The predictive Lasso

Minh-Ngoc Tran · David J. Nott · Chenlei Leng

Received: 24 September 2010 / Accepted: 26 July 2011 / Published online: 21 September 2011

Abstract We propose a shrinkage procedure for simultaneous variable selection and estimation in generalized linear models (GLMs) with an explicit predictive motivation. The procedure estimates the coefficients by minimizing the Kullback-Leibler divergence of a set of predictive distributions to the corresponding predictive distributions for the full model, subject to an l_1 constraint on the coefficient vector. This results in selection of a parsimonious model with similar predictive performance to the full model. Thanks to its similar form to the original Lasso problem for GLMs, our procedure can benefit from available l_1-regularization path algorithms. Simulation studies and real data examples confirm the efficiency of our method in terms of predictive performance on future observations.

Keywords Generalized linear models · Kullback-Leibler divergence · Lasso · Optimal prediction · Variable selection

1 Introduction

A primary goal in statistics is to develop algorithms that predict future data well from past observations. In regression problems where a large number of predictors are involved, predictive accuracy in statistical modeling may depend to a large extent on model selection strategies. For generalized linear models (GLMs), for example, a large number of potential predictors are often given in order to reduce modeling bias, and one then would like to select a smaller subset achieving some kind of optimality properties. Popular methods such as the Lasso and its variants can achieve model selection consistency (under some conditions, see, e.g., Zhao and Yu 2006), i.e., if the true model was included in the model set under consideration, these methods would be able to identify (asymptotically) the true model. However, whether or not the true model exists is a controversial issue. For a real dataset, it is believed either that no true model exists or that the true model has an infinite number of parameters (Burnham and Anderson 2002). In this paper we deal with the problem of estimation and variable selection for GLMs with the goal of prediction in mind.

From the Bayesian perspective it is sometimes argued that the full model should be used to achieve the best prediction accuracy (Aitchison 1975; Geisser 1993). However, with many predictors prior specification and elicitation may be difficult, and the full model does not have *interpretability*—a property that is often desirable for many statistical procedures—because it does not tell us in an easily accessible way which predictors are important. Another drawback of using the full model is that if there is a cost associated with data collection then it would be inadvisable to use all of the predictors. This motivates the idea of choosing a submodel whose predictive distribution is close to that of the full model. This idea has been somewhat recognized in the literature. Brown et al. (2002) look at Bayesian model averaging incorporating variable selection for prediction. Tran (2011) and Vehtari and Lampinen (2004) propose model selection methods based on Kullback-Leibler diver-

The authors would like to thank the Editor, Associate Editor and referees for insightful comments which helped to improve the manuscript.

M.-N. Tran (✉) · D.J. Nott · C. Leng
Department of Statistics and Applied Probability, National University of Singapore, Singapore 117546, Singapore
e-mail: ngoctm@nus.edu.sg

Present address:
M.-N. Tran
Australian School of Business, University of New South Wales, Sydney, NSW 2052, Australia
e-mail: minh-ngoc.tran@unsw.edu.au

gence from the predictive distribution of the full model to the predictive distributions of the submodels. These works are motivated by the idea of trading off between prediction accuracy and parsimony. However, these methods are challenging to implement because searching over the whole model space is computationally infeasible. Similar to the idea of the Lasso (Tibshirani 1996), we overcome this problem by using l_1 constraints on the coefficients. By doing this, we can enjoy the computational advantages of the algorithms for convex optimization with l_1 constraints. Unlike the Lasso, however, our approach has an explicit predictive motivation which aims at selecting a useful model with high prediction accuracy. A related approach is considered by Nott and Leng (2010) based on Kullback-Leibler projections, motivated by earlier work of Dupuis and Robert (2003) although these approaches are not based directly on posterior predictive distributions.

For a collection of N predictive distributions obtained from the full model, we write $\mathrm{KL}_i(M_{\text{full}}\|M_\beta)$, $i=1,\ldots,N$ for the Kullback-Leibler divergences from the predictive distributions of the model based on coefficient vector $\boldsymbol{\beta}$ to those of the full model. Our approach in its general form is to solve for $\boldsymbol{\beta}$ the following optimization problem

$$\min_{\beta}\sum_{i=1}^{N}\mathrm{KL}_i(M_{\text{full}}\|M_\beta)+\lambda\|\boldsymbol{\beta}\|_{l_1}$$

with λ a shrinkage parameter as in the original Lasso. The main contribution of the present paper is to motivate and develop such a procedure for variable selection and estimation in GLMs that (i) automatically simultaneously estimates the coefficients and selects significant predictors; (ii) achieves good prediction accuracy; (iii) is broadly applicable; (iv) is computationally efficient. This procedure will be called *the predictive Lasso* or pLasso for short.

The pLasso for GLMs will be presented in Sect. 2. Section 3 presents useful prior specifications which can facilitate computation. In particular, we discuss in more detail the pLasso for linear models and extend our previous discussion to a weighted version of the basic approach. Simulation and real data examples are presented in Sect. 4 to demonstrate the use of the pLasso and to compare it with the adaptive Lasso (Zou 2006) in terms of predictive performance. Section 5 contains concluding remarks.

2 The predictive Lasso

We consider the problem of estimation and variable selection for GLMs with potential covariates $\boldsymbol{x}=(x_0\equiv 1,x_1,\ldots,x_p)'\in\mathcal{X}$ and the response $y\in\mathcal{Y}$. With a suitable link function g, $g(E(y|\boldsymbol{x}))$ is assumed to be a linear combination of $\boldsymbol{x}$

$$g(E(y|\boldsymbol{x}))=\beta_0+\beta_1x_1+\cdots+\beta_px_p=\boldsymbol{x}'\boldsymbol{\beta}. \tag{1}$$

We assume that the covariates x_i are in their final forms, no further transformations are needed (i.e., for various reasons and in order to keep things simple, we restrict ourselves to the linear approximation (1)). The sampling distribution of an observation $\Delta_i=(\boldsymbol{x}_i,\ y_i)$ then is assumed to have the following form

$$\begin{aligned}p(\Delta_i|\boldsymbol{\beta},\phi)&=p(\boldsymbol{x}_i)p(y_i|\boldsymbol{x}_i,\boldsymbol{\beta},\phi)\\&\propto p(\boldsymbol{x}_i)\exp\left(\frac{1}{a(\phi)}\left[y_i\theta(\boldsymbol{x}_i'\boldsymbol{\beta})-b(\theta(\boldsymbol{x}_i'\boldsymbol{\beta}))\right]\right),\end{aligned}$$

where $\boldsymbol{\beta}\in\mathbb{R}^{p+1}$, $\phi>0$ are the coefficient vector and scale parameter, respectively, and θ, a and b are known functions. In order to discuss the methodology in a general setting, we consider predictors $\boldsymbol{x}$ as random. Bayesian variable selection with a random covariate has been considered in a decision theoretic framework where the main concern is prediction of a future observation for which the corresponding predictor is not yet observed (see, for example, Lindley 1968). The case with fixed design points can be considered as a special case, then the density $p(\boldsymbol{x}_i)$ in the above expression can be omitted.

We are concerned with the problem of simultaneous coefficient estimation and variable selection with the goal of prediction in mind. Like the Lasso, we would like to develop a method for simultaneous variable selection and parameter estimation. However, unlike the Lasso our approach has a more explicit predictive motivation, which aims at producing a useful model with high prediction accuracy.

Given the past dataset D and certain priors for parameters $(\boldsymbol{\beta},\ \phi)$ of the full model, the predictive distribution $p(\Delta|D)$ for a future observation $\Delta=(\boldsymbol{x},y)$ is given by

$$\begin{aligned}p(\Delta|D)&=p(\boldsymbol{x}|D)p(y|\boldsymbol{x},D)\\&=p(\boldsymbol{x}|D)\iint p(y|\boldsymbol{x},\boldsymbol{\beta},\phi)p(\boldsymbol{\beta},\phi|D)d\boldsymbol{\beta}d\phi.\end{aligned} \tag{2}$$

We can assume that $p(\boldsymbol{x}|D)\equiv p(\boldsymbol{x})$, i.e., future design points are independent of past data. We propose to estimate the coefficient vector $\boldsymbol{\beta}$ by solving the following optimization problem:

$$\begin{aligned}&\min_{\beta}\iint\log\frac{p(\Delta|D)}{p(\Delta|\boldsymbol{\beta},\phi)}p(\Delta|D)d\boldsymbol{x}dy\\&\text{s.t.}\quad\sum_{j=1}^{p}w_j|\beta_j|\le\tau\end{aligned} \tag{3}$$

where the tuning parameter $\tau\ge 0$ and weights $w_j\ge 0$ are chosen later. As usual in the regularization methods, we do not regularize the intercept. As will become clear shortly, ϕ plays no role in this optimization problem, we can assume at the moment that ϕ is known. Note that the objective function is the Kullback-Leibler divergence from $p(\Delta|\boldsymbol{\beta},\phi)$ to the predictive distribution $p(\Delta|D)$. We refer to this procedure of estimating $\boldsymbol{\beta}$ through the optimization of (3) as the predictive Lasso (pLasso).

Let $\{\Delta_t = (\boldsymbol{x}_t, y_t),\ t = 1, \ldots, T\}$ be Markov chain Monte Carlo (MCMC) samples from the predictive distribution $p(\Delta|D)$. The integral in (3) then can be approximated by the average $(1/T)\sum_{t=1}^{T}\log[p(\Delta_t|D)/p(\Delta_t|\boldsymbol{\beta}, \phi)]$, and (3) becomes

$$\begin{aligned}&\min -\frac{1}{T}\sum_{t=1}^{T}\log p(\Delta_t|\boldsymbol{\beta}, \phi)\\ &\text{s.t.}\quad \sum_{j=1}^{p} w_j|\beta_j| \le \tau,\end{aligned} \tag{4}$$

or more specifically

$$\begin{aligned}&\min \frac{1}{T}\sum_{t=1}^{T}\big[b(\theta(\boldsymbol{x}_t'\boldsymbol{\beta})) - y_t\theta(\boldsymbol{x}_t'\boldsymbol{\beta})\big]\\ &\text{s.t.}\quad \sum_{j=1}^{p} w_j|\beta_j| \le \tau.\end{aligned} \tag{5}$$

This optimization problem is also equivalent to

$$\min \frac{1}{T}\sum_{t=1}^{T}\big[b(\theta(\boldsymbol{x}_t'\boldsymbol{\beta})) - y_t\theta(\boldsymbol{x}_t'\boldsymbol{\beta})\big] + \lambda\sum_{j=1}^{p} w_j|\beta_j| \tag{6}$$

where λ is a tuning parameter. Such an optimization problem is easier to deal with if the objective function is convex. The convexity of the objective function turns out to depend on the link function, and holds for most popular GLMs with the natural link functions.

Often, the integral in $\boldsymbol{x}$ is approximated by a sum over N points $\boldsymbol{x}_1^f, \ldots, \boldsymbol{x}_N^f$. These points might not coincide with the observed design points, they "come from the future" (hence the superscript "f" stands for "future"). For each $\boldsymbol{x}_i^f$, let $\bar{y}_i^f$ be the mean of MCMC samples $\{y_{it}, t = 1, \ldots, T_0\}$ drawn from $p(y_i^f|\boldsymbol{x}_i^f, D)$—the predictive distribution of the future response y_i^f at design point $\boldsymbol{x}_i^f$ given past data D. Then, it is easy to see that (6) becomes

$$\min \frac{1}{N}\sum_{i=1}^{N}\big[b(\theta(\boldsymbol{\beta}'\boldsymbol{x}_i^f)) - \bar{y}_i^f\theta(\boldsymbol{\beta}'\boldsymbol{x}_i^f)\big] + \lambda\sum_{j=1}^{p} w_j|\beta_j|. \tag{7}$$

Note that, under the squared error loss, $\bar{y}_i^f$ is an estimate of the best prediction (w.r.t. the predictive distribution $p(y_i^f|\boldsymbol{x}_i^f, D)$) for the response at $\boldsymbol{x}_i^f$. As will be seen in Sect. 3, for linear regression with a convenient specification of priors there is no need to conduct MCMC because the predictions $\bar{y}_i^f = E(y_i^f|\boldsymbol{x}_i^f, D)$ have a closed form.

We have approximated the integral over $\boldsymbol{x}$ by a sum over N "future" points $\boldsymbol{x}_i^f$, $i = 1, \ldots, N$. Typically, these points are specified depending on the context and/or on the distribution $p(\boldsymbol{x})$ over $\mathcal{X}$. As a default implementation of our procedure, however, we propose to identify the future points $\boldsymbol{x}_i^f$ with the observed training points $\boldsymbol{x}_i$, $i = 1, \ldots, n$. The reason behind this is that if the sample size n is large enough and the observed training points $\boldsymbol{x}_i$ were randomly selected from $p(\boldsymbol{x})$, then by the law of large numbers the integral over $\boldsymbol{x}$ can be well approximated by the sum over $\boldsymbol{x}_i$. In what follows therefore, if not otherwise specified, we consider the pLasso for GLMs in the following form

$$\min \frac{1}{n}\sum_{i=1}^{n}\big[b(\theta(\boldsymbol{x}_i'\boldsymbol{\beta})) - \bar{y}_i^f\theta(\boldsymbol{x}_i'\boldsymbol{\beta})\big] + \lambda\sum_{j=1}^{p} w_j|\beta_j|. \tag{8}$$

Note that the original (adaptive) Lasso for GLMs is

$$\min \frac{1}{n}\sum_{i=1}^{n}\big[b(\theta(\boldsymbol{x}_i'\boldsymbol{\beta})) - y_i\theta(\boldsymbol{x}_i'\boldsymbol{\beta})\big] + \lambda\sum_{j=1}^{p} w_j|\beta_j|. \tag{9}$$

The pLasso in this form differs from the original Lasso only in the way it replaces the observed responses y_i by the predictions $\bar{y}_i^f = E(y_i^f|\boldsymbol{x}_i, D)$. Available routines to solve (9) then can be used for (8).

We have not yet considered the issue of choice of the tuning parameters in the pLasso. As the primary goal of the pLasso is to predict the future, cross-validation is a very natural choice for estimating λ. As in the adaptive Lasso, the weights w_j can be assigned as $1/|\tilde{\beta}_j|$ with $\tilde{\beta}_j$ the MLE of β_j. When the MLE is not available, the Lasso method (more exactly, the non-adaptive pLasso method, i.e., the adaptive penalty term $\lambda\sum w_j|\beta_j|$ in (8) is replaced by $\lambda\sum|\beta_j|$) can be used as a screening tool to effectively eliminate unimportant predictors from consideration in the first stage. The weights corresponding to remaining predictors then will be assigned as $1/\tilde{\beta}_j$ with $\tilde{\beta}_j$ the non-adaptive pLasso estimates. In a Bayesian context it is also natural to consider $\tilde{\beta}_j$ as the posterior mode, and we follow this strategy in the following examples for our pLasso. As suggested by a reviewer, an alternative method for estimating λ is Bayesian estimation (Park and Casella 2008; Leng et al. 2010).

3 Some useful prior specifications

Given the available routines to solve the optimization problem of form (8), all what we need to implement the pLasso is to calculate the quantities $\bar{y}_i^f = E(y_i^f|\boldsymbol{x}_i, D)$. To do so, in general, we first need to specify a useful prior for parameters, determine posterior distributions and then estimate $\bar{y}_i^f = E(y_i^f|\boldsymbol{x}_i, D)$ by MCMC or some other method. However, in some cases there is no need to conduct MCMC. We first present in this section a prior specification for linear models in which the predictions $\bar{y}_i^f$ have closed form. For generalized linear models, we present here two prior specifications. The first is adapted from Chen and Ibrahim (2003) which is interpretable in terms of observables rather than parameters. The second one proposed recently by Gelman et al. (2008) is useful for routine applied use.

3.1 Prior specification for linear models

Consider the linear model

$$\boldsymbol{y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where $\boldsymbol{y}$ is the n-vector of responses, X is an $n \times (p+1)$ design matrix and $\boldsymbol{\epsilon}$ is an n-vector of iid normal errors with mean zero and variance σ^2. The $(p+1)$-vector $\boldsymbol{\beta}$ consists of unknown mean parameters and we consider the situation where σ^2 is also unknown. Consider the conjugate prior specification (O'Hagan and Forster 2004, Chap. 11) $p(\boldsymbol{\beta}, \sigma^2) = p(\sigma^2)p(\boldsymbol{\beta}|\sigma^2)$ in which $p(\sigma^2)$ is inverse gamma

$$p(\sigma^2) = \frac{(a/2)^{(d/2)}}{\Gamma(d/2)}(\sigma^2)^{-d/2-1}\exp\left(-\frac{a}{2\sigma^2}\right)$$

and $p(\boldsymbol{\beta}|\sigma^2)$ is multivariate normal, $N(\boldsymbol{m}, \sigma^2 V)$. With these priors the predictive distribution of a new observation $\Delta = (\boldsymbol{x}, y)$ is $p(\Delta|D) = p(\boldsymbol{x}|D)p(y|\boldsymbol{x}, D)$ with $p(y|\boldsymbol{x}, D) = t_{d+n}(\boldsymbol{x}'\tilde{\boldsymbol{\beta}}, s^2(1+\boldsymbol{x}'\hat{V}\boldsymbol{x}))$ where

$$\tilde{\boldsymbol{\beta}} = (X'X + V^{-1})^{-1}(V^{-1}\boldsymbol{m} + X'\boldsymbol{y}),$$

$$\hat{V} = (V^{-1} + X'X)^{-1},$$

$$s^2 = \frac{a + \boldsymbol{m}'V^{-1}\boldsymbol{m} + \boldsymbol{y}'\boldsymbol{y} - (V^{-1}\boldsymbol{m} + X'\boldsymbol{y})'(V^{-1} + X'X)^{-1}(V^{-1}\boldsymbol{m} + X'\boldsymbol{y})}{n+d-2},$$

$$\hat{\boldsymbol{\beta}} = (X'X)^{-1}X'\boldsymbol{y}.$$

We write $w(\boldsymbol{x}) = 1 + \boldsymbol{x}'\hat{V}\boldsymbol{x}$.

Now consider the predictive Lasso (3) where as usual the integral over $\boldsymbol{x}$ is approximated by a sum over N "future" points $\boldsymbol{x}_i^f$. Then equivalently, we need to minimize (the scale ϕ is now re-denoted by σ^2)

$$\sum_{i=1}^{N}\int\left[-\log p(y_i^f|\boldsymbol{x}_i^f, \boldsymbol{\beta}, \sigma^2)\right]p(y_i^f|\boldsymbol{x}_i^f, D)dy_i^f \qquad (10)$$
$$\text{s.t.} \quad \sum_{j=1}^{p} w_j|\beta_j| \le \tau.$$

Noting that

$$\log p(y_i^f|\boldsymbol{x}_i^f, \boldsymbol{\beta}, \sigma^2) = -\frac{1}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}(y_i^f - (\boldsymbol{x}_i^f)'\boldsymbol{\beta})^2,$$

minimizing (10) is equivalent to minimizing

$$\frac{N}{2}\log\sigma^2 + \frac{1}{2\sigma^2}\sum_{i=1}^{N} E\left((y_i^f - (\boldsymbol{x}_i^f)'\boldsymbol{\beta})^2|\boldsymbol{x}_i^f, D\right) \qquad (11)$$
$$\text{s.t.} \quad \sum_{j=1}^{p} w_j|\beta_j| \le \tau.$$

With the closed form of the predictive distribution as a t-distribution we have

$$E\left((y_i^f - (\boldsymbol{x}_i^f)'\boldsymbol{\beta})^2|\boldsymbol{x}_i^f, D\right) = s^2 w(\boldsymbol{x}_i^f) + \left((\boldsymbol{x}_i^f)'\tilde{\boldsymbol{\beta}} - (\boldsymbol{x}_i^f)'\boldsymbol{\beta}\right)^2.$$

Substituting this into (11) we must minimize

$$\frac{N}{2}\log\sigma^2 + \frac{1}{2\sigma^2}\sum_{i=1}^{N} s^2 w(\boldsymbol{x}_i^f) + \frac{1}{2\sigma^2}\sum_{i=1}^{N}\left((\boldsymbol{x}_i^f)'\tilde{\boldsymbol{\beta}} - (\boldsymbol{x}_i^f)'\boldsymbol{\beta}\right)^2 \qquad (12)$$

subject to the constraint. Minimizing this as a function of $\boldsymbol{\beta}$ amounts as before to an ordinary Lasso problem where the responses are replaced with the fitted values from the full model at the future design points $\boldsymbol{x}_i^f$, $i = 1, \ldots, N$. In the case with a non-informative prior, $n > p$, and the $\boldsymbol{x}_i^f$ as the observed design points $\boldsymbol{x}_i$, this is the ordinary Lasso, since in this case $\tilde{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}$ and for the least squares estimator

$$\sum_{i=1}^{n}(y_i - \boldsymbol{x}_i'\boldsymbol{\beta})^2 = \sum_{i=1}^{n}(y_i - \boldsymbol{x}_i'\hat{\boldsymbol{\beta}})^2 + \sum_{i=1}^{n}(\boldsymbol{x}_i'\hat{\boldsymbol{\beta}} - \boldsymbol{x}_i'\boldsymbol{\beta})^2$$

where the first term on the right hand side does not depend on $\boldsymbol{\beta}$.

If (12) has been minimized with respect to $\boldsymbol{\beta}$ subject to the constraint to obtain an estimate $\hat{\boldsymbol{\beta}}_{\text{pLasso}}$ (this in general depends on the constraint τ but we suppress this in the notation) then substituting in $\hat{\boldsymbol{\beta}}_{\text{pLasso}}$ and minimizing with respect to σ^2 gives

$$\begin{aligned}
&\hat{\sigma}^2_{\text{pLasso}} \\
&\quad = \frac{\sum_{i=1}^N \text{Var}(y_i^f|\boldsymbol{x}_i^f, D) + \sum_{i=1}^N ((\boldsymbol{x}_i^f)'\tilde{\boldsymbol{\beta}} - (\boldsymbol{x}_i^f)'\hat{\boldsymbol{\beta}}_{\text{pLasso}})^2}{N} \\
&\quad = \frac{\sum_{i=1}^N s^2 w(\boldsymbol{x}_i^f) + \sum_{i=1}^N ((\boldsymbol{x}_i^f)'\tilde{\boldsymbol{\beta}} - (\boldsymbol{x}_i^f)'\hat{\boldsymbol{\beta}}_{\text{pLasso}})^2}{N}. \qquad (13)
\end{aligned}$$

The weighted version of pLasso One extension that gives different results to the ordinary Lasso in the noninformative case with the $\boldsymbol{x}_i^f$ the observed design points $\boldsymbol{x}_i$ is the following. Suppose that instead of considering predictive distributions in our predictive Lasso objective function where the variance does not depend on $\boldsymbol{x}$ we predict y_i^f with

$$p\big(y_i^f|\boldsymbol{\beta}, \sigma^2 w(\boldsymbol{x}_i^f)\big) = N\big((x_i^f)'\boldsymbol{\beta}, \sigma^2 w(\boldsymbol{x}_i^f)\big).$$

That is, we allow our normal form predictive distributions to have variances which vary in proportion to the true predictive variances in the full model $\text{Var}(y_i^f|\boldsymbol{x}_i^f, D)$. The standard deviation in the full model $\sqrt{\text{Var}(y_i^f|\boldsymbol{x}_i^f, D)}$ is often considered a more realistic estimate of the standard error, because it incorporates model uncertainty. We now consider minimization of

$$\sum_{i=1}^N \int \big[-\log p(y_i^f|\boldsymbol{\beta}, \sigma^2 w(\boldsymbol{x}_i^f))\big] p(y_i^f|\boldsymbol{x}_i^f, D) dy_i^f$$

subject to the constraint and following a similar argument to our previous one we must minimize

$$\sum_{i=1}^N \frac{1}{w(\boldsymbol{x}_i^f)} \big((\boldsymbol{x}_i^f)'\tilde{\boldsymbol{\beta}} - (\boldsymbol{x}_i^f)'\boldsymbol{\beta}\big)^2$$

subject to the constraint in order to estimate $\boldsymbol{\beta}$. This is similar to before, but now with weights of $1/w(\boldsymbol{x}_i^f)$ for the different design points. We will refer to this procedure as the weighted pLasso (wpLasso). After $\boldsymbol{\beta}$ has been estimated as $\hat{\boldsymbol{\beta}}_{\text{wpLasso}}$ say, the minimization with respect to σ^2 gives

$$\begin{aligned}
\hat{\sigma}^2_{\text{wpLasso}} &= \frac{\sum_{i=1}^N \frac{1}{w(x_i)} \text{Var}(y_i^f|\boldsymbol{x}_i^f, D) + \sum_{i=1}^N \frac{1}{w(x_i^f)} ((\boldsymbol{x}_i^f)'\tilde{\boldsymbol{\beta}} - (\boldsymbol{x}_i^f)'\tilde{\boldsymbol{\beta}}_{\text{wpLasso}})^2}{N} \\
&= \frac{\sum_{i=1}^N s^2 + \sum_{i=1}^n \frac{1}{w(x_i^f)} ((\boldsymbol{x}_i^f)'\tilde{\boldsymbol{\beta}} - (\boldsymbol{x}_i^f)'\tilde{\boldsymbol{\beta}}_{\text{wpLasso}})^2}{N}.
\end{aligned}$$

Elicitation of hyperparameters We now discuss on the choice of the hyperparameters $\boldsymbol{m}$ and V. There are many different ways proposed for choosing the matrix V. For example, Zellner (1986) proposed the so-called *g-prior* in which V is set equal to $c(X'X)^{-1}$ with some $c > 0$ ($c = n$ is a common choice). Raftery et al. (1997) proposed an alternative where V is a block-diagonal matrix. For noncategorical covariates, V is a diagonal matrix $\text{diag}(s_y^2, \kappa^2 s_1^{-2}, \dots, \kappa^2 s_p^{-2})$ where s_y^2 is the sample variance of $\boldsymbol{y}$, and s_i^2 are the variances of the columns of X. For a categorical covariate, the corresponding diagonal element will be a matrix induced from the corresponding dummy variables. Raftery et al. (1997) proposed a value of 2.85 for κ together with $a = 0.72$ and $d = 2.58$. For the parameter $\boldsymbol{m}$, they proposed the default value of $\boldsymbol{m} = (\hat{\beta}_0^{\text{OLS}}, 0, \dots, 0)'$ where $\hat{\beta}_0^{\text{OLS}}$ is the OLS estimate of β_0. An alternative is $\boldsymbol{m} = \boldsymbol{0}$. These two choices of $\boldsymbol{m}$ often lead to very similar inferences. We will use the setup of Raftery et al. (1997) in our following numerical examples.

Comparison with the elastic net The pLasso method can be viewed as a two-stage procedure: a prior-based regularization in the first stage, and a Lasso-type thresholding in the second stage. As suggested by a reviewer, this makes the pLasso method closely related to the elastic net method of Zou and Hastie (2005), which consists of a ridge-type regularization followed by a Lasso-type thresholding. The elastic net (enet) method has proved to be superior to the Lasso in terms of prediction accuracy. Therefore, a comparison between the two methods may provide insight into the pLasso.

Consider the elastic net problem

$$\begin{aligned}
&\min_{\beta} (\boldsymbol{y} - X\boldsymbol{\beta})'(\boldsymbol{y} - X\boldsymbol{\beta}) + \boldsymbol{\beta}' V^{-1}\boldsymbol{\beta} \\
&\text{s.t.} \quad \sum_{j=1}^p |\beta_j| \le \tau,
\end{aligned} \qquad (14)$$

which is equivalent to

$$\min_{\beta}(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}})'(X'X+V^{-1})(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}})$$
$$\text{s.t.} \quad \sum_{j=1}^{p}|\beta_j| \le \tau. \tag{15}$$

If we select the future predictors $\boldsymbol{x}_i^f$ such that $X^{f\prime}X^f = X'X + V^{-1}$ (we may simply select X^f as the square roof of $X'X + V^{-1}$), then the pLasso problem (12) is exactly the same as the elastic net problem (15). Now, with $\boldsymbol{x}_i^f \equiv \boldsymbol{x}_i$, the pLasso problem (12) is

$$\min_{\beta}(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}})'X'X(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}})$$
$$\text{s.t.} \quad \sum_{j=1}^{p}|\beta_j| \le \tau. \tag{16}$$

If we use the Zellner g-prior $V = c(X'X)^{-1}$ in the first stage of elastic net, then the elastic net problem (15) is equivalent to the pLasso problem (16). If X is centered and has orthonormal columns, using the Raftery et al. prior also leads to the equivalence between the two methods.

That is, under some conditions, the two methods are equivalent to each other. However, the pLasso framework is more flexible than the elastic net. Unlike the elastic net, the pLasso is not restricted to the penalty that comes from use of a normal prior. Other regularization priors, such as those based on the Cauchy distribution, can be used too (Sect. 4). Furthermore, no restriction is put on the predictive distribution $p(\Delta|D)$ of the full model. Any model that gives good predictions $\bar{y}_i^f$ can be used.

3.2 Prior specifications for generalized linear models

There is an extensive literature on prior specifications for GLMs. We will briefly present here two of them: the first one is due to Chen and Ibrahim (2003) and the second is proposed recently by Gelman et al. (2008).

The Chen and Ibrahim prior Recall that the sampling distribution of observables $\boldsymbol{y} = (y_1, \dots, y_n)$ in the GLM case is

$$p(\boldsymbol{y}|X,\boldsymbol{\beta},\phi) \propto \exp\left(\sum_{1}^{n}\frac{1}{a(\phi)}\left[y_i\theta(\boldsymbol{x}_i'\boldsymbol{\beta}) - b(\theta(\boldsymbol{x}_i'\boldsymbol{\beta}))\right]\right)$$
$$= \exp\left(\frac{1}{a(\phi)}\left[\boldsymbol{y}'\boldsymbol{\theta} - \mathbb{1}'\boldsymbol{b}(\boldsymbol{\theta})\right]\right)$$

where $\boldsymbol{\theta} = \boldsymbol{\theta}(\boldsymbol{\beta}) = (\theta_1, \dots, \theta_n)'$, $\theta_i = \theta(\boldsymbol{x}_i'\boldsymbol{\beta})$, $\boldsymbol{b}(\boldsymbol{\theta}) = (b(\theta_1), \dots, b(\theta_n))'$ and $\mathbb{1}$ is an n-vector of 1s. For ease of exposition, we assume that ϕ is known (and therefore suppressed in the notation), as, for example, in logistic and Poisson regression. Chen and Ibrahim (2003) proposed the following prior for $\boldsymbol{\beta}$

$$p(\boldsymbol{\beta}) \propto \exp\left(\gamma_0\frac{1}{a(\phi)}\left[\boldsymbol{\alpha}_0'\boldsymbol{\theta} - \mathbb{1}'\boldsymbol{b}(\boldsymbol{\theta})\right]\right) \tag{17}$$

where $\gamma_0 \ge 0$ and $\boldsymbol{\alpha}_0 \in \mathbb{R}^n$ are hyperparameters determined later on. Denote this distribution by $\boldsymbol{\beta}|\phi \sim D(\gamma_0, \boldsymbol{\alpha}_0)$. They proved that the prior (17) is proper and that this prior is conjugate with the posterior $\boldsymbol{\beta}|X,\boldsymbol{y} \sim D(1+\gamma_0, (\gamma_0\boldsymbol{\alpha}_0 + \boldsymbol{y})/(1+\gamma_0))$.

As shown by Chen and Ibrahim (2003), $E(\boldsymbol{y}) = \boldsymbol{\alpha}_0$, it is natural to choose $\boldsymbol{\alpha}_0$ as a prior guess for $E(\boldsymbol{y})$. Therefore, in practice, $\boldsymbol{\alpha}_0$ should be obtained from experts in the field although default empirical Bayes alternatives such as choosing $\boldsymbol{\alpha}_0$ as the fitted values based on the MLE or other methods are also possible. The parameter γ_0 weighs the importance of the prior guess. In general, γ_0 should be taken such that $\gamma_0 = \gamma_0(n) \to 0$ as $n \to \infty$, i.e., the prior has less influence when more data is available. An advantage of this prior specification is that it is interpretable in terms of observables rather than parameters which are sometimes not easy to elicit.

The Gelman et al. prior Gelman et al. (2008) proposed a weakly informative prior distribution for GLMs, constructed by first standardizing the covariates to have mean zero and standard deviation 0.5, and then putting independent t-distributions on the coefficients. As a default choice, they recommended a central Cauchy distribution with scale 10 for the intercept and central Cauchy distributions with scale 2.5 for other coefficients. As argued by Gelman et al. (2008), this prior specification has many advantages; besides, it works in an automatic fashion with no hyperparameter elicitation needed.

Recall that all what we need to implement the pLasso is to calculate the quantities $\bar{y}_i^f = E(y_i^f|\boldsymbol{x}_i, D)$. After the prior has been specified, $\bar{y}_i^f$ can be estimated by MCMC or some other method. It is well-known that

$$E(\boldsymbol{y}|X,\boldsymbol{\beta}) = \dot{\boldsymbol{b}}(\boldsymbol{\theta}) = (\dot{b}(\theta_1), \dots, \dot{b}(\theta_n))',$$

so that

$$\bar{\boldsymbol{y}}^f = E(\boldsymbol{y}^f|X,\boldsymbol{y}) = E_{\beta|X,y}[E(\boldsymbol{y}^f|X,\boldsymbol{\beta})]$$
$$= E_{\beta|X,y}[\dot{\boldsymbol{b}}(\boldsymbol{\theta}(\boldsymbol{\beta}))] \tag{18}$$

which can be easily estimated by MCMC samples from the posterior distribution $\boldsymbol{\beta}|X,\boldsymbol{y}$.

A procedure for fitting GLMs with the Gelman et al. prior has been implemented in R by Gelman et al. (available online at http://cran.r-project.org/web/packages/arm). In the following numerical examples for logistic regression where no expert advice is available, we use the default prior of Gelman et al. For high-dimensional cases where using MCMC may be time consuming, we suggest using the plug-in predictive density to estimate the predictions $\bar{y}_i^f$. Our experiences show that this is very fast compared to MCMC.

4 Experiments

In this section, we study the pLasso through simulations and real data examples. We use the convenient prior specifications as in Sect. 3. The tuning parameter λ is selected by 5-fold cross-validation. The code implementing the pLasso and conducting the examples is available on the authors' websites. For the elastic net, we use the Matlab implementation written by Hui Jiang and available at http://www-stat.stanford.edu/~tibs/glmnet-matlab/. The tuning parameters λ_1 and λ_2 are tuned using cross-validation.

A popular measure of predictive ability is the *partial predictive score* (PPS) (Good 1952; Geisser 1980; Hoeting et al. 1999). Suppose that the data is split into two parts, the training set D^T and the prediction set D^P. The partial predictive score of the distributions induced by model parameters $\boldsymbol{\theta}^*$ is defined as

$$\text{PPS} = -\frac{1}{|D^P|} \sum_{\Delta=(x,y)\in D^P} \log p(y|\boldsymbol{x}, \boldsymbol{\theta}^*). \tag{19}$$

It is understood that smaller PPS means better predictive performance.

Although the PPS is widely used in practice, Gneiting and Raftery (2007) argued that it is sometimes sensitive to extreme cases because the use of logarithmic scale puts a high penalty on low probability cases, and suggested using the so-called continuous ranked probability score (CRPS) as an alternative. Let F be the cumulative distribution function (cdf) of the predictive distribution in use and x be an actual observation. The CRPS is defined as

$$\text{CRPS}(F, x) = -\int_R (F(y) - \mathbb{1}_{y\geq x})^2 dy$$

which corresponds to the integral of the Brier scores (Hersbach 2000). A problem with using CRPS is that the above integral is in general not available in closed form and needs to be estimated in some way. However, when F is the cdf of the normal distribution with mean μ and variance σ^2, the CRPS is given by Gneiting and Raftery (2007, p. 367)

$$\begin{aligned}&\text{CRPS}(N(\mu, \sigma^2), x)\\ &\quad= \sigma\left[\frac{1}{\sqrt{\pi}} - 2\varphi\left(\frac{x-\mu}{\sigma}\right) - \frac{x-\mu}{\sigma}\left(2\phi\left(\frac{x-\mu}{\sigma}\right) - 1\right)\right]\end{aligned}$$

where φ and ϕ are pdf and cdf of the standard Gaussian variable; when F is the cdf of a Bernoulli variable X with probability of success $p = P(X = 1)$, the CRPS is given by

$$\text{CRPS}(F(p), x = 0) = -p^2 \quad \text{and}$$

$$\text{CRPS}(F(p), x = 1) = -(1 - p)^2.$$

In our paper, the CRPS (evaluated on a prediction set D^P) of the predictive distributions induced by model parameters $\boldsymbol{\theta}^*$ is defined as

$$\text{CRPS} \equiv \text{CRPS}(\boldsymbol{\theta}^*) = -\frac{1}{|D^P|} \sum_{\Delta\in D^P} \text{CRPS}(F(\boldsymbol{\theta}^*), \Delta). \tag{20}$$

Under this formulation, it is (similar to PPS) understood that smaller CRPS means better predictive performance.

In the simulation studies below, we also use mean squared errors (MSE) in terms of coefficients and numbers

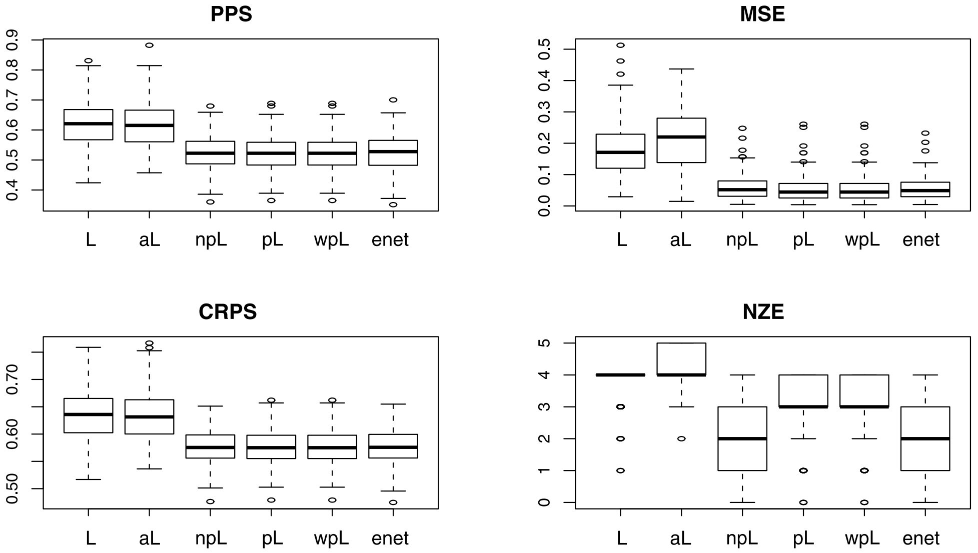

Fig. 1 Boxplots of performance measures over replications for comparing the methods in linear regression: small p case with normal predictors, $n = 200$ and $\sigma = 1$

Table 1 Simulation results for linear regression: small p and normal predictors. The numbers in parentheses are standard deviations

$n_T = n_P$	σ	Measure	Lasso	aLasso	npLasso	pLasso	wpLasso	enet
50	1	PPS	0.79 (0.17)	0.78 (0.16)	0.61 (0.13)	0.61 (0.13)	0.60 (0.12)	0.64 (0.17)
		MSE	0.53 (0.28)	0.55 (0.31)	0.23 (0.13)	0.24 (0.16)	0.24 (0.17)	0.26 (0.13)
		NZE	4.02 (0.82)	5.02 (0.64)	2.47 (1.17)	3.42 (1.02)	3.51 (1.03)	2.08 (1.23)
		CRPS	0.73 (0.11)	0.69 (0.11)	0.61 (0.07)	0.60 (0.06)	0.60 (0.07)	0.62 (0.07)
		CFR	0.48	0.18	0.16	0.43	0.48	0.10
	3	PPS	1.85 (0.16)	1.84 (0.17)	1.69 (0.14)	1.70 (0.14)	1.69 (0.13)	1.72 (0.16)
		MSE	4.24 (2.23)	4.53 (2.58)	2.08 (1.63)	2.37 (1.80)	2.24 (1.76)	1.96 (1.56)
		NZE	5.97 (0.90)	6.64 (0.58)	3.40 (1.56)	4.22 (1.38)	4.46 (1.37)	3.07 (1.62)
		CRPS	2.17 (0.32)	2.17 (0.35)	1.83 (0.20)	1.82 (0.21)	1.84 (0.22)	1.83 (0.21)
		CFR	0.01	0	0.07	0.04	0.04	0.06
100	1	PPS	0.67 (0.11)	0.66 (0.11)	0.54 (0.09)	0.54 (0.09)	0.54 (0.08)	0.55 (0.10)
		MSE	0.30 (0.14)	0.33 (0.13)	0.12 (0.09)	0.12 (0.09)	0.12 (0.09)	0.12 (0.08)
		NZE	3.81 (0.64)	4.64 (0.56)	2.14 (1.14)	3.25 (0.94)	3.18 (1.10)	1.98 (1.27)
		CRPS	0.67 (0.07)	0.65 (0.07)	0.59 (0.05)	0.58 (0.05)	0.58 (0.05)	0.59 (0.05)
		CFR	0.67	0.34	0.14	0.47	0.48	0.14
	3	PPS	1.77 (0.10)	1.76 (0.10)	1.64 (0.08)	1.64 (0.08)	1.64 (0.08)	1.65 (0.08)
		MSE	3.06 (1.41)	3.54 (1.78)	1.01 (0.76)	0.93 (0.79)	0.92 (0.83)	0.92 (0.59)
		NZE	5.60 (0.84)	6.39 (0.69)	2.94 (1.43)	3.84 (1.20)	3.90 (1.36)	2.59 (1.45)
		CRPS	2.00 (0.20)	1.98 (0.20)	1.76 (0.13)	1.75 (0.13)	1.75 (0.13)	1.75 (0.13)
		CFR	0.04	0	0.11	0.10	0.08	0.09
200	1	PPS	0.62 (0.07)	0.61 (0.07)	0.52 (0.05)	0.52 (0.05)	0.52 (0.05)	0.52 (0.06)
		MSE	0.18 (0.07)	0.22 (0.09)	0.06 (0.04)	0.05 (0.03)	0.05 (0.04)	0.05 (0.03)
		NZE	3.84 (0.42)	4.31 (0.50)	2.18 (1.26)	3.19 (0.97)	3.11 (1.19)	1.92 (1.31)
		CRPS	0.63 (0.05)	0.63 (0.04)	0.57 (0.03)	0.57 (0.03)	0.57 (0.03)	0.58 (0.03)
		CFR	0.85	0.66	0.15	0.48	0.50	0.11
	3	PPS	1.72 (0.07)	1.71 (0.07)	1.62 (0.05)	1.62 (0.05)	1.62 (0.05)	1.62 (0.06)
		MSE	1.59 (0.65)	2.04 (0.88)	0.47 (0.26)	0.42 (0.30)	0.42 (0.30)	0.46 (0.25)
		NZE	5.44 (0.74)	6.22 (0.62)	2.53 (1.32)	3.63 (1.14)	3.62 (1.31)	2.23 (1.31)
		CRPS	1.90 (0.14)	1.89 (0.13)	1.73 (0.09)	1.73 (0.09)	1.73 (0.09)	1.73 (0.09)
		CFR	0.07	0	0.14	0.20	0.20	0.13

of zero-estimated (NZE) coefficients to measure the performance. We also report correctly-fitted rates (CFR), i.e., frequency of correctly-fitted models. Note, however, that our methodology does not focus on finding the true model but useful one for prediction.

4.1 Simulation studies

A simulation study for linear regression Consider the following linear model

$$y = 2 + \boldsymbol{x}'\boldsymbol{\beta} + \sigma\epsilon \tag{21}$$

where $\boldsymbol{\beta} = (3,\ 1.5,\ 0,\ 0,\ 0.5,\ 0.5,\ 0,\ 0)'$ (so that there are some main and also small effects), ϵ is iid $N(0,1)$, and $\sigma > 0$ is the noise level. We want to compare the performance of the pLasso and the wpLasso to that of the adaptive Lasso (aLasso). We also consider the original Lasso and the non-adaptive pLasso (i.e., the adaptive penalty term $\lambda\sum w_j|\beta_j|$ in (8) is replaced by $\lambda\sum|\beta_j|$) which will be abbreviated as npLasso.

In our first simulation study, design points $\boldsymbol{x}_j$ are simulated from a multivariate normal distribution $N_8(\mathbf{0}, \Sigma)$ with $\sigma_{ij} = 0.5^{|i-j|}$. We first generate from model (21) a dataset which serves as the training set D^T. Another dataset D^P then is generated, which is used to test the predictive performance. Table 1 presents the PPS (after ignoring the constants independent of models), MSE, NZE, CRPS and CFR averaged over 500 replications with various factors $n = n_T$

Table 2 Simulation results for linear regression: small p and long-tailed t-distribution predictors. The numbers in parentheses are standard deviations

$n_T = n_P$	σ	Measure	Lasso	aLasso	npLasso	pLasso	wpLasso	enet
50	1	PPS	7.07 (43.7)	5.44 (27.3)	8.24 (96.5)	2.43 (10.8)	0.70 (0.30)	6.78 (49.5)
		MSE	0.20 (0.19)	0.23 (0.26)	0.44 (4.81)	0.40 (4.44)	0.42 (4.70)	0.92 (13.2)
		NZE	3.01 (0.99)	3.88 (0.81)	1.66 (1.16)	3.13 (1.08)	3.22 (1.21)	2.26 (1.23)
		CRPS	0.94 (0.60)	0.94 (0.54)	0.90 (1.22)	0.84 (1.09)	0.84 (1.09)	0.92 (1.61)
		CFR	0.32	0.61	0.06	0.49	0.58	0.11
	3	PPS	3.15 (4.03)	3.29 (8.47)	3.00 (6.47)	2.81 (6.13)	1.74 (0.16)	3.61 (16.92)
		MSE	1.39 (1.11)	1.71 (1.56)	0.84 (0.71)	0.77 (0.69)	0.78 (0.72)	0.84 (1.14)
		NZE	4.06 (1.32)	5.12 (1.11)	2.22 (1.33)	3.45 (1.14)	3.71 (1.24)	2.41 (1.50)
		CRPS	2.54 (1.20)	2.69 (1.19)	2.17 (0.73)	2.16 (0.79)	2.15 (0.73)	2.18 (0.86)
		CFR	0.15	0.09	0.06	0.22	0.26	0.10
100	1	PPS	3.90 (35.6)	1.86 (7.10)	1.01 (2.69)	0.86 (1.83)	0.61 (0.20)	1.52 (8.95)
		MSE	0.07 (0.07)	0.09 (0.27)	0.07 (0.31)	0.06 (0.30)	0.06 (0.31)	0.13 (0.91)
		NZE	3.11 (1.08)	3.69 (0.73)	1.62 (1.21)	3.13 (0.96)	3.20 (1.13)	2.44 (1.24)
		CRPS	0.76 (0.30)	0.73 (0.26)	0.68 (0.31)	0.66 (0.30)	0.66 (0.30)	0.72 (0.51)
		CFR	0.47	0.75	0.06	0.44	0.54	0.19
	3	PPS	7.12 (74.8)	14.9 (51.5)	1.96 (1.74)	2.00 (3.08)	1.67 (0.11)	3.61 (28.8)
		MSE	0.57 (0.39)	0.68 (0.65)	0.31 (0.54)	0.28 (0.54)	0.29 (0.54)	0.30 (0.53)
		NZE	3.45 (1.26)	4.66 (0.98)	1.84 (1.16)	3.19 (1.03)	3.26 (1.17)	2.09 (1.35)
		CRPS	2.30 (1.25)	2.35 (1.86)	1.91 (0.34)	1.88 (0.36)	1.89 (0.36)	1.94 (0.67)
		CFR	0.26	0.29	0.07	0.45	0.53	0.13
200	1	PPS	1.26 (3.04)	1.24 (4.26)	0.71 (0.65)	0.74 (1.11)	0.57 (0.19)	0.93 (2.39)
		MSE	0.03 (0.03)	0.04 (0.06)	0.03 (0.13)	0.03 (0.13)	0.03 (0.13)	0.08 (0.38)
		NZE	3.26 (0.91)	3.77 (0.55)	1.48 (1.28)	3.00 (1.02)	3.09 (1.14)	2.76 (1.12)
		CRPS	0.67 (0.11)	0.67 (0.14)	0.63 (0.18)	0.62 (0.18)	0.62 (0.18)	0.68 (0.35)
		CFR	0.51	0.82	0.07	0.38	0.46	0.24
	3	PPS	3.47 (15.3)	3.57 (11.0)	2.05 (3.12)	1.98 (2.92)	1.63 (0.06)	2.26 (4.40)
		MSE	0.27 (0.22)	0.31 (0.26)	0.12 (0.11)	0.10 (0.11)	0.10 (0.12)	0.14 (0.21)
		NZE	3.29 (0.92)	4.14 (0.84)	1.85 (1.15)	3.21 (0.91)	3.32 (1.06)	2.24 (1.21)
		CRPS	2.05 (0.49)	2.08 (0.51)	1.81 (0.23)	1.79 (0.21)	1.79 (0.21)	1.85 (0.36)
		CFR	0.46	0.60	0.08	0.48	0.60	0.13

(size of training set) = n_P (size of prediction set) and σ. The numbers in parentheses are standard deviations. In general, the enet and pLasso methods appear to have similar predictive performance, and outperform (having smaller PPS and CRPS) the Lasso methods. The results also suggest that, in the current setting, the pLasso and wpLasso work slightly better than the elastic net. As one may expect for predictively motivated methods, models selected by the elastic net and pLasso methods are less sparse (having smaller NZE) than those selected by the Lasso and aLasso. In particular, the pLasso methods constantly produce sparser models than the enet, while still having good predictive performance. In order to better compare the behaviour of the methods over the replications, we plot in Fig. 1 boxplots for the case $n = 200$ and $\sigma = 1$.

In our second simulation study, design points $\boldsymbol{x}_j$ are simulated from a multivariate t-distribution with degrees of freedom being 1.5. By doing so, we intend to simulate situations in which some predictors have high leverage, i.e., their distributions have long tails. The simulation results are presented in Table 2. As one may expect, the wpLasso appears to work better and to be more stable than the other methods because the variance is modeled to vary in proportion to the true predictive variance. Boxplots of the measures over replications for the case $n = 200$, $s = 1$ are given in Fig. 2.

In our last simulation study, we try a high-dimensional example. We consider the linear model (21) with $p = 100$

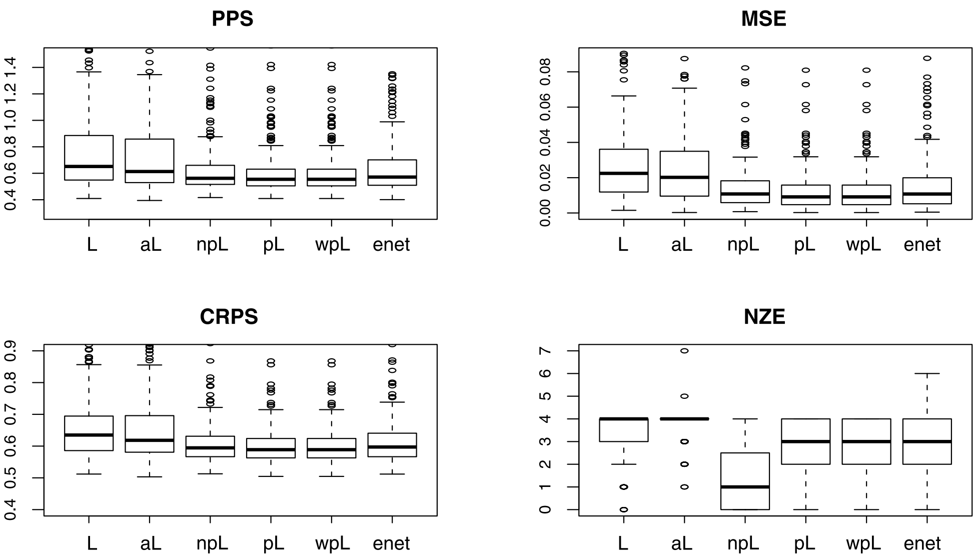

Fig. 2 Boxplots of performance measures over replications for comparing the methods in linear regression: small p case with long-tailed predictors, $n = 200$ and $\sigma = 1$. For ease of comparison, the y axes have been scaled so that the extreme outliers are omitted

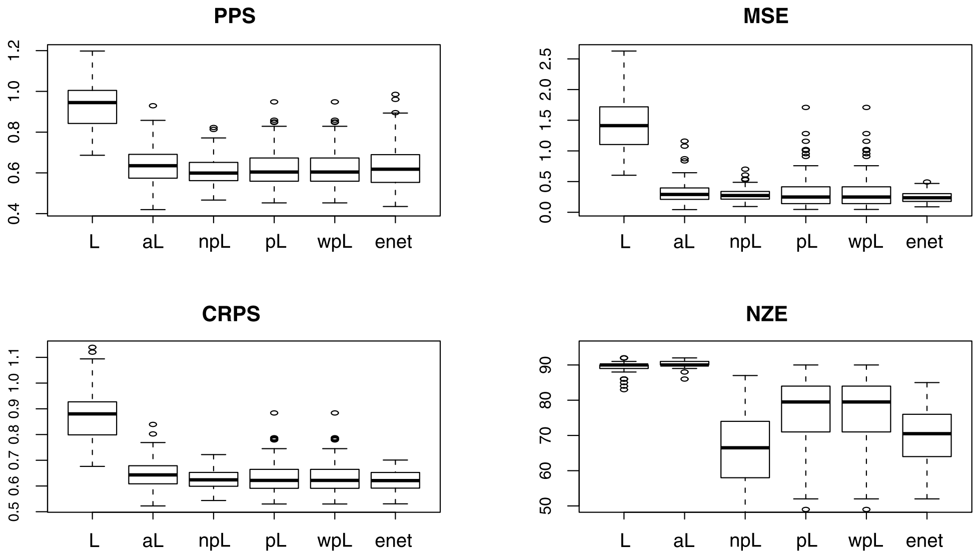

Fig. 3 Boxplots of performance measures over replications for comparing the methods in linear regression: large p case with normal predictors, $n = 200$ and $\sigma = 1$. For ease of comparison, the y axes have been scaled so that the extreme outliers are omitted

and most of the coefficients are zero except $\beta_j = 5,\ j = 10, 20, \ldots, 100$. The results reported in Table 3 suggest that, with large n, the pLasso and wpLasso compare favorably with the others in this example. The aLasso is the best in terms of identifying the true model. Note, however, that our main goal is to select useful models for making good predictions rather than selecting the true one—a philosophy that is likely to be more useful in many real data applications where all models under consideration may be misspecified. Boxplots for the case $n = 200,\ \sigma = 1$ are given in Fig. 3.

In summary, the simulation study reveals that the elastic net and pLasso methods have, in general, similar predictive ability and show a better predictive performance than the Lasso methods. In some cases, the pLasso and wpLasso appear to work slightly better than the elastic net: they select sparser models while still enjoying good predictive performance.

Bayesian adaptive pLasso As mentioned in the last paragraph of Sect. 2, in a Bayesian context, it would be natural to select shrinkage parameters using Bayesian estimation. A full Bayesian treatment of the Lasso was first developed in Park and Casella (2008), in which Gibbs samples from the hierarchical Bayesian counterpart of the Lasso are used for inference about λ. The non-adaptive pLasso with Bayesian estimation for λ will be in the following referred to as Bayesian pLasso (BpLasso). The Bayesian treatment of the Lasso is further extended in Leng et al. (2010) to the adaptive Lasso and other variants. Leng et al. (2010) propose a hybrid Bayesian-frequentist method for simultaneous variable selection and coefficient estimation, in which the adaptive shrinkage parameters $\lambda_i = \lambda w_i$ in optimization problem (9) are estimated using Gibbs samples from a hierarchical Bayesian formulation. The adaptive pLasso with Bayesian

Table 3 Simulation results for linear regression: large p and normal predictors. The numbers in parentheses are standard deviations

$n_T = n_P$	σ	Measure	Lasso	aLasso	npLasso	pLasso	wpLasso	enet
50	1	PPS	3.39 (5.70)	2.28 (6.65)	2.49 (2.17)	2.59 (8.48)	2.58 (0.23)	1.89 (0.89)
		MSE	13.2 (38.1)	11.1 (36.7)	5.88 (17.8)	6.38 (23.6)	6.61 (26.7)	8.57 (19.6)
		NZE	74.6 (6.69)	86.0 (5.04)	64.7 (9.07)	73.9 (9.86)	80.4 (8.19)	81.6 (3.97)
		CRPS	1.58 (1.52)	1.32 (1.53)	1.32 (0.74)	1.18 (1.08)	1.21 (1.20)	1.60 (0.77)
		CFR	0	0.30	0	0.01	0.01	0
	3	PPS	3.45 (2.37)	2.96 (2.49)	8.53 (14.7)	6.68 (10.5)	3.25 (0.38)	3.76 (1.95)
		MSE	51.9 (59.6)	39.6 (61.7)	36.6 (43.9)	28.6 (34.4)	33.0 (49.6)	25.8 (24.8)
		NZE	78.5 (7.20)	87.4 (4.83)	68.0 (8.77)	75.6 (8.36)	81.3 (7.17)	73.1 (4.06)
		CRPS	4.18 (1.93)	3.44 (2.07)	3.89 (1.57)	3.49 (1.53)	3.83 (1.92)	3.39 (1.12)
		CFR	0	0.27	0	0	0	0
100	1	PPS	2.25 (3.31)	1.18 (0.69)	0.74 (0.32)	0.72 (0.45)	0.82 (0.08)	0.86 (0.27)
		MSE	3.68 (2.12)	11.7 (5.42)	0.63 (0.58)	0.49 (0.86)	0.21 (0.55)	0.47 (0.20)
		NZE	59.3 (6.92)	85.8 (8.51)	63.8 (13.2)	76.0 (14.5)	86.2 (7.26)	67.8 (7.75)
		CRPS	3.34 (0.98)	2.69 (1.34)	0.69 (0.10)	0.66 (0.14)	0.62 (0.09)	0.68 (0.07)
		CFR	0.07	0.47	0	0.10	0.24	0
	3	PPS	2.75 (2.23)	2.91 (0.33)	2.06 (0.88)	2.11 (1.19)	1.79 (0.09)	1.92 (0.21)
		MSE	8.61 (7.40)	12.0 (24.0)	5.73 (4.74)	5.25 (8.33)	1.95 (2.44)	4.28 (1.53)
		NZE	69.2 (12.5)	86.9 (2.81)	62.9 (14.5)	74.5 (15.5)	85.3 (6.52)	68.6 (6.66)
		CRPS	3.41 (1.17)	2.80 (1.06)	2.12 (0.34)	2.05 (0.50)	1.95 (0.23)	2.04 (0.20)
		CFR	0.02	0.41	0	0.06	0.16	0
200	1	PPS	0.96 (0.12)	0.59 (0.07)	0.61 (0.05)	0.58 (0.06)	0.58 (0.03)	0.64 (0.08)
		MSE	1.55 (0.58)	0.19 (0.14)	0.27 (0.09)	0.16 (0.13)	0.08 (0.06)	0.25 (0.08)
		NZE	89.2 (1.07)	89.5 (0.71)	67.6 (8.93)	79.1 (9.54)	86.2 (4.18)	70.5 (6.98)
		CRPS	0.89 (0.10)	0.61 (0.04)	0.62 (0.03)	0.60 (0.04)	0.59 (0.03)	0.63 (0.04)
		CFR	0.52	0.62	0	0.02	0.04	0
	3	PPS	1.98 (0.11)	1.70 (0.09)	1.71 (0.08)	1.68 (0.09)	1.66 (0.05)	1.74 (0.11)
		MSE	10.3 (3.54)	2.01 (1.57)	2.37 (0.87)	1.66 (1.41)	0.85 (0.51)	2.24 (0.69)
		NZE	88.7 (1.41)	89.7 (0.59)	67.8 (9.12)	77.6 (10.1)	85.6 (4.14)	70.1 (7.34)
		CRPS	2.46 (0.26)	1.87 (0.16)	1.88 (0.13)	1.83 (0.14)	1.79 (0.12)	1.88 (0.13)
		CFR	0.39	0.75	0	0.03	0.07	0

estimation for λ_i will be referred to as the Bayesian adaptive pLasso (BapLasso).

We now consider a simulation study for the BpLasso and BapLasso. We use the posterior median from the Gibbs samples of the smoothing parameters as their point estimate, which appears to give better predictive performance. Using again the data sets generated from model (21), boxplots over replications for comparing the methods are shown in Fig. 4. It appears that the BapLasso works slightly better than the pLasso and the others. The Lasso performs poorly and adaptivity proves useful as both (cross-validation adaptive) pLasso and BapLasso appear to be superior to the BpLasso. In the following, however, we no longer consider the BapLasso in order to avoid heavy computation resulting from many MCMC runs over simulation replications.

A simulation study for logistic regression We simulate independent observations from Bernoulli distributions with probabilities of success

$$\mu_i = P(y_i = 1|\boldsymbol{x}_i, \boldsymbol{\beta}) = \frac{\exp(2 + \boldsymbol{x}_i'\boldsymbol{\beta})}{1 + \exp(2 + \boldsymbol{x}_i'\boldsymbol{\beta})}$$

where the design points $\boldsymbol{x}_i$ are generated from a normal distribution as in the previous example. We consider two cases: a small p case with $\boldsymbol{\beta} = (3, 1.5, 0.5, 0.5, 0, 0, 0, 0)^\top$ and a large p case with most of the β_j zero except the first four entries which are 3, 1.5, 0.5 and 0.5. We use the Gelman et

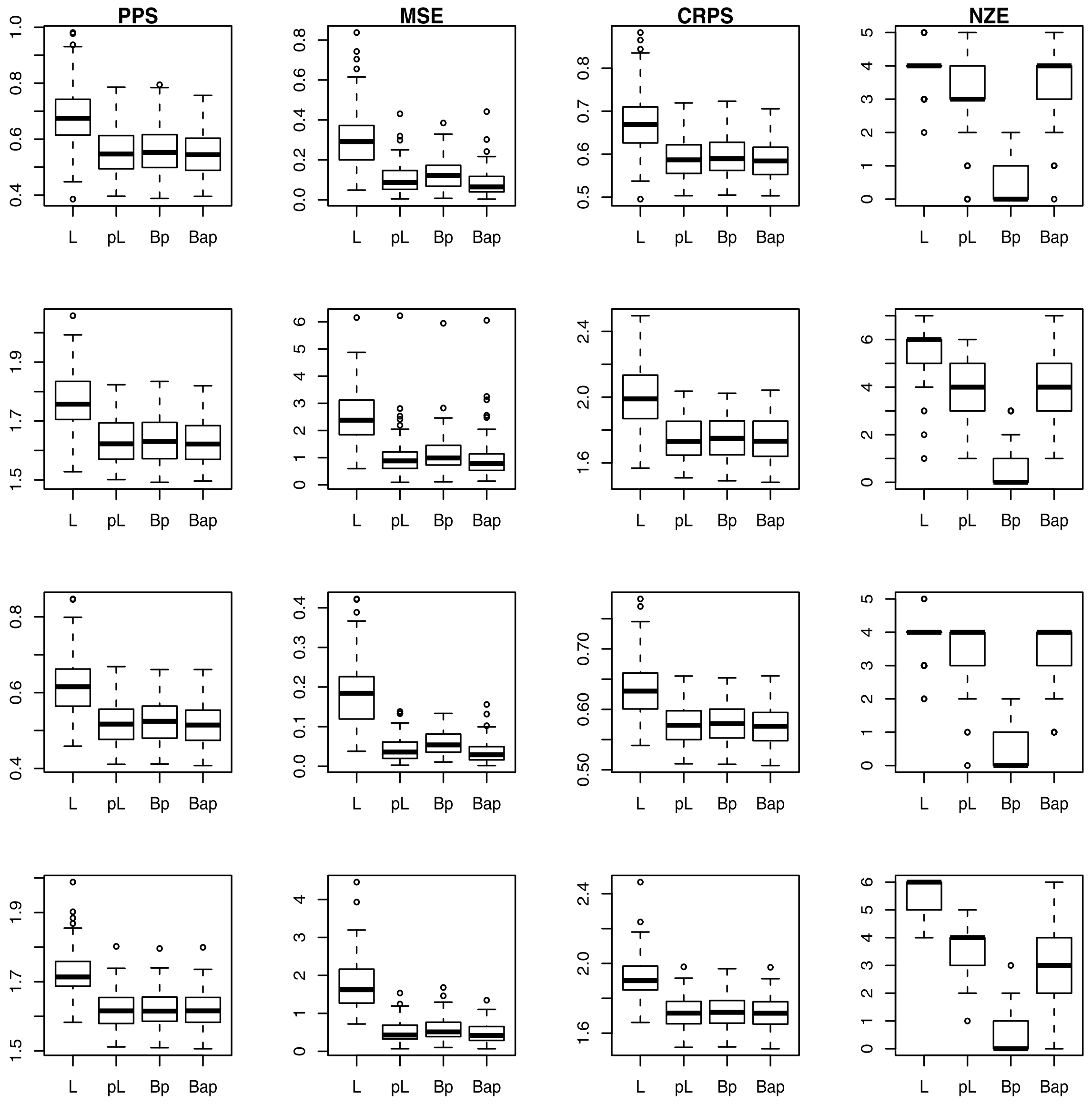

Fig. 4 Boxplots for comparing the Lasso (L), pLasso (pL), BpLasso (Bp) and BapLasso (Bap) in four cases: $n = 100, 200$ and $\sigma = 1, 3$. Data sets are generated from model (21) in the case with normal predictors and small p

al. prior and the plug-in method discussed earlier for estimating the predictions $\bar{y}_i^f$ (using MCMC would give a more accurate estimation but may be time consuming in simulation when performance measures are to be averaged over many replications). The simulation results are summarized in Tables 4 and 5 with various sample sizes. Boxplots for two cases are given in Figs. 5–6. As shown, both pLasso methods outperform the Lasso methods in terms of PPS, CRPS and MSE. Furthermore, the aLasso seems to be unstable and work poorly in the large p case when the number of observations is small. The pLasso with the regularization prior of Gelman et al. (2008) works surprisingly well in this example.

4.2 Real data examples

Example 1 (Linear regression—body fat percentage prediction) Percentage of body fat is one important measure of health, which can be accurately estimated by underwater weighing techniques (Bailey 1994). These techniques often require special equipment and are sometimes not convenient, thus fitting percent body fat to simple body measurements is a convenient way to predict body fat. Johnson (1996) introduced a dataset in which percent body fat and 13 simple body measurements (such as weight, height and abdomen circumference) are recorded for 252 men. After omitting observations 39 (because a weight value of 363.15 pounds is unusually large), 42 (because a height value of

Table 4 Simulation results for logistic regression: small p case

$n_T = n_P$	Measure	Lasso	aLasso	npLasso	pLasso
100	PPS	0.284 (0.048)	0.281 (0.076)	0.276 (0.048)	0.272 (0.052)
	MSE	3.515 (2.197)	5.203 (29.81)	2.455 (1.122)	1.953 (1.266)
	NZE	1.95 (1.47)	3.11 (1.08)	0.65 (1.41)	2.41 (1.40)
	CRPS	0.093 (0.021)	0.094 (0.024)	0.091 (0.021)	0.092 (0.023)
	CFR	0.14	0.42	0.02	0.24
200	PPS	0.278 (0.034)	0.278 (0.038)	0.275 (0.036)	0.274 (0.038)
	MSE	1.409 (0.765)	1.099 (0.816)	0.950 (0.484)	0.902 (0.524)
	NZE	1.638 (1.423)	3.01 (1.325)	0.404 (1.218)	2.132 (1.427)
	CRPS	0.087 (0.012)	0.088 (0.014)	0.086 (0.012)	0.086 (0.013)
	CFR	0.14	0.15	0.03	0.14
500	PPS	0.266 (0.021)	0.266 (0.022)	0.264 (0.021)	0.263 (0.022)
	MSE	0.605 (0.333)	0.477 (0.295)	0.389 (0.236)	0.351 (0.222)
	NZE	2.071 (1.249)	3.820 (1.122)	1.093 (0.975)	2.660 (1.207)
	CRPS	0.083 (0.007)	0.083 (0.008)	0.082 (0.007)	0.081 (0.007)
	CFR	0.13	0.34	0.02	0.19

Table 5 Simulation results for logistic regression: large p case

$n_T = n_P$	Measure	Lasso	aLasso	npLasso	pLasso
100	PPS	0.328 (0.043)	0.563 (0.211)	0.328 (0.044)	0.310 (0.050)
	MSE	4.567 (1.260)	17.53 (8.902)	4.378 (1.249)	2.636 (1.085)
	NZE	91.82 (5.271)	69.50 (6.228)	89.26 (7.676)	96.49 (2.157)
	CRPS	0.100 (0.016)	0.145 (0.043)	0.101 (0.017)	0.096 (0.019)
	CFR	0.01	0	0	0
500	PPS	0.278 (0.018)	0.693 (0.429)	0.279 (0.028)	0.269 (0.026)
	MSE	2.105 (0.595)	15.90 (18.10)	1.135 (0.432)	0.696 (0.367)
	NZE	89.62 (6.924)	45.82 (30.56)	60.82 (17.73)	82.18 (9.228)
	CRPS	0.085 (0.006)	0.124 (0.034)	0.087 (0.009)	0.084 (0.008)
	CFR	0.04	0	0	0
1000	PPS	0.263 (0.015)	0.260 (0.018)	0.273 (0.021)	0.255 (0.018)
	MSE	1.320 (0.405)	0.593 (0.279)	0.792 (0.407)	0.297 (0.202)
	NZE	89.58 (5.533)	96.18 (1.023)	38.9 (14.98)	83.58 (6.981)
	CRPS	0.081 (0.006)	0.081 (0.007)	0.085 (0.007)	0.079 (0.006)
	CFR	0.1	0.34	0	0

29.5 inches is unreasonable), and 182 (because the response value is 0), we obtain a dataset of size 249.

We are concerned with the problem of constructing a model that predicts the response from the covariates. Following Hoeting et al. (1999), we use a linear regression model. The primary goal is prediction accuracy for future observations; besides this, parsimony is another important objective, since a simple model is preferred for the sake of scientific insight into the x–y relationship.

Using the full dataset, the aLasso, pLasso and wpLasso estimates of $\boldsymbol{\beta}$ are given in Table 6. These methods simultaneously do parameter estimation and variable selection, because some of the estimated coefficients are exactly zero. Recall that the goals at which the methods aim are somewhat different: pLasso and wpLasso have a more explicit predictive motivation; besides, the wpLasso in some cases is somewhat more realistic in the sense that it allows the variances to vary in proportion to the predictive variance of the full model.

We now examine the predictive performance of these three procedures. To this end, we split the dataset into two parts: the first 125 observations are used as the training set D, the remaining observations are used as the prediction set

Table 6 Predicting percent body fat. The abbreviations "aL", "pL" and "wpL" stand for aLasso, pLasso and wpLasso, respectively

	Full data			Case I			Case II			Case III		
	aL	pL	wpL	aL	pL	wpL	aL	pL	wpL	aL	pL	wpL
C	−18.00	6.79	−0.18	−14.78	2.88	−0.28	−15.69	−2.95	−4.59	−23.31	−0.61	−3.87
X_1	0	0.06	0.04	0.02	0.09	0.08	0	0	0	0	0	0
X_2	0	0	0	0	0	0	0	0	0	0	0	0
X_3	−0.20	−0.29	−0.27	−0.26	−0.40	−0.39	0	−0.17	−0.14	0	−0.24	−0.22
X_4	0	−0.30	−0.11	0	−0.24	−0.17	0	0	0	0	−0.34	−0.25
X_5	0	−0.09	0	0	0	0	0	0	0	0	0	0
X_6	0.55	0.78	0.68	0.55	0.70	0.68	0.38	0.66	0.66	0.45	0.69	0.69
X_7	0	−0.09	0	0	−0.09	0	0	0	0	0	0	0
X_8	0	0.09	0	0	0.16	0.08	0	0	0	0	0	0
X_9	0	0	0	0	0.09	0	0	0	0	0	0	0
X_{10}	0	0.09	0	0	0.22	0.17	0	−0.39	−0.43	0	−0.04	0
X_{11}	0	0.13	0.04	0	0	0	0	0.10	0.10	0	0.20	0.20
X_{12}	0	0.19	0	0	0	0	0	0	0	0	0.19	0.07
X_{13}	0	−1.62	−1.31	0	−1.34	−1.20	0	−1.16	−1.15	0	−1.44	−1.35
PPS				1.946	1.933	1.933	2.112	1.913	1.902	2.075	1.965	1.951
CRPS				2.443	2.362	2.368	3.005	2.350	2.349	2.937	2.340	2.262

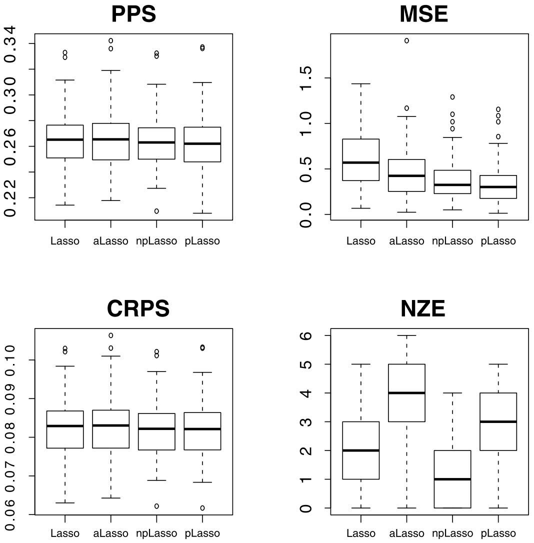

Fig. 5 Boxplots of performance measures over replications for comparing the methods in logistic regression: small p case with $n = 500$

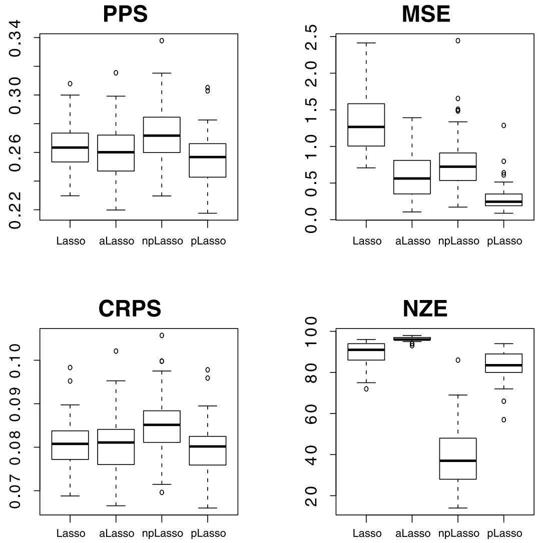

Fig. 6 Boxplots of performance measures over replications for comparing the methods in logistic regression: large p case with $n = 1000$

D^P. The aLasso, pLasso and wpLasso estimates and their PPS and CRPS are given in Table 6 (case I). As a second examination, the first 125 observations are used as the prediction set D^P, the remaining observations are used as the training set D. For the third examination, we randomly split the full dataset into two (roughly) equal parts which serve as the training and prediction sets. The coefficient estimates, PPS and CRPS are summarized in Table 6. As one may expect for predictively motivated methods, the variables selected by pLasso and wpLasso in general contain those selected by aLasso, i.e., the models selected by pLasso and wpLasso are bigger than the one selected by aLasso. In all cases, the pLasso and wpLasso show a better predictive performance over the aLasso. Indeed, the PPS of the aLasso, pLasso and wpLasso averaged over 50 such random partitions are 2.055, 1.998, 1.924, respectively and the averaged CRPS are 2.703, 2.385, 2.370, respectively. It seems that modelling the variances to vary in proportion to the predictive variance of the full model is appropriate in this example,

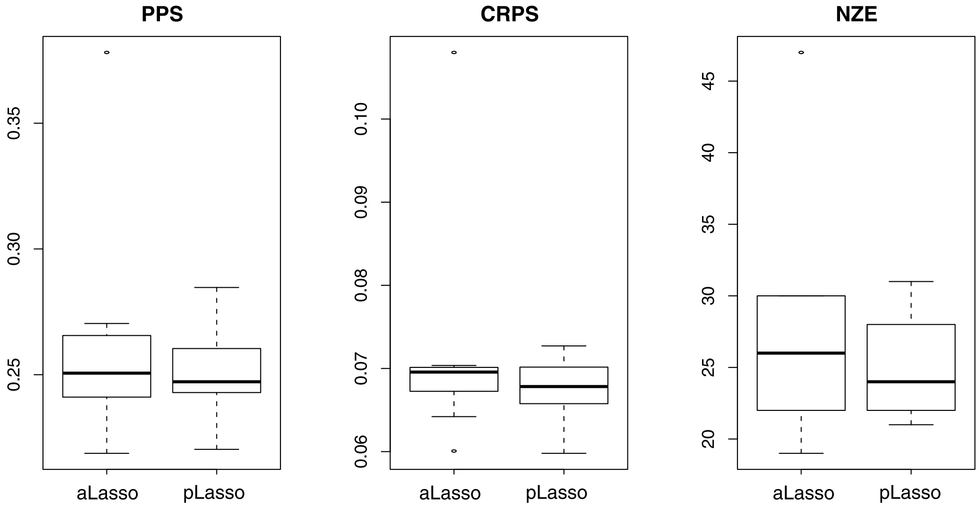

Fig. 7 Spambase data: boxplots of performance measures over 50 random partitions

because the wpLasso has a similar or better predictive performance compared with the pLasso.

Example 2 (Logistic regression—the spambase data) We consider in this example an application of the predictive Lasso in the logistic regression framework with many predictors and instances. We consider the spam email data set created by Mark Hopkins, Erik Reeber, George Forman and Jaap Suermondt at the Hewlett-Packard Labs. The data set consists of 4061 messages, each has been already classified as email or spam together with 57 attributes (predictors) which are relative frequencies of commonly occurring words. The goal is to design a spam filter that could filter out spam before clogging the users' mailboxes. Our goal as usual is to construct a parsimonious model with a good prediction accuracy.

With a large number of predictors and observations, using MCMC may be time consuming so that we use the plug in method discussed earlier. To access the performance of the aLasso and pLasso methods, we randomly split the data set into two parts (training set and prediction set) and record performance measures PPS, CRPS and NZE across 50 such random partitions. The averaged PPS, CRPS and NZE for the aLasso are 0.261, 0.072, 27.2 and for the pLasso are 0.251, 0.067, 25.1, respectively. Figure 7 gives side by side boxplots for these three measures over the partitions. As shown, the pLasso gives a better predictive performance overall while selecting roughly 2 predictors more than the aLasso.

5 Conclusion

The popular Lasso as a procedure for simultaneous variable selection and estimation has many attractive properties, and under certain conditions is able to identify the true model if it is assumed to exist. Our suggested pLasso has a more explicit predictive motivation which aims at selecting a useful model for prediction; besides, it enjoys the attractive properties of Lasso. A notable feature of pLasso is that we put no restriction on the predictive distribution $p(\Delta|D)$. Although we have considered $p(\Delta|D)$ as arising from a full model including all potential covariates, it can in fact arise from any model where a GLM approximation with variable selection is desired. The approximation can also be an appropriately local one in the covariate space through a judicious choice of the design points in the pLasso criterion, which need not correspond to the observed design points. In this paper we have motivated and developed the idea of pLasso only for GLMs. It is clear that this idea can be extended to other models rather than GLMs, and this is a topic for future research.

References

Aitchison, J.: Goodness of prediction fit. Biometrika **62**, 547–554 (1975)

Bailey, C.: Smart Exercise: Burning Fat, Getting Fit. Houghton-Mifflin, Boston (1994)

Brown, P.J., Vannucci, M., Fearn, T.: Bayes model averaging with selection of regressors. J. R. Stat. Soc. B **64**, 519–536 (2002)

Burnham, K.P., Anderson, D.: Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach. Springer, New York (2002)

Chen, M.H., Ibrahim, J.G.: Conjugate priors for generalized linear models. Stat. Sin. **13**, 461–476 (2003)

Dupuis, J.A., Robert, C.P.: Variable selection in qualitative models via an entropic explanatory power. J. Stat. Plan. Inference **111**, 77–94 (2003)

Geisser, S.: Discussion of "Sampling and Bayes' inference in scientific modelling and robustness" by G.E.P. Box. J. R. Stat. Soc., Ser. A **143**, 416–417 (1980)

Geisser, S.: Predictive Inference: An Introduction. Chapman & Hall, New York (1993)

Gelman, A., Jakulin, A., Grazia, P., Su, Y.-S.: A weakly informative default prior distribution for logistic and other regression models. Ann. Appl. Stat. **2**, 1360–1383 (2008)

Gneiting, T., Raftery, A.: Strictly proper scoring rules, prediction, and estimation. J. Am. Stat. Assoc. **102**, 359–378 (2007)

Good, I.J.: Rational decisions. J. R. Stat. Soc. B **14**, 107–114 (1952)

Hersbach, H.: Decomposition of the continuous ranked probability score for ensemble prediction systems. Weather Forecast. **15**, 559–570 (2000)

Hoeting, J.A., Madigan, D., Raftery, A.E., Volinsky, C.T.: Bayesian model averaging: a tutorial. Stat. Sci. **14**, 382–417 (1999)

Johnson, R.W.: Fitting percentage of body fat to simple body measurements. J. Stat. Educ. **4**, 1 (1996)

Leng, C., Tran, M.-N., Nott, D.J.: Bayesian adaptive Lasso. Technical Report (2010). arXiv:1009.2300v1

Lindley, D.V.: The choice of variables in multiple regression (with discussion). J. R. Stat. Soc. B **30**, 31–66 (1968)

Nott, D.J., Leng, C.: Bayesian projection approaches to variable selection in generalized linear models. Comput. Stat. Data Anal. **54**, 3227–3241 (2010)

O'Hagan, A., Forster, J.: The Advanced Theory of Statistics, Bayesian Inference, vol. 2B. Edward Arnold, London (2004)

Park, T., Casella, G.: The Bayesian Lasso. J. Am. Stat. Assoc. **103**, 681–686 (2008)

Raftery, A.E., Madigan, D., Hoeting, J.A.: Bayesian model averaging for linear regression models. J. Am. Stat. Assoc. **92**, 179–191 (1997)

Tibshirani, R.: Regression shrinkage and selection via the Lasso. J. R. Stat. Soc. B **58**, 267–288 (1996)

Tran, M.N.: A criterion for optimal predictive model selection. Commun. Stat., Theory Methods **40**, 893–906 (2011)

Vehtari, A., Lampinen, J.: Model selection via predictive explanatory power. Report B38, Laboratory of Computational Engineering, Helsinki University of Technology (2004)

Zellner, A.: On assessing prior distributions and Bayesian regression analysis with g-prior distributions. In: Bayesian Inference and Decision Techniques: Essays in Honour of Bruno De Finetti, pp. 233–243. North-Holland, Amsterdam (1986)

Zhao, P., Yu, B.: On model selection consistency of Lasso. J. Mach. Learn. Res. **7**, 2541–2563 (2006)

Zou, H.: The adaptive Lasso and its oracle properties. J. Am. Stat. Assoc. **101**, 1418–1429 (2006)

Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. J. R. Stat. Soc. B **67**, 301–320 (2005)