of high dimension low sample size data. This is particularly attractive to the statistical analysis of microarray data.

A number of extensions could be further made. First, loss functions other than the pseudo-quadratic loss could be used to approximate the deviance loss. Second, the current paper focuses on the parametric linear form of $\theta(\mathrm{x})$; it would be interesting to investigate whether non-parametric forms of $\theta(\mathrm{x})$ could gain more flexibilities over those overly restrictive forms. Third, the L_2-penalty $\sum_{j=1}^{p}\beta_j^2$ in (3.3) for the PsLR method could be replaced by the L_q-penalty, $\sum_{j=1}^{p}|\beta_j|^q$ where $q\geqslant 0$, or other alternatives. Fourth, for high-dimensional data, identifying a small subset of significant input variables is one useful way of achieving dimension reduction. From a formulation viewpoint, variable selection, sample classification, and probability estimation aim at non-overlapping targets. Examining whether and to what extent the estimates of the class label probabilities will be improved, following the variable selection procedure as in L_1 SVM (Bradley and Mangasarian, 1998) and SCAD SVM (Zhang et al., 2006), is a non-trivial but an interesting investigation. We intend to systematically explore these issues in future work.

Acknowledgment

The research is supported in part by National Science Foundation Grant DMS-03-53941 and Wisconsin Alumni Research Foundation. We thank Dr. Jerry Zhu, at Computer Sciences Department, University of Wisconsin-Madison, for helpful comments and discussions. The authors are also grateful to the Guest Editors, Professor Hans-Hermann Bock and Professor Maurizio Vichi, and two anonymous referees for helpful comments and suggestions.

Appendix A. Equivalence between (3.3) and (3.4)–(3.5)

To facilitate the discussion, we will establish the equivalence in the more general set-up: for $k>0$, the optimization problems

$$\text{(I)}: \quad \min_{\beta_0,\boldsymbol{\beta}}\left[\sum_{i=1}^{n}[\max\{1-y_i^*(\beta_0+\mathrm{x}_i^{\mathrm{T}}\boldsymbol{\beta}),0\}]^k+P(\boldsymbol{\beta})\right]$$

and

$$\text{(II)}: \quad \begin{cases}\min\limits_{\beta_0,\boldsymbol{\beta},\{\xi_i\}} & \sum_{i=1}^{n}\xi_i^k+P(\boldsymbol{\beta})\\ \text{s.t.} & \xi_i\geqslant\max\{1-y_i^*(\beta_0+\mathrm{x}_i^{\mathrm{T}}\boldsymbol{\beta}),0\},\quad i=1,\ldots,n\end{cases}$$

are equivalent, where $P(\boldsymbol{\beta})$ is a function of $\boldsymbol{\beta}$. The PsLR corresponds to $k=2$.

To show the equivalence, we first note that (I) can be rewritten as

$$\text{(I)}: \quad \begin{cases}\min\limits_{\beta_0,\boldsymbol{\beta}} & \sum_{i=1}^{n}\xi_i^k+P(\boldsymbol{\beta})\\ \text{s.t.} & \xi_i=\max\{1-y_i^*(\beta_0+\mathrm{x}_i^{\mathrm{T}}\boldsymbol{\beta}),0\},\quad i=1,\ldots,n.\end{cases}$$

Since the two optimization problems have a common objective function and the constraints in (I) are contained in (II), we only need to show that the minimizers of (II) fulfill the constraints in (I).

Suppose that $(\widehat{\beta}_0,\widehat{\boldsymbol{\beta}},\widehat{\xi}_1,\ldots,\widehat{\xi}_n)$ are the minimizers of (II). It follows that $\widehat{\xi}_i\geqslant\max\{1-y_i^*(\widehat{\beta}_0+\mathrm{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}}),0\}, i=1,\ldots,n$. Then we conclude that $(\widehat{\beta}_0,\widehat{\boldsymbol{\beta}})$ must be the minimizers of (I). To verify this, it suffices to show that in (II), the inequalities

must be equalities

$$\widehat{\xi}_i = \max\{1 - y_i^*(\widehat{\beta}_0 + \mathrm{x}_i^{\mathrm{T}}\widehat{\boldsymbol{\beta}}), 0\}$$

for all $i = 1, \ldots, n$. Otherwise, there exist $i_0 \in \{1, \ldots, n\}$ and $C_{i_0} > 0$, such that

$$\widehat{\xi}_{i_0} = \max\{1 - y_{i_0}^*(\widehat{\beta}_0 + \mathrm{x}_{i_0}^{\mathrm{T}}\widehat{\boldsymbol{\beta}}), 0\} + C_{i_0}.$$

Then $\widehat{\xi}_{i_0} - C_{i_0}$ satisfies the constraint in (II) for i_0. Clearly, $(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}, \widehat{\xi}_1, \ldots, \widehat{\xi}_{i_0-1}, \widehat{\xi}_{i_0} - C_{i_0}, \widehat{\xi}_{i_0+1}, \ldots, \widehat{\xi}_n)$ gives a smaller value of the objective function in (II) than $(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}, \widehat{\xi}_1, \ldots, \widehat{\xi}_n)$ does. This is a contradiction to the assumption on $(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}, \widehat{\xi}_1, \ldots, \widehat{\xi}_n)$. The proof is thus completed.

Appendix B. Equivalence between (3.4)–(3.5) and (3.6)–(3.7)

We observe that the two optimization problems have a common objective function, and that the constraints (3.5) are contained in (3.7). It suffices to show that the minimizers of (3.6)–(3.7) satisfy the constraints in (3.5).

Suppose that $(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}, \widehat{\xi}_1, \ldots, \widehat{\xi}_n)$ are the minimizers of (3.6)–(3.7). It follows that $\widehat{\xi}_i \geqslant d_1 - d_2 y_i^*(\beta_0 + \mathrm{x}_i^{\mathrm{T}}\boldsymbol{\beta})$, $i = 1, \ldots, n$. We conclude that $(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}, \widehat{\xi}_1, \ldots, \widehat{\xi}_n)$ must be the minimizers of (3.4)–(3.5). To show this, we only need to show $\widehat{\xi}_i \geqslant 0$, $i = 1, \ldots, n$. Otherwise, there exists $i_0 \in \{1, \ldots, n\}$, such that $\widehat{\xi}_{i_0} < 0$ and therefore $0 > d_1 - d_2 y_{i_0}^*(\beta_0 + \mathrm{x}_{i_0}^{\mathrm{T}}\boldsymbol{\beta})$. It is easy to see that $(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}, \widehat{\xi}_1, \ldots, \widehat{\xi}_{i_0-1}, 0, \widehat{\xi}_{i_0+1}, \ldots, \widehat{\xi}_n)$ also satisfy (3.7), but clearly achieves a smaller value of the objective function than $(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}, \widehat{\xi}_1, \ldots, \widehat{\xi}_n)$ does. This leads to a contradiction to the assumption on $(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}, \widehat{\xi}_1, \ldots, \widehat{\xi}_n)$. The proof is completed.

Appendix C. Derivation of the algorithm for PsLR

Now we consider the minimization problem (3.6)–(3.7). By adding the Lagrange multipliers, the Lagrange primal function can be written as

$$L_{\mathrm{P},2} = \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + \frac{1}{\gamma_2}\sum_{i=1}^{n}\xi_i^2 - \sum_{i=1}^{n}\alpha_i\{d_2 y_i^*(\beta_0 + \mathrm{x}_i^{\mathrm{T}}\boldsymbol{\beta}) - d_1 + \xi_i\}, \tag{A.1}$$

where $\alpha_i \geqslant 0$, $i = 1, \ldots, n$. By setting the partial derivatives of $L_{\mathrm{P},2}$ with respect to $\boldsymbol{\beta}$, β_0, and ξ_i to zeros, we have that

$$\boldsymbol{\beta} = d_2\sum_{i=1}^{n}\alpha_i y_i^*\mathrm{x}_i, \tag{A.2}$$

$$\sum_{i=1}^{n}\alpha_i y_i^* = 0, \tag{A.3}$$

$$\xi_i = \alpha_i\gamma_2/2, \quad i = 1, \ldots, n. \tag{A.4}$$

Substituting (A.2)–(A.4) to (A.1), we obtain the Wolfe dual function

$$\max_{\{\alpha_i\}} \quad d_1\sum_{i=1}^{n}\alpha_i - d_2^2/2\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i^* y_j^*\langle \mathrm{x}_i, \mathrm{x}_j\rangle - \gamma_2\sum_{i=1}^{n}\alpha_i^2/4$$
$$\text{s.t.} \quad \begin{cases} \alpha_i \geqslant 0, \quad i = 1, \ldots, n, \\ \sum_{i=1}^{n}\alpha_i y_i^* = 0. \end{cases}$$

By using the matrix notations, the above formula can be written as (3.8)–(3.9).

The problem becomes a quadratic programming problem with linear constraints which can be solved by ordinary software. After obtaining $\widehat{\boldsymbol{\alpha}}$, (A.2) can be used to calculate $\widehat{\boldsymbol{\beta}}$ in (3.10).

By the KKT conditions, we know that the optimal solutions should satisfy the following equations:

$$\alpha_i\{d_2 y_i^*(\beta_0 + \mathrm{x}_i^{\mathrm{T}}\boldsymbol{\beta}) - d_1 + \xi_i\} = 0$$

for $i = 1, \ldots, n$. So β_0 can be estimated from it. Choose an index j such that $\widehat{\alpha}_j > 0$. We obtain (3.11). The proof is finished.

Appendix D. Proof of Theorem 1

Note that $E[\mathscr{L}_{\mathrm{Q}}\{Y^* g(\mathrm{X})\}] = E(E[\mathscr{L}_{\mathrm{Q}}\{Y^* g(\mathrm{X})\}|\mathrm{X}])$. To minimize $E[\mathscr{L}_{\mathrm{Q}}\{Y^* g(\mathrm{X})\}]$, we only need to minimize for each fixed x, the function $\mathscr{E}(g) = E[\mathscr{L}_{\mathrm{Q}}\{Y^* g\}|\mathrm{X} = \mathrm{x}]$ with respect to g. Simple calculations give that

$$\begin{aligned}\mathscr{E}(g) &= E[\{\max(0, d_1 - d_2 Y^* g)\}^2|\mathrm{X} = \mathrm{x}] \\ &= d_1^2 m(\mathrm{x})[\max\{0, 1 - (d_2/d_1)g\}]^2 + d_1^2\{1 - m(\mathrm{x})\}[\max\{0, 1 + (d_2/d_1)g\}]^2 \\ &= \begin{cases} d_1^2 m(\mathrm{x})\{1 - (d_2/d_1)g\}^2, & \text{if } g < -d_1/d_2, \\ d_1^2 m(\mathrm{x})\{1 - (d_2/d_1)g\}^2 + d_1^2\{1 - m(\mathrm{x})\}\{1 + (d_2/d_1)g\}^2, & \text{if } |g| \leqslant d_1/d_2, \\ d_1^2\{1 - m(\mathrm{x})\}\{1 + (d_2/d_1)g\}^2, & \text{if } g > d_1/d_2. \end{cases} \end{aligned} \tag{A.5}$$

By a graphical approach, it is easy to see that the minimizer of $\mathscr{E}(g)$ must be located in the interval $[-d_1/d_2, d_1/d_2]$. Denote by $\mathscr{E}_1(g)$ the function $d_1^2 m(\mathrm{x})\{1 - (d_2/d_1)g\}^2 + d_1^2\{1 - m(\mathrm{x})\}\{1 + (d_2/d_1)g\}^2$ in (A.5). Note that $\mathscr{E}_1(g)$ is a quadratic function in g and thus achieves its global minimum at $g_{\mathrm{B}} = d_1/d_2\{2m(\mathrm{x}) - 1\}$. Since $|2m(\mathrm{x}) - 1| \leqslant 1$, the minimizer g_{B} of $\mathscr{E}_1(g)$ falls in the interval $[-d_1/d_2, d_1/d_2]$. Henceforth, we deduce that g_{B} is also the global minimizer of $\mathscr{E}(g)$.

The proof is completed by noticing that $\mathrm{sign}\{g_{\mathrm{B}}(\mathrm{x})\} = \mathrm{sign}\{m(\mathrm{x}) - \frac{1}{2}\}$.

References

Albert, A., Anderson, J.A., 1984. On the existence of maximum likelihood estimates in logistic regression models. Biometrika 71, 1–10.

Alon, U., Barkai, N., Notterman, D.A., Gish, K., Ybarra, S., Mack, D., Levine, A.J., 1999. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proc. Natl. Acad. Sci. USA 96, 6745–6750.

Bradley, P.S., Mangasarian, O.L., 1998. Feature selection via concave minimization and support vector machines. In: Proceedings of the 13th International Conference on Machine Learning, CA, pp. 82–90.

Burges, C.J.C., 1998. A tutorial on support vector machines for pattern recognition. Data Min. Knowl. Discov. 2, 121–167.

Burges, C.J.C., Crisp, D.J., 2000. Uniqueness of the SVM solution. Adv. Neural Inform. Process. Syst. 12, 223–229.

Cristianini, N., Shawe-Taylor, J., 2000. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, Cambridge.

Ding, B., Gentleman, R., 2005. Classification using generalized partial least squares. J. Comput. Graph. Statist. 14, 280–298.

Dudoit, S., Fridlyand, J., Speed, T.P., 2002. Comparison of discrimination methods for the classification of tumors using gene expression data. J. Amer. Statist. Assoc. 97 (457), 77–87.

Eilers, P., Boer, J., van Ommen, G., van Houwelingen, H., 2001. Classification of microarray data with penalized logistic regression. In: Proceedings of SPIE 2001, vol. 4266. Prog. Biomed. Optics Images 2, 187–198.

Evgeniou, T., Pontil, M., Poggio, T., 2000. Regularization networks and support vector machines. Adv. Comput. Math. 13, 1–50.

Firth, D., 1993. Bias reduction of maximum likelihood estimates. Biometrika 80, 27–38.

Fletcher, R., 1987. Practical Methods of Optimization. Wiley, New York.

Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfiel, C.D., Lander, E.S., 1999. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286, 531–536.

Hastie, T., Tibshirani, R., Friedman, J., 2001. The Elements of Statistical Learning. Springer, New York.

Lim, T.-S., Loh, W.-Y., Shih, Y.-S., 2000. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. Mach. Learn. J. 40, 203–228.

McCullagh, P., Nelder, J.A., 1989. Generalized Linear Models. second ed. Chapman & Hall, London.

Osuna, E., Freund, R., Girosi, F., 1997. An improved training algorithm for support vector machines. In: Neural Networks for Signal Processing VII—Proceedings of the 1997 IEEE Signal Processing Society Workshop, pp. 276–285.

Santner, T.J., Duffy, D.E., 1986. A note on A. Albert and J.A. Anderson's conditions for existence of maximum likelihood estimates in logistic regression models. Biometrika 73, 755–758.

Shen, X., Tseng, G.C., Zhang, X., Wong, W.H., 2003. On ψ-learning. J. Amer. Statist. Assoc. 98, 724–734.

Smith, J., Everhart, J., Dickson, W., Knowler, W., Johannes, R., 1988. Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In: Proceedings of the Symposium on Computer Applications and Medical Care. IEEE Computer Society Press, New York, pp. 261–265.

Vapnik, V., 1996. The Nature of Statistical Learning Theory. Springer, New York.

Wahba, G., 1999. Support vector machines, reproducing kernel Hilbert spaces and the randomized GACV. In: Schölkopf, B., Burges, C., Smola, A. (Eds.), Advances in Kernel Methods Support Vector Learning. MIT Press, Cambridge, MA, pp. 69–88.

Weisberg, S., 2005. Applied Linear Regression. third ed. Wiley, New York.

West, M., Blanchette, C., Dressman, H., Huang, E., Ishida, S., Spang, R., Zuzan, H., Olson Jr., J.A., Marks, J.R., Nevins, J.R., 2001. Predicting the clinical status of human breast cancer by using gene expression profiles. Proc. Natl. Acad. Sci. USA 98, 11462–11467.

Wolberg, W.H., Mangasarian, O.L., 1990. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. Proc. Natl. Acad. Sci. USA 87, 9193–9196.

Zhang, C.M., Fu, H., 2006. Masking effects on linear regression in multi-class classification. Statist. Probab. Lett. 76, 1800–1807.

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

Zhu, J., Hastie, T., 2004. Classification of gene microarrays by penalized logistic regression. Biostatistics 5, 427–443.