Can't ridge regression perform variable selection?

Yichao Wu

Abstract

Ridge regression was introduced to deal with the instability issue of the ordinary least squares estimate due to multicollinearity. It essentially penalizes the least squares loss by applying a ridge penalty on the regression coefficients. The ridge penalty shrinks the regression coefficient estimate towards zero, but not exactly zero. For this reason, the ridge regression has long been criticized of not being able to perform variable selection. In this paper, we proposed a new variable selection method based on an individually penalized ridge regression, a slightly generalized version of the ridge regression. An adaptive version is also provided. Our new methods are shown to perform competitively based on simulation and a real data example. Supplementary materials for some simulation results are available online.

1 Introduction

Regression analysis has been a very important technique in statistics. Its target is to study how a response variable depends on one or more predictor variables. The most elementary form of regression is ordinary least squares regression. It has wide applicability in all kinds of application areas ranging from biomedical science to engineering to financial industry.

On top of the ordinary least squares regression, Hoerl and Kennard (1970) proposed ridge regression, which penalizes the least squares loss by a L_2 penalty on the regression coefficients. The L_2 penalty was added mainly to deal with the instability issue of the ordinary least squares estimate due to multicollinearity. See Hastie (2020) for a review on ridge regression and related developments. The L_2 penalty shrinks regression coefficients estimate towards zero, but not exactly zero. As a result, ridge regression has long been criticized of not being able to perform variable selection.

The advance of modern technologies for data collection and storage has made it possible to collect many potential predictor variables while studying one response variable of interest. Such advance made it highly desirable to conduct variable selection, and has consequently motivated the reasearch area of variable selection. The fundmental goal of variable selection is to identify important predictor variables that can be used to explain

how the response variable varies. There has been a very rich literature on variable selection, as echoed by many review papers published in different journals and different research areas such as Fan and Lv (2010); Anzanello and Fogliatto (2014); Barcella et al. (2017); Desboulets (2018); Heinze et al. (2018); Kirpich et al. (2018); Talbot (2019).

The simplest type of regression is linear regression, for which many variable selection methods have been developed. See Fan and Lv (2010) for a selective overview of variable selection methods for linear regression. In particular, the least absolute shrinkage and selection operator (LASSO, Tibshirani 1996) has been a very successful variable selection method and is widely used. It essentially achieves variable selection by penalizing the least squares loss by a L_1 penalty on the regression coefficients.

Tibshirani (1996) provided a geometric interpretation why the LASSO is capable of performing varible selection while the ridge regression is not. The fundmental reason is that the L_1 function is singular at the origin since the left and right derivatives at 0 are not equal, while the L_2 function is smooth. This singularity at the origin is the key to the success of all LASSO-type variable selection methods. Examples are Tibshirani (1996); Fan and Li (2001); Zou (2006); Zhang and Lu (2007); Zhang (2010) among many others.

In this paper, we will work closely with the ridge regression and propose a new variable selection method based on it. We consider a more general version of the classical ridge regression: individually penalized ridge regression. Instead of using a same ridge regularization paramter for all regression coefficent components as done in the ridge regression, the individually penalized ridge regression uses different ridge regularization parameters for different regression coefficient components. Intuitively speaking, the individual ridge regularization parameter corresponding to a small true regression coefficient should be set to be large to apply more shrinkage to its corresponding estimate towards zero, and vice versa. In the extreme, if an infinity individual ridge regularization parameter is used, the corresponding ridge estimate will be exactly zero. In this way, the job of variable selection boils down to the identification of regression coefficient components for which we should use an infinity individual ridge regularization parameter. The main objective of the current paper is to devise a new method to achieve this goal. There are some ealier tries along this line. Examples are Joseph and Delaney (2008) and Wipf and Nagarajan (2008) based on Bayesian approaches. Shao and Deng (2012) proposed another variable selection method based on ridge regression via thresholding. Frommlet and Nuel (2016) proposed an adaptive ridge procedure for L_0 regularization.

The rest of the paper is organized as follows. Section 2 reviews the classical ridge regression. Our new method, ridge selection operator, is presented in Section 3, with its

adaptive version given in Section 4. Section 5 compares our new methods with the LASSO and adaptive LASSO (Zou 2006; Zhang and Lu 2007) together with other methods using simulation studies. A real data example is given in Section 6. We conclude with a brief discussion in Section 7.

2 Classical ridge regression

We consider the most elementary linear regression model

$$Y = \beta_0 + \sum_{j=1}^{p} X_j \beta_j + \epsilon$$

with p-dimension predictors $X_1, X_2, \ldots, X_p$ and random error ϵ of mean zero and a finite variance. The interest is to estimate the unknown regression coefficients β_0 and $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_p)^T$ based on a random sample $\{(\mathbf{x}_i, y_i) : i = 1, 2, \ldots, n\}$ from this linear regression model, where $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$. We denote $\mathbf{y} = (y_1, y_2, \ldots, y_n)^T$, and $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)^T$. We assume without loss of generality that the predictors have been standardized such that

$$\mathbf{X}^T \mathbf{1}_{n\times 1} = \mathbf{0}_{p\times 1} \text{ and } \sum_{i=1}^{n} x_{ij}^2 = n,\ j = 1, 2, \ldots, p, \tag{1}$$

where $\mathbf{1}_{n\times 1}$ denotes a $n \times 1$ vector of ones and $\mathbf{0}_{p\times 1}$ a $p \times 1$ vector of zeros. Here for method development, it is not required to rescale each predictor to have variance one. But as common in practice, such a rescaling can also be applied to each predictor beforehand.

Hoerl and Kennard (1970) proposed to penalize the least squares loss by a L_2 penalty (also called the ridge penalty). The classical ridge regression estimates of the regression coefficients β_0 and $\boldsymbol{\beta}$ are defined as the optimizer of

$$\min_{\beta_0, \boldsymbol{\beta}} \ \frac{1}{2} \sum_{i=1}^{n} (y_i - \beta_0 - \mathbf{x}_i^T \boldsymbol{\beta})^2 + \frac{\nu}{2} \sum_{j=1}^{p} \beta_j^2 \tag{2}$$

with ridge regularization parameter $\nu \geq 0$. It is obvious that the ridge regression estimate simplifies to the ordinary least squares estimate when $\nu = 0$. The ridge regression has enjoyed great success and been widely used since its inception.

3 Variable selection via individually penalized ridge regression

Our interest is sparse regression with some components of the true regression coefficients being exactly zero. Let $\mathcal{A} = \{j : \beta_j \neq 0\}$ be the set of important predictors.

3.1 Individually penalized ridge regression

Towards variable selection, we consider a slightly generalized version of the classical ridge regression (2), namely individually penalized ridge regression

$$\min_{\beta_0,\boldsymbol{\beta}} \frac{1}{2}\sum_{i=1}^{n}(y_i-\beta_0-\mathbf{x}_i^T\boldsymbol{\beta})^2+\frac{1}{2}\sum_{j=1}^{p}\nu_j\beta_j^2 \tag{3}$$

with $\nu_j \geq 0$ for $j = 1, 2, \ldots, p$. The only difference is that different ridge regularization parameters may be used here for different regression coefficient components while a same ridge regularization parameter is used for all regression coefficient components in (2). By incorporating the intercept term, we denote the augmented data by $\widetilde{\mathbf{x}}_i = (1, \mathbf{x}_i^T)^T$ and $\widetilde{\mathbf{X}} = (\widetilde{\mathbf{x}}_1, \widetilde{\mathbf{x}}_2, \ldots, \widetilde{\mathbf{x}}_n)^T \equiv (\mathbf{1}_{n\times 1}, \mathbf{X})$. The solution of (3) is given by

$$(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}^T)^T = \left[\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}} + \mathrm{diag}((0, \boldsymbol{\nu}^T)^T)\right]^{-1}\widetilde{\mathbf{X}}^T\mathbf{y},$$

where $\boldsymbol{\nu} = (\nu_1, \nu_2, \ldots, \nu_p)^T$ and $\mathrm{diag}(\boldsymbol{\nu})$ denotes a diagonal matrix with elements of $\boldsymbol{\nu}$ sitting on the diagonal.

3.2 Ridge selection operator

It is well known that the ridge penalty shrinks the regression coefficient estimates towards zero due to the L_2 penalty. Intuitively, a large ridge penalty should be used if the corresponding true regression coefficient is zero or of a small magnitude. More precisely, a large ridge regularization parameter ν_j should be used if the absolute value of the corresponding true regression coefficient β_j is small, and vice versa. In extreme, if $\nu_j = \infty$ is used in (3), the corresponding optimizer $\widehat{\beta}_j$ will be exactly zero. If we know *a priori* which components of the true regression coefficients are zero, we can set the corresponding ridge regularization parameters to be infinity in (3) to achieve variable selection. The challenge is that we do not have this *a priori* information in hand. In fact, if we know this information beforehand, there is no need to perform variable selection any more.

Motivated by Stefanski et al. (2014), we propose the following idea to let the data speak for themselves and tell us which components favor an infinity ridge regularization parameter in (3), achieving variable selection.

We reparametrize $\lambda_j = 1/\nu_j$ and $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_p)^T$, and introduce notation $\boldsymbol{\lambda}^{-1} = (1/\lambda_1, 1/\lambda_2, \ldots, 1/\lambda_p)^T$. As a result, $\boldsymbol{\nu} = \boldsymbol{\lambda}^{-1}$. With these notations, the solution of (3) is given by

$$(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}^T)^T \equiv (\widehat{\beta}_0(\boldsymbol{\lambda}), \widehat{\boldsymbol{\beta}}(\boldsymbol{\lambda})^T)^T = \left[\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}} + \mathrm{diag}((0, (\boldsymbol{\lambda}^{-1})^T)^T)\right]^{-1}\widetilde{\mathbf{X}}^T\mathbf{y} \tag{4}$$

and the corresponding hat matrix is

$$\mathbf{H}(\boldsymbol{\lambda}) = \widetilde{\mathbf{X}} \left[\widetilde{\mathbf{X}}^T \widetilde{\mathbf{X}} + \mathrm{diag}((0, (\boldsymbol{\lambda}^{-1})^T)^T) \right]^{-1} \widetilde{\mathbf{X}}^T. \tag{5}$$

We propose a new variable selection method by solving

$$\begin{aligned} \min_{\boldsymbol{\lambda}} \quad & \langle \mathbf{y} - \mathbf{H}(\boldsymbol{\lambda})\mathbf{y}, \mathbf{y} - \mathbf{H}(\boldsymbol{\lambda})\mathbf{y} \rangle & (6) \\ \text{subject to} \quad & \lambda_j \geq 0, j = 1, 2, \ldots, p; & (7) \\ & \sum_{j=1}^{p} \lambda_j \leq \tau & (8) \end{aligned}$$

for some regularization parameter $\tau \geq 0$, where $\langle \cdot, \cdot \rangle$ denotes the standard inner product operator. The nonnegativity constraint in (7) is easy to understand since λ_j is the reciprocal of the nonnegative ridge regularization parameter ν_j. The additional constraint (8) enforces to apply certain amount of ridge regularization measured by the harmonic mean of the ridge regularization parameters:

$$\left(\frac{\frac{1}{\nu_1} + \frac{1}{\nu_2} + \cdots + \frac{1}{\nu_p}}{p} \right)^{-1} \geq \frac{p}{\tau}.$$

Denote the optimizer by $\widehat{\boldsymbol{\lambda}} \triangleq (\widehat{\lambda}_1, \widehat{\lambda}_2, \ldots, \widehat{\lambda}_p)^T \equiv (\widehat{\lambda}_1(\tau), \widehat{\lambda}_2(\tau), \ldots, \widehat{\lambda}_p(\tau))^T \triangleq \widehat{\boldsymbol{\lambda}}(\tau)$. For an appropriately tuned τ, some components of the corresponding optimizer will be exactly zero. Then an estimate of the set of important predictors is given by $\widehat{\mathcal{A}} = \{j : \widehat{\lambda}_j > 0\}$ as explained next. We name the proposed new method ridge selection operator (RSO).

3.3 Some implementation issues

Within the feasible domain specified by constraints (7) and (8), some components of $\boldsymbol{\lambda}$ could be exactly zero. In this case, the second term inside the hat matrix $\mathbf{H}(\boldsymbol{\lambda})$ cannot be evaluated exactly in this way since it may have the potential issue of inverting zero in $\boldsymbol{\lambda}^{-1}$. This potential issue can be perfectly avoided by noting that

$$\begin{aligned} \left[\widetilde{\mathbf{X}}^T \widetilde{\mathbf{X}} + \mathrm{diag}((0, (\boldsymbol{\lambda}^{-1})^T)^T) \right]^{-1} &= \left[\begin{pmatrix} n & \mathbf{0}^T \\ \mathbf{0} & \mathbf{X}^T\mathbf{X} \end{pmatrix} + \begin{pmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & \mathrm{diag}(\boldsymbol{\lambda}^{-1}) \end{pmatrix} \right]^{-1} \\ &= \begin{pmatrix} 1/n & \mathbf{0}^T \\ \mathbf{0} & \left[\mathbf{X}^T\mathbf{X} + \mathrm{diag}(\boldsymbol{\lambda}^{-1}) \right]^{-1} \end{pmatrix} \end{aligned} \tag{9}$$

due to (1), and further that

$$\left[\mathbf{X}^T\mathbf{X} + \mathrm{diag}(\boldsymbol{\lambda}^{-1})\right]^{-1} = \mathrm{diag}(\sqrt{\boldsymbol{\lambda}})\left[\mathrm{diag}(\sqrt{\boldsymbol{\lambda}})\mathbf{X}^T\mathbf{X}\mathrm{diag}(\sqrt{\boldsymbol{\lambda}}) + \mathbf{I}\right]^{-1}\mathrm{diag}(\sqrt{\boldsymbol{\lambda}}), \quad (10)$$

where $\sqrt{\boldsymbol{\lambda}} = (\sqrt{\lambda_1}, \sqrt{\lambda_2}, \ldots, \sqrt{\lambda_p})^T$ and $\mathbf{I}$ denotes the identity matrix of an appropriate size depending on the context. It retains the symmetric property of the hat matrix, and can be used in the implementation of the objective function in (6) to avoid any numerical issue of dividing by zero in $\boldsymbol{\lambda}^{-1}$.

Note that if $\widehat{\lambda}_j = 0$, the corresponding estimate of β_j is also exactly zero, namely $\widehat{\beta}_j(\widehat{\boldsymbol{\lambda}}) = 0$, since

$$\begin{aligned}\widehat{\boldsymbol{\beta}}(\boldsymbol{\lambda}) &= \left[\mathbf{X}^T\mathbf{X} + \mathrm{diag}(\boldsymbol{\lambda}^{-1})\right]^{-1}\mathbf{X}^T\mathbf{y} \\ &= \mathrm{diag}(\sqrt{\boldsymbol{\lambda}})\left[\mathrm{diag}(\sqrt{\boldsymbol{\lambda}})\mathbf{X}^T\mathbf{X}\mathrm{diag}(\sqrt{\boldsymbol{\lambda}}) + \mathbf{I}\right]^{-1}\mathrm{diag}(\sqrt{\boldsymbol{\lambda}})\mathbf{X}^T\mathbf{y}\end{aligned}$$

due to (9) and (10). It provides justification for $\widehat{\mathcal{A}} = \{j : \widehat{\lambda}_j > 0\}$ to be used as an estimate of the set of important predictors.

It looks like that we need to invert a $p \times p$ matrix in (10). By noting that if $\boldsymbol{\lambda}$ is sparse with some components being zero, $\mathrm{diag}(\sqrt{\boldsymbol{\lambda}})\mathbf{X}^T\mathbf{X}\mathrm{diag}(\sqrt{\boldsymbol{\lambda}}) + \mathbf{I}$ can be transformed to a block diagonal matrix after applying an appropriate row permutation and the corresponding column permutation. Inverting this block matrix essentially needs only to invert a matrix of size $\#(\boldsymbol{\lambda}) \times \#(\boldsymbol{\lambda})$, where $\#(\boldsymbol{\lambda})$ denotes the number of nonzero components in $\boldsymbol{\lambda}$.

It can be shown that the equality "=" in (8) will always be attained at the optimal solution. Consequently it is equivalent to replace constraint (8) with an equality constraint $\sum_{j=1}^p \lambda_j = \tau$. The corresponding optimization problem with $\sum_{j=1}^p \lambda_j = \tau$

$$\begin{aligned}\min_{\boldsymbol{\lambda}} \quad & \langle \mathbf{y} - \mathbf{H}(\boldsymbol{\lambda})\mathbf{y}, \mathbf{y} - \mathbf{H}(\boldsymbol{\lambda})\mathbf{y}\rangle & (11)\\ \text{subject to} \quad & \lambda_j \geq 0, j = 1, 2, \ldots, p; & (12)\\ & \sum_{j=1}^p \lambda_j = \tau & (13)\end{aligned}$$

can be efficiently solved by using the modified coordinate descent algorithm introduced in Stefanski et al. (2014). More explicitly, suppose that the current solution is $\widehat{\boldsymbol{\lambda}}^{(c)} = (\widehat{\lambda}_1^{(c)}, \widehat{\lambda}_2^{(c)}, \ldots, \widehat{\lambda}_p^{(c)})^T$ and we are updating the jth component. Let $\mathbf{e}_j$ be the jth standard basis for the p dimensional Euclidean space. Namely $\mathbf{e}_j$ is a vector of length p with its jth element being one and all other elements zero. Denote $\widehat{\boldsymbol{\lambda}}_{-j}^{(c)} = (\widehat{\boldsymbol{\lambda}} - \widehat{\lambda}_j^{(c)}\mathbf{e}_j)/\sum_{j' \neq j}\widehat{\lambda}_{j'}^{(c)}$. Then for any $\gamma \in [0, \tau]$, $\gamma\widehat{\boldsymbol{\lambda}}_{-j}^{(c)} + (\tau - \gamma)\mathbf{e}_j$ satisfies the nonnegativity constraint (12) and

the sum-to-τ constraint (13). We update the solution by $\widehat{\gamma}\widehat{\boldsymbol{\lambda}}^{(c)}_{-j} + (\tau - \widehat{\gamma})\mathbf{e}_j$, where $\widehat{\gamma}$ is given by

$$\widehat{\gamma} = \text{argmin}_\gamma \left\langle \mathbf{y} - \mathbf{H}(\gamma\widehat{\boldsymbol{\lambda}}^{(c)}_{-j} + (\tau-\gamma)\mathbf{e}_j)\mathbf{y}, \mathbf{y} - \mathbf{H}(\gamma\widehat{\boldsymbol{\lambda}}^{(c)}_{-j} + (\tau-\gamma)\mathbf{e}_j)\mathbf{y} \right\rangle.$$

We repeat cycling through $j = 1, 2, \ldots, p$ until convergence.

3.4 How does it work in the orthonormal design case?

Note that the hat matrix $\mathbf{H}(\boldsymbol{\lambda})$ can be simplified as

$$\begin{aligned}
\mathbf{H}(\boldsymbol{\lambda}) &= \widetilde{\mathbf{X}}\left[\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}} + \text{diag}((0, (\boldsymbol{\lambda}^{-1})^T)^T)\right]^{-1}\widetilde{\mathbf{X}}^T \\
&= \widetilde{\mathbf{X}}\begin{pmatrix} 1/n & \mathbf{0}^T \\ \mathbf{0} & \left[\mathbf{X}^T\mathbf{X} + \text{diag}(\boldsymbol{\lambda}^{-1})\right]^{-1} \end{pmatrix}\widetilde{\mathbf{X}}^T \\
&= \frac{1}{n}\mathbf{1}_{n\times n} + \mathbf{X}\left[\mathbf{X}^T\mathbf{X} + \text{diag}(\boldsymbol{\lambda}^{-1})\right]^{-1}\mathbf{X}^T
\end{aligned}$$

by using (9) and consequently

$$\mathbf{H}(\boldsymbol{\lambda})\mathbf{H}(\boldsymbol{\lambda}) = \frac{1}{n}\mathbf{J} + \mathbf{X}\left[\mathbf{X}^T\mathbf{X} + \text{diag}(\boldsymbol{\lambda}^{-1})\right]^{-1}\mathbf{X}^T\mathbf{X}\left[\mathbf{X}^T\mathbf{X} + \text{diag}(\boldsymbol{\lambda}^{-1})\right]^{-1}\mathbf{X}^T$$

since $\mathbf{X}$ is column centered and satisfies (1). Here $\mathbf{J}$ denotes a $n \times n$ matrix of ones.

With the above simplification, the objective function of (6) simplifies to

$$\begin{aligned}
& \langle \mathbf{y} - \mathbf{H}(\boldsymbol{\lambda})\mathbf{y}, \mathbf{y} - \mathbf{H}(\boldsymbol{\lambda})\mathbf{y}\rangle \\
= \;& \mathbf{y}^T\mathbf{y} - 2\mathbf{y}^T\mathbf{H}(\boldsymbol{\lambda})\mathbf{y} + \mathbf{y}^T\mathbf{H}(\boldsymbol{\lambda})\mathbf{H}(\boldsymbol{\lambda})\mathbf{y} \\
= \;& \mathbf{y}^T\mathbf{y} - \frac{1}{n}\mathbf{y}^T\mathbf{J}\mathbf{y} - 2\mathbf{y}^T\mathbf{X}\left[\mathbf{X}^T\mathbf{X} + \text{diag}(\boldsymbol{\lambda}^{-1})\right]^{-1}\mathbf{X}^T\mathbf{y} \\
& + \mathbf{y}^T\mathbf{X}\left[\mathbf{X}^T\mathbf{X} + \text{diag}(\boldsymbol{\lambda}^{-1})\right]^{-1}\mathbf{X}^T\mathbf{X}\left[\mathbf{X}^T\mathbf{X} + \text{diag}(\boldsymbol{\lambda}^{-1})\right]^{-1}\mathbf{X}^T\mathbf{y} \\
= \;& \mathbf{y}_c^T\mathbf{y}_c - 2\mathbf{y}_c^T\mathbf{X}\left[\mathbf{X}^T\mathbf{X} + \text{diag}(\boldsymbol{\lambda}^{-1})\right]^{-1}\mathbf{X}^T\mathbf{y}_c \\
& + \mathbf{y}_c^T\mathbf{X}\left[\mathbf{X}^T\mathbf{X} + \text{diag}(\boldsymbol{\lambda}^{-1})\right]^{-1}\mathbf{X}^T\mathbf{X}\left[\mathbf{X}^T\mathbf{X} + \text{diag}(\boldsymbol{\lambda}^{-1})\right]^{-1}\mathbf{X}^T\mathbf{y}_c, \qquad (14)
\end{aligned}$$

where $\mathbf{y}_c = \mathbf{y} - \frac{1}{n}\mathbf{J}\mathbf{y}$ denotes the centered $\mathbf{y}$.

To gain further insights, we consider the special orthonormal design case with $\mathbf{X}^T\mathbf{X} =$

I. In this case, the right hand side of (14) simplifies to

$$\begin{aligned}
& \mathbf{y}_c^T\mathbf{y}_c - 2\mathbf{y}_c^T\mathbf{X}\left[\mathbf{I}+\mathrm{diag}(\boldsymbol{\lambda}^{-1})\right]^{-1}\mathbf{X}^T\mathbf{y}_c + \mathbf{y}_c^T\mathbf{X}\left[\mathbf{I}+\mathrm{diag}(\boldsymbol{\lambda}^{-1})\right]^{-1}\mathbf{I}\left[\mathbf{I}+\mathrm{diag}(\boldsymbol{\lambda}^{-1})\right]^{-1}\mathbf{X}^T\mathbf{y}_c \\
= \;& \mathbf{y}_c^T\mathbf{y}_c - 2\sum_{j=1}^{p}\frac{\lambda_j}{1+\lambda_j}\widetilde{\beta}_j^2 + \sum_{j=1}^{p}\left(\frac{\lambda_j}{1+\lambda_j}\right)^2\widetilde{\beta}_j^2 \\
= \;& \mathbf{y}_c^T\mathbf{y}_c + \sum_{j=1}^{p}(1-\frac{\lambda_j}{1+\lambda_j})^2\widetilde{\beta}_j^2 - \sum_{j=1}^{p}\widetilde{\beta}_j^2 \\
= \;& \sum_{j=1}^{p}(1-\frac{\lambda_j}{1+\lambda_j})^2\widetilde{\beta}_j^2 + \text{constant}, \qquad (15)
\end{aligned}$$

where $\widetilde{\beta}_j$ denotes the jth component of the corresponding ordinary least squares estimate $\widetilde{\boldsymbol{\beta}} = \mathbf{X}^T\mathbf{y}_c$ in the orthonormal design case and the constant term in (15) does not depend on $\boldsymbol{\lambda}$.

With the above simplification for the special orthonormal design case, the optimization problem (6) simplifies to

$$\begin{aligned}
\min_{\boldsymbol{\lambda}} \quad & \sum_{j=1}^{p}\frac{1}{(1+\lambda_j)^2}\widetilde{\beta}_j^2 \qquad (16) \\
\text{subject to} \quad & \lambda_j \geq 0, j = 1, 2, \ldots, p; \\
& \sum_{j=1}^{p}\lambda_j \leq \tau.
\end{aligned}$$

The objective function of (16) is as simple as an additive function of λ_j's.

3.5 A toy example for the orthonormal design case

Now we use a toy example to illustrate how the solution path looks like for the orthonormal design case (16) explained above. Take $\widetilde{\boldsymbol{\beta}} = (3, 1.5, .1, .08, 2, .15, .2, .05)^T$ for example. The big values of $\widetilde{\beta}_1$, $\widetilde{\beta}_2$, and $\widetilde{\beta}_5$ indicate important predictors and small values of other components indicate unimportant predictors, trying to mimic the simulation example to be presented in Section 5.

In the top-left panel of Figure 1, we plot the solution path of the optimizer $\widehat{\boldsymbol{\lambda}}$ of (16) as a function of τ. It clearly shows that it does generate sparsity along the solution path. In particular, $\widehat{\lambda}_1$, $\widehat{\lambda}_2$, and $\widehat{\lambda}_5$ corresponding to "important" predictors quickly changed from zero to nonzero as τ increases from zero. In addition, the optimizer $\widehat{\boldsymbol{\lambda}}$ increases gradually as the regularization parameter τ increases. This gradual increasing pattern is similar to that of the LASSO solution path.

The bottom-left plot of Figure 1 presents the corresponding solution path of the individually penalized ridge regression estimate $\widehat{\boldsymbol{\beta}}$ corresponding to the optimizer $\widehat{\boldsymbol{\lambda}}$ as a function of τ. Note that the sparsity of $\widehat{\boldsymbol{\lambda}}$ is maintained in the corresponding individually penalized ridge regression estimate, leading to successful variable selection as desired. It is remarkable to note that the magnitude of any nonzero components of $\widehat{\boldsymbol{\beta}}$ increases rapidly to near its corresponding maximum once moving away from zero. This rapid early-stage change makes the RSO estimate less biased, and is much different from the rather slow change of $\widehat{\boldsymbol{\lambda}}$. This difference is due to the relationship $\widehat{\beta}_j = \frac{\widehat{\lambda}_j}{1+\widehat{\lambda}_j}\widetilde{\beta}_j$ for the orthonormal design case being considered. Note that $\frac{\partial}{\partial \widehat{\lambda}_j}\left(\frac{\widehat{\lambda}_j}{1+\widehat{\lambda}_j}\widetilde{\beta}_j\right) = \frac{1}{(1+\widehat{\lambda}_j)^2}\widetilde{\beta}_j$ and $\frac{1}{(1+\widehat{\lambda}_j)^2}$ is monotonically decreasing over $\widehat{\lambda}_j \in [0,\infty)$. This is due to the reparametrization $\boldsymbol{\lambda} = \boldsymbol{\nu}^{-1}$.

Note that the above simplification was made possible only with the assumption of orthonormal design. Of cause, real-world applications are complicated and orthonormal design is most likely an unrealistic assumption. Yet the above discussion sheds some important insights on how the proposed RSO works.

4 Adaptive variable selection via adaptively weighted individually penalized ridge regression

Note that different regression parameter components are treated equally in the aforementioned RSO optimization problem (6) based on the individually penalized ridge regression (3). Yet we may treat different regression coefficient components differently by incorporating some prior information as done in the adaptive LASSO (Zou 2006; Zhang and Lu 2007). By using adaptive weights, the adaptive LASSO has been shown to enjoy superior properties than the original LASSO (Tibshirani 1996).

4.1 Adaptively weighted individually penalized ridge regression

More precisely, we can consider the following adaptively weighted individually penalized ridge regression

$$\min_{\beta_0,\boldsymbol{\beta}} \ \frac{1}{2}\sum_{i=1}^{n}(y_i - \beta_0 - \mathbf{x}_i^T\boldsymbol{\beta})^2 + \frac{1}{2}\sum_{j=1}^{p} w_j\nu_j\beta_j^2 \tag{17}$$

with regularization parameter $\nu_j \geq 0$ and *prespecified* weight $w_j \geq 0$ for $j = 1, 2, \ldots, p$. The weights w_j's are prespecified in such a way to incorporate our prior knowledge on the relative importance of different predictors with more (resp. less) important predictors receiving smaller (resp. larger) weights. Denote $\mathbf{w} = (w_1, w_2, \ldots, w_p)^T$. Then the

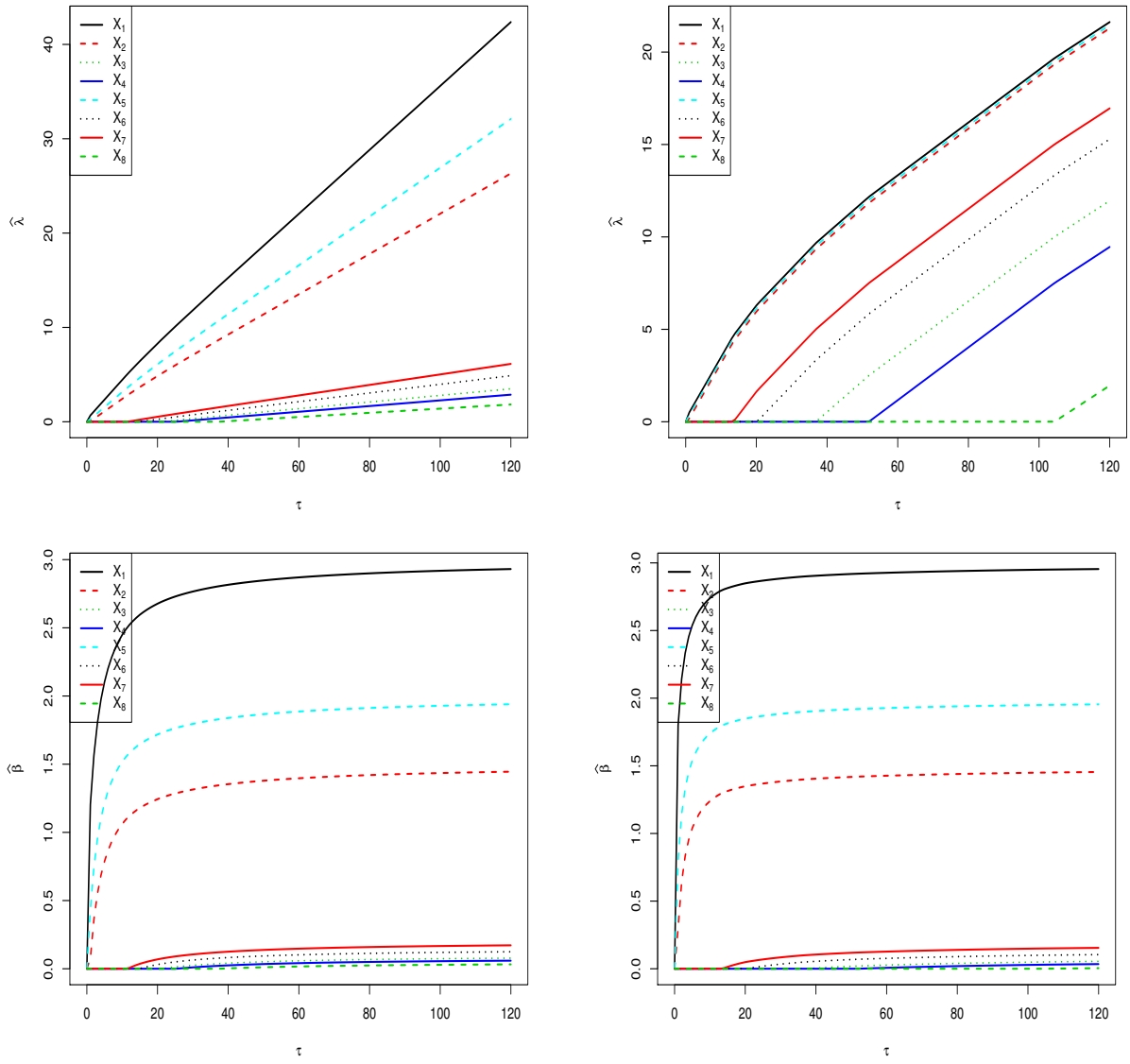

Figure 1: Solution paths of a toy example for the orthonormal design case: RSO on the left and aRSO on the right; $\widehat{\boldsymbol{\lambda}}$ path in the top row and $\widehat{\boldsymbol{\beta}}$ path in the bottom row.

corresponding optimizer of $(\beta_0, \boldsymbol{\beta}^T)^T$ is given by

$$\left[\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}} + \begin{pmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & \mathrm{diag}(\mathbf{w})\mathrm{diag}(\boldsymbol{\nu}) \end{pmatrix}\right]^{-1} \widetilde{\mathbf{X}}^T\mathbf{y}. \tag{18}$$

With similar reparametrization $\boldsymbol{\nu} = \boldsymbol{\lambda}^{-1}$, we denote the corresponding hat matrix for the above adaptively weighted individually penalized regression by

$$\mathbf{H}_{\mathbf{w}}(\boldsymbol{\lambda}) = \widetilde{\mathbf{X}} \left[\widetilde{\mathbf{X}}^T \widetilde{\mathbf{X}} + \begin{pmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & \mathrm{diag}(\mathbf{w})\mathrm{diag}(\boldsymbol{\lambda}^{-1}) \end{pmatrix} \right]^{-1} \widetilde{\mathbf{X}}^T. \tag{19}$$

4.2 Adaptive ridge selection operator

The corresponding adaptive variable selection is to solve

$$\begin{aligned} \min_{\boldsymbol{\lambda}} \quad & \langle \mathbf{y} - \mathbf{H}_{\mathbf{w}}(\boldsymbol{\lambda})\mathbf{y}, \mathbf{y} - \mathbf{H}_{\mathbf{w}}(\boldsymbol{\lambda})\mathbf{y} \rangle \\ \text{s.t.} \quad & \lambda_j \geq 0, j = 1, 2, \ldots, p; \\ & \sum_{j=1}^{p} \lambda_j \leq \tau \end{aligned} \tag{20}$$

for appropriately tuned $\tau \geq 0$. The aforementioned potential issue of inverting zero can happen in $\mathbf{H}_{\mathbf{w}}(\boldsymbol{\lambda})$ and the objective function of (20) as well. But it can be similarly addressed as follows by noting that the second term in $\mathbf{H}_{\mathbf{w}}(\boldsymbol{\lambda})$ can be rewritten as

$$\begin{aligned} & \left[\widetilde{\mathbf{X}}^T \widetilde{\mathbf{X}} + \begin{pmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & \mathrm{diag}(\mathbf{w})\mathrm{diag}(\boldsymbol{\lambda}^{-1}) \end{pmatrix} \right]^{-1} \\ = & \begin{pmatrix} 1/n & \mathbf{0}^T \\ \mathbf{0} & \left[\mathbf{X}^T\mathbf{X} + \mathrm{diag}(\mathbf{w})\mathrm{diag}(\boldsymbol{\lambda}^{-1}) \right]^{-1} \end{pmatrix} \end{aligned}$$

and

$$\left[\mathbf{X}^T\mathbf{X} + \mathrm{diag}(\mathbf{w})\mathrm{diag}(\boldsymbol{\lambda}^{-1}) \right]^{-1} = \mathrm{diag}(\sqrt{\boldsymbol{\lambda}}) \left[\mathrm{diag}(\sqrt{\boldsymbol{\lambda}})\mathbf{X}^T\mathbf{X}\mathrm{diag}(\sqrt{\boldsymbol{\lambda}}) + \mathrm{diag}(\mathbf{w}) \right]^{-1} \mathrm{diag}(\sqrt{\boldsymbol{\lambda}}).$$

For the choice of weight parameter w_j, we may use the reciprocal of the absolute value of the corresponding component of the ordinary least squares estimate as done in the adaptive LASSO (Zou 2006; Zhang and Lu 2007). This adaptive version of our newly proposed variable selection method is named as adaptive ridge selection operator (aRSO).

4.3 The orthonormal design case

The above discussion for the orthonormal design case extends straightforwardly to the aRSO. For the adaptive RSO, (16) simplifies to

$$\begin{aligned} \min_{\boldsymbol{\lambda}} \quad & \sum_{j=1}^{p} \frac{1}{(1+\lambda_j/w_j)^2} \widetilde{\beta}_j^2 \\ \text{subject to} \quad & \lambda_j \geq 0, j = 1, 2, \ldots, p; \\ & \sum_{j=1}^{p} \lambda_j \leq \tau \end{aligned} \tag{21}$$

in the orthonormal design case.

For the above toy example, the corresponding paths for this adaptive version (21) with weights $\mathbf{w} = |\widetilde{\boldsymbol{\beta}}|^{-1}$ are given in the top-right and bottom-right panels of Figure 1. A similar pattern is observed. But the rapid early-stage change in $\widehat{\boldsymbol{\beta}}$ is more dramatic, making the aRSO estimate even less biased. That is exactly the advantage of the weighted version by using a smaller weight on the ridge penalty term for predictors with regression coefficients of a larger magnitude.

5 Simulation studies

In this example, data are generated from the model

$$Y = \beta_0 + \mathbf{X}^T \boldsymbol{\beta} + \sigma\epsilon,$$

where $\beta_0 = 0$, $\boldsymbol{\beta} = (3, 1.5, 0, 0, 2, 0, 0, 0)^T$, and $\epsilon \sim N(0, 1)$. The predictors are generated from a multivariate Gaussian distribution with zero mean vector and $\text{cov}(X_j, X_k) = 0.5^{|j-k|}$. This example is taken from Example 1 of Fan and Li (2001). An independent test set of size 10000 is generated to evaluate prediction performance of each method being considered. Different signal-to-noise ratios will be considered by varying σ and training data of different sample sizes will be used to illustrate the performance of the proposed (adaptive) RSO.

We compare the proposed (adaptive) RSO with the (adaptive) LASSO. Both methods involve some regularization parameter to be tuned. In our implementation, we tune any necessary regularization parameter by using BIC as it has been shown that the BIC is better for variable selection consistency than AIC or cross validation (Wang et al. 2007). The calculation of BIC requires to gauge the degrees of freedom. For any variable selection method, its estimate's number of nonzero components can be used as one measure of the

degrees of freedom. Note that the (adaptively weighted) individually penalized ridge regression estimate has a closed-form representation (4) and (18). The corresponding hat matrix is well defined as presented above in (5) and (19). In this case, the trace of the corresponding hat matrix can also be used as another measure of the degrees of freedom according to Ye (1998). But this alternative choice is not available for the (adaptive) LASSO. For our new method (adaptive) RSO, we implement BIC using both measures of degrees of freedom. We use BICtr and BIC0 to denote BIC based on the trace of the hat matrix and the number of nonzero estimate components, respectively.

5.1 Simulation results

As in Fan and Li (2001), we consider three cases: $\sigma = 1$ and $n = 60$, $\sigma = 1$ and $n = 40$, and $\sigma = 3$ and $n = 40$. For these three cases, Tables 1, S.1 and S.3 (in Supplementary Materials) summarize the corresponding simulation results of the comparison between the proposed variable selection method (adaptive) RSO and the (adaptive) LASSO over 100 random repetitions. We report the frequency for each predictor being selected and average prediction error (with standard errors in parentheses) over the independent test set. The last column reports the frequency of solution path being consistent, namely the true model with X_1, X_2 and X_5 is included along the solution path (Yuan and Lin 2007).

It is observed that the important predictors X_1, X_2, and X_5 are selected every time out of the 100 repetitions for all four methods in the two cases with $\sigma = 1$. When the signal-to-noise ratio decreases as σ increases to 3, the important predictors are not selected all the time, but still with a high frequency. On the other hand, unimportant predictors are selected at relatively low frequency. Overall, the (adaptive) LASSO is shown to have a better variable selection performance than the (adaptive) RSO. Especially, the adaptive LASSO does much better.

To figure out why the (adaptive) RSO is not performing well in terms of variable selection, we take a closer look at the (adaptive) RSO solution path. In the last column of these three tables, we report the frequency of solution path consistency. We are surprised to observe that the (adaptive) RSO does no worse, even slightly better, than the (adaptive) LASSO in terms of solution path consistency. It indicates that the unsatisfactory variable selection performance of the (adaptive) RSO is not due to the method itself. It may be blamed on the tuning method we have adopted.

Note that the LASSO, RSO, and (adaptive) RSO all lead to a biased estimate. Only the adaptive LASSO was shown to be asymptotically unbiased. We propose a refitting step as follows to correct estimation bias for the regression coefficients. For each (adaptive)

RSO or (adaptive) LASSO estimate $\widehat{\boldsymbol{\beta}}$ along the solution path, we denote the corresponding estimated set of important predictors by $\widetilde{\mathcal{A}} = \{j : \widehat{\beta}_j \neq 0\}$. We obtain the refitted regression coefficients estimate by performing ordinary least squares regression of Y on $\{X_j : j \in \widetilde{\mathcal{A}}\}$, where no penalty term is used to avoid introducing estimation bias. To be more specific, for our proposed RSO, the RSO estimate is $\widehat{\boldsymbol{\beta}}(\widehat{\boldsymbol{\lambda}}(\tau))$ using the above notations in Section 3. For each τ, the corresponding estimate set of important predictors is $\widetilde{\mathcal{A}}(\tau) = \{j : \widehat{\beta}_j(\widehat{\boldsymbol{\lambda}}(\tau)) \neq 0\}$, and refitting is done by performing ordinary least squares regression of Y on $\{X_j : j \in \widetilde{\mathcal{A}}(\tau)\}$.

The performance of different methods after the refitting step is shown in Tables 2, S.2 and S.4 (in Supplementary Materials). Note that the BICtr is not appropriate for the (adaptive) RSO with refitting any more and is not used. We observe that the refitting improves the variable selection results for all methods. The proposed (adaptive) RSO performs competitively, even with a slight advantage, and the adaptive RSO does better than the RSO.

All numerical examples are done in R (R Core Team 2018). The (adaptive) LASSO was implemented using the "glmnet" package, which provides a full solution path. On the other hand, the tuning of the proposed (adaptive) RSO is based on a grid search in our implementation. We know that the solution path-based tuning is computationally much more efficient. Considering this, we track the CPU time needed to run the RSO or LASSO for a single tuning parameter. On average the RSO takes 59 milliseconds while the LASSO is much faster and takes only 1 milliseconds on a laptop equipped with Intel Core i7-7600U CPT 2.80GHz. We admit that the proposed RSO (resp. adaptive RSO) is slower than the LASSO (resp. adaptive LASSO). Yet it is fast enough to run all numerical examples on a laptop.

Upon the request of a reviewer during the review process, we also include comparison with SCAD (Fan and Li 2001), MCP (Zhang 2010), and SDAR (Huang et al. 2018). It is observed that SCAD, MCP and SDAR perform better than LASSO, RSO and aRSO as shown in Tables 1, S.1 and S.3 (in Supplementary Materials). Yet Tables 2, S.2 and S.4, shows that with a refitting step LASSO, RSO and aRSO perform very similarly as SCAD, MCP and SDAR.

5.2 Solution path

For a random sample of the case with $n = 60$ and $\sigma = 1$, Figure 2 presents the corresponding solution paths of $\widehat{\boldsymbol{\lambda}}$ (in top panels) and $\widehat{\boldsymbol{\beta}}$ (in bottom panels). The left panels are for the RSO while the right panels for the adaptive RSO. It clearly shows that the

Table 1: Simulation results for $\sigma = 1$ and $n = 60$.

		Selection frequency								test error	path
		X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8		
aRSO	BICtr	100	100	28	27	100	29	32	37	1.144 (0.008)	100
	BIC0	100	100	19	23	100	24	22	26	1.160 (0.009)	
aLASSO	BIC0	100	100	3	6	100	3	3	5	1.110 (0.007)	100
RSO	BICtr	100	100	42	49	100	48	53	67	1.188 (0.010)	95
	BIC0	100	100	41	43	100	43	41	61	1.198 (0.011)	
LASSO	BIC0	100	100	28	26	100	21	24	21	1.163 (0.010)	95
SCAD	BIC0	100	100	2	6	100	3	5	5	1.109 (0.007)	100
MCP	BIC0	100	100	3	7	100	3	5	5	1.114 (0.008)	100
SDAR	BIC0	100	100	3	8	100	3	4	6	1.115 (0.008)	98

Table 2: Post-refitting simulation results for $\sigma = 1$ and $n = 60$.

		Selection frequency								test error
		X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	
aRSO	BIC0	100	100	2	6	100	3	4	7	1.116 (0.008)
aLASSO	BIC0	100	100	3	8	100	3	4	7	1.118 (0.008)
RSO	BIC0	100	100	2	10	100	2	2	6	1.114 (0.008)
LASSO	BIC0	100	100	2	10	100	2	2	6	1.114 (0.008)

(adaptive) RSO can really perform variable selection and the adaptive RSO produce less bias with the more rapid early-stage change as explained earlier in the toy orthonormal design example.

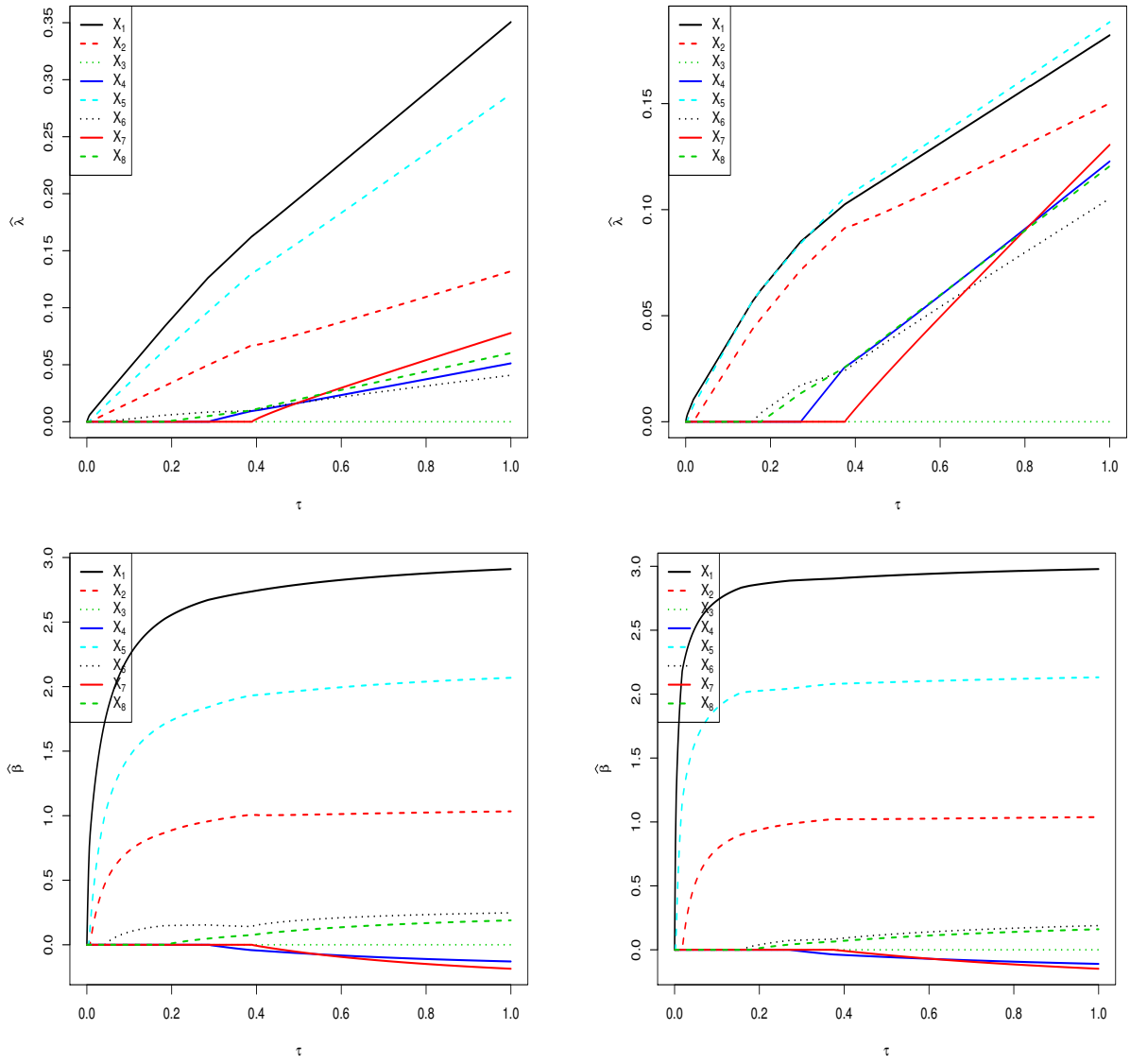

Figure 2: Solution paths of a random sample of the example in Section 5.

5.3 High dimensional case

We next consider a high dimensional case with $p = 1000$ predictors. Predictors are generated in the same way and the response is also generated in the same way $Y = 3X_1 + 1.5X_2 + 2X_5 + \sigma\epsilon$ as above. In Table 3, we report the simulation results over 100 repetitions for the case with $n = 200$ and $\sigma = 1$ only. Since the dimensionality $p = 1000$ is larger than the sample size $n = 200$, the ordinary least squares estimate is not well defined. In this case, it is not clear how to specify the adaptive weights in the adaptive

RSO and adaptive LASSO, and consequently we only apply the RSO and LASSO. For the simulation results reported above, we have learned that the tuning with BIC0 does slightly better than the tuning with BICtr. In this example, we only choose to implement the tuning with BIC0.

In Table 3, we report the selection frequency of important predictors X_1, X_2, and X_5 as well as the average selection frequency of unimportant predictors. It is observed that all the important predictors are selected across all 100 repetitions while the unimportant predictors are selected at very low frequency for both methods. The solution path consistency is also high for both methods.

A refitting step could potentially be added to improve variable selection results as well as prediction results in terms of test error as done above. In addition, note also that the number of predictors is larger than the sample size in this example. It was shown that the BIC is not an optimal tuning method for such a case (Chen and Chen 2008). They proposed an extended BIC, which has been shown to perform better in terms of variable selection consistency for the case with a diverging p. We may also try to improve our simulation results by using the extended BIC. Yet the main goal of this current example is to illustrate that the proposed RSO can be applied to high dimensional data even when the dimensionality is larger than the sample size. As a result, we skip the refitting step and do not implement the extended BIC.

Table 3: Simulation results for a high dimensional case with $\sigma = 1$, $n = 200$, and $p = 1000$.

	Selection frequency				test error	path
	Important			Average over unimportant		
	X_1	X_2	X_5	$X_j : j \notin \mathcal{A}$		
RSO	100	100	100	4.35 (0.34)	1.710 (0.335)	100
LASSO	100	100	100	1.13 (0.14)	1.173 (0.136)	100
SCAD	100	100	100	0.05 (0.02)	1.044 (0.002)	100
MCP	100	100	100	0.28 (0.08)	1.046 (0.002)	100
SDAR	100	100	100	6.57 (0.09)	1.353 (0.008)	100

6 A real data example

In this section, we are going to use one real dataset to illustrate the performance of our new methods in comparison with the (adaptive) LASSO. The dataset is available at the UCI Machine Learning Repository at `https://archive.ics.uci.edu/ml/datasets/Superconductivty+Data`. It contains 81 features extracted from 21263 superconductors along with the critical temperature. The goal is to predict the critical temperature based on these features. See Hamidieh (2018) for more details on the dataset's background. We simply consider the linear regression model by regressing the critical temperature on these 81 features. We randomly sample 200 observations as training data. The rest is used at the test set. We apply the proposed (adaptive) RSO or the (adaptive) LASSO on the training data to perform variable selection with BIC0 tuning. The selected model for each method is applied to the test set to evaluate its prediction performance in terms of test error. The whole process is repeated for 20 times. We report the average number of selected features and the average test error with refitting and without refitting for all four methods: RSO, LASSO, aRSO, and aLASSO in the first half of Table 4. The performance of SCAD, MCP and SDAR is reported similarly in the second half of Table 4. Numbers in parentheses are the corresponding standard errors. It shows that the (adaptive) RSO and the (adaptive) LASSO perform similarly. The (adaptive) RSO does slightly better in terms of prediction error by selecting a couple of more features.

Table 4: Results of the real data example.

	without refitting		with refitting	
	no.	test error	no.	test error
RSO	12.35(0.66)	452.332(8.140)	10.15(0.93)	457.959(8.385)
LASSO	9.65(0.87)	467.456(8.687)	10.40(1.24)	460.664(9.314)
aRSO	14.05(0.78)	443.133(7.379)	15.30(0.95)	438.644(8.123)
aLASSO	11.95(1.32)	474.880(11.055)	13.55(1.09)	471.784(9.689)
SCAD	12.05(0.74)	457.185(5.823)		
MCP	9.65(0.78)	466.867(6.639)		
SDAR	8.40(1.06)	470.813(6.467)		

7 Discussion

In this paper, we propose a new variable selection method based on ridge regression, which has been criticized of not being able to perform variable selection. The proposed method requires tuning of the corresponding regularization parameter, which is typical for many variable selection methods. The current implementation is based on a grid search over a set of candidate regularization parameters. This grid search tuning method has long been recognized to be computationally inefficient. For many existing variable selection methods, solution path following algorithms have been developed. Efficient regularization parameter tuning can be achieved based on solution paths. It is not immediately clear how to develop a solution path following algorithm for the proposed new methods of variable selection. This can be a potential future research topic to be investigated. During the review process, AE and one reviewer asked whether the proposed new method can be extended to generalized linear model (GLM) and Cox proportional-hazards model (COX). In theory, the answer is positive as long as the ridge penalized solution for GLM and COX is well defined. Yet the ridge penalized solution for GLM and COX does not have a closed-form expression as in the linear regression case. This will make the corresponding implementation very computationally challenging.

Acknowledgment

We thank three reviewers, an associate editor, and the editor for their most helpful comments that lead to substantial improvements in the paper. The research is partially supported by NSF grants DMS-1821171 and CCF-1934915.

Supplementary Materials

Additional simulation results: A separate pdf file contains the simulation results of the simulation example in Section 5.1 for the case with $\sigma = 1$ and $n = 40$, and the case with $\sigma = 3$ and $n = 40$.

R code: A file (RSO.R) contains the R code for the ridge selection operator and another file (demoRSO.R) demonstrates how to apply the ridge selection operator and perform refitting.

REFERENCES

Anzanello, M. J. and F. S. Fogliatto (2014). A review of recent variable selection methods in industrial and chemometrics applications. *European J. of Industrial Engineering 8*, 619–645.

Barcella, W., M. De Iorio, and G. Baio (2017). A comparative review of variable selection techniques for covariate dependent dirichlet process mixture models. *Canadian Journal of Statistics 45*, 254–273.

Chen, J. and Z. Chen (2008). Extended Bayesian information criteria for model selection with large model spaces. *Biometrika 95*(3), 759–771.

Desboulets, L. D. D. (2018). A Review on Variable Selection in Regression Analysis. *Econometrics 6*(4), 1–27.

Fan, J. and R. Li (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association 96*(456), 1348–1360.

Fan, J. and J. Lv (2010). A selective overview of variable selection in high dimensional feature space. *Statistica Sinica 20*(1), 101–148.

Frommlet, F. and G. Nuel (2016). An adaptive ridge procedure for l0 regularization. *PLOS ONE 11*(2), 1–23.

Hamidieh, K. (2018). A data-driven statistical model for predicting the critical temperature of a superconductor. *Computational Materials Science 154*, 346 – 354.

Hastie, T. (2020). Ridge Regularizaton: an Essential Concept in Data Science. *Technometrics*, to appear.

Heinze, G., C. Wallisch, and D. Dunkler (2018). Variable selection – a review and recommendations for the practicing statistician. *Biometrical journal 60*, 431–449.

Hoerl, A. E. and R. W. Kennard (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics 12*, 55–67.

Huang, J., Y. Jiao, Y. Liu, and X. Lu (2018). A constructive approach to l_0 penalized regression. *Journal of Machine Learning Research 19*(10), 1–37.

Joseph, V. R. and J. D. Delaney (2008). Analysis of optimization experiments. *Journal of Quality Technology 40*, 282–298.

Kirpich, A., E. A. Ainsworth, J. M. Wedow, J. R. B. Newman, G. Michailidis, and L. M. McIntyre (2018). Variable selection in omics data: A practical evaluation of small sample sizes. *PLOS ONE 13*, e0197910.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.

Shao, J. and X. Deng (2012). Estimation in high-dimensional linear models with deterministic design matrices. *The Annals of Statistics 40*(2), 812–831.

Stefanski, L. A., Y. Wu, and K. R. White (2014). Variable selection in nonparametric classification via measurement error model selection likelihoods. *Journal of the American Statistical Association 109*, 574–589.

Talbot, D. (2019). A descriptive review of variable selection methods in four epidemiologic journals: there is still room for improvement. *European Journal of Epidemiology 34*, 725–730.

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

Wang, H., R. Li, and C.-L. Tsai (2007). Tuning parameter selectors for the smoothly clipped absolute deviation method. *Biometrika 94*(3), 553–568.

Wipf, D. P. and S. S. Nagarajan (2008). A new view of automatic relevance determination. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis (Eds.), *Advances in Neural Information Processing Systems 20*, pp. 1625–1632. Curran Associates, Inc.

Ye, J. (1998). On measuring and correcting the effects of data mining and model selection. *Journal of the American Statistical Association 93*(441), 120–131.

Yuan, M. and Y. Lin (2007). On the non-negative garrotte estimator. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 69*(2), 143–161.

Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics 38*(2), 894–942.

Zhang, H. H. and W. Lu (2007). Adaptive Lasso for Cox's proportional hazards model. *Biometrika 94*(3), 691–703.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association 101*(476), 1418–1429.