Strong rules for nonconvex penalties and their implications for efficient algorithms in high-dimensional regression

Sangin Lee and Patrick Breheny

The University of Iowa

Abstract

We consider approaches for improving the efficiency of algorithms for fitting nonconvex penalized regression models such as SCAD and MCP in high dimensions. In particular, we develop rules for discarding variables during cyclic coordinate descent. This dimension reduction leads to a substantial improvement in the speed of these algorithms for high-dimensional problems. The rules we propose here eliminate a substantial fraction of the variables from the coordinate descent algorithm. Violations are quite rare, especially in the locally convex region of the solution path, and furthermore, may be easily detected and corrected by checking the Karush-Kuhn-Tucker conditions. We extend these rules to generalized linear models, as well as to other nonconvex penalties such as the ℓ_2-stabilized Mnet penalty, group MCP, and group SCAD. We explore three variants of the coordinate decent algorithm that incorporate these rules and study the efficiency of these algorithms in fitting models to both simulated data and on real data from a genome-wide association study.

Keywords: Coordinate descent algorithms, Local convexity, Nonconvex penalties, Dimension reduction.

1 Introduction

Consider the linear regression model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{1}$$

where $\mathbf{y} = (y_1, \cdots, y_n)'$ is the vector of n response variables, $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_p)$ is the $n \times p$ design matrix with the jth column $\mathbf{x}_j = (x_{1j}, \ldots, x_{nj})'$, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)'$ is the vector of regression coefficients and $\boldsymbol{\varepsilon} = (\varepsilon_1, \cdots, \varepsilon_n)'$ is the vector of random errors. We assume that the responses and covariates are centered so that the intercept term is zero. We are interested in estimating the vector of regression coefficients $\boldsymbol{\beta}$. Penalized regression methods accomplish this by minimizing an objective function Q that is composed of the sum of squared residuals plus a penalty. The penalized least squares estimator is defined as the minimizer of

$$Q_{\lambda,\gamma}(\boldsymbol{\beta}) = \frac{1}{2n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \sum_{j=1}^{p} J_{\lambda,\gamma}(|\beta_j|), \tag{2}$$

where $J_{\lambda,\gamma}(\cdot)$ is a penalty function indexed by a regularization parameter λ that controls the balance between the fit of the model and the penalty, and the penalty function may depend on one or more tuning parameters γ.

Here we focus on optimization algorithms for penalized regression methods. There has been much work on developing efficient algorithms for many problems with various penalties, including Efron et al. (2004), Friedman et al. (2007), and Wu and Lange (2008) for the least absolute selection operator (LASSO), and Kim et al. (2008), Zou and Li (2008), and Breheny and Huang (2011) for nonconvex penalties such as smoothly clipped absolute deviation (SCAD) and the minimax concave penalty (MCP). Recently, several authors have investigated rules for discarding variables during certain steps of the above algorithms, thereby saving computational time through dimension reduction.

For the LASSO, El Ghaoui et al. (2011) proposed the basic SAFE rule discards the jth variable if

$$|\mathbf{x}_j'\mathbf{y}/n| < \lambda - \frac{1}{n}\|\mathbf{x}_j\|_2\|\mathbf{y}\|_2(\lambda_{\max} - \lambda)/\lambda_{\max}, \tag{3}$$

where $\lambda_{\max} = \max_j |\mathbf{x}_j'\mathbf{y}/n|$ is the smallest tuning parameter value for which all estimated coefficients are zero. They proved that the estimated coefficient for any variable satisfying the basic SAFE rule (3) must be zero in the solution at λ. Tibshirani et al. (2012) proposed the basic strong rule by modifying the basic SAFE rule (3). For a standardized design matrix ($\|\mathbf{x}_j\|_2/\sqrt{n} = 1$ for all j), we have $\|\mathbf{y}\|_2/\sqrt{n} > \lambda_{\max}$ by the Cauchy-Schwarz inequality and therefore $2\lambda - \lambda_{\max}$ is an upper bound of the quantity on the right hand side of (3). The strong rule therefore discards the jth variable if

$$|\mathbf{x}_j'\mathbf{y}/n| < 2\lambda - \lambda_{\max}. \tag{4}$$

Being an upper bound of the SAFE rule, the strong rule (4) discards more variables than the SAFE rule. Unlike the SAFE rule, however, it is possible for the strong rule to be violated. Because strong rules can mistakenly discard active variables (i.e., variables whose solution is nonzero for that value of λ), Tibshirani et al. (2012) proposed checking the discarded variables against the Karush-Kuhn-Tucker (KKT) conditions to correct for any violations that may have occurred during the optimization.

These basic rules are most useful at large values of λ and rarely eliminate variables at smaller λ values. This is unfortunate from an algorithmic perspective, since the majority of time required to fit a regularization path is spent during optimization for the small λ values. To overcome this drawback, Tibshirani et al. (2012) proposed sequential strong rules. For a decreasing sequence of tuning parameter $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$, the sequential strong rule discards the jth variable from the optimization problem at λ_k if

$$|\mathbf{x}_j'\mathbf{r}_{k-1}/n| < 2\lambda_k - \lambda_{k-1}, \tag{5}$$

where $\mathbf{r}_{k-1} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}(\lambda_{k-1})$ is the vector of residuals at λ_{k-1}. Unlike the basic rules, the sequential strong rule discards a large proportion of inactive variables at all values of λ. In addition, the rule is rarely violated, and is therefore unlikely to discard active variables by mistake.

In this paper, we investigate sequential strong rules for discarding variables in penalized regression with nonconvex penalties, as well as strategies for incorporating these rules into coordinate descent algorithms for fitting these models. In addition, we derive rules for discarding variables in various related problems with nonconvex penalties, such generalized linear models, ℓ_2-stabilized penalties (the "Mnet" estimator), and grouped penalties. We provide a publicly available implementation of these algorithms in the updated `ncvreg` package (available at `http://cran.r-project.org`), which was used to fit all the models in this paper.

2 Strong rules in nonconvex penalized regression

The basic idea of the sequential strong rule (Tibshirani et al., 2012) is that the solution path $\hat{\boldsymbol{\beta}}(\lambda)$ is a continuous function; furthermore, one can obtain an approximate bound on how fast the solution path can change as a function of lambda. Thus, when solving for $\hat{\boldsymbol{\beta}}$ at λ_{k-1} and then again at λ_k, we can exclude certain variables from the optimization procedure because they aren't close enough to the threshold for inclusion in the model to reach that threshold in the distance between λ_{k-1} and λ_k. The effect is that the dimension of the optimization problem is reduced – instead of cycling over all variables, the estimation procedure needs only to cycle over a much smaller set of variables capable of entering the model at λ_k.

The bound investigated by (Tibshirani et al., 2012) is given by

$$|c_j(\lambda) - c_j(\tilde{\lambda})| \leq |\lambda - \tilde{\lambda}|, \text{ for any } \lambda \text{ and } \tilde{\lambda}, \tag{6}$$

where $c_j(\lambda) = \mathbf{x}_j'\mathbf{r}(\lambda)/n$ is the correlation[1] between variable j and the residual at λ. This condition is equivalent to $c_j(\lambda)$ being continuous everywhere, differentiable almost everywhere, and satisfying $|\nabla c_j(\lambda)| \leq 1$ wherever this derivative exists. Tibshirani et al. called condition (6) the unit slope bound. If condition (6) holds, then for any variable j satisfying the sequential strong rule (5), we have $c_j(\lambda_k) < \lambda_k$, and thus, $\hat{\beta}_j(\lambda_k) = 0$ by the KKT conditions of the LASSO.

In this section, we examine whether a variation of condition (6) holds for the MCP and SCAD penalties, and use a modified version of (6) to develop strong rules for those nonconvex penalties. Also, we provide numerical examples to illustrate the application of the strong rules on a simulated data set. Lastly, we note that unlike the LASSO case, for nonconvex penalties the function $c_j(\lambda)$ is not guaranteed to be continuous; we explore the consequences of this fact in Section 2.4.

[1] Strictly speaking, $c_j(\lambda)$ is not a correlation, since $\mathbf{r}$ is not standardized, and thus only proportional to the correlation between $\mathbf{x}_j$ and $\mathbf{r}$. However, the term is widely used; see, e.g., Efron et al. (2004).

2.1 MCP

Zhang (2010) proposed the MCP which is defined as

$$J_{\lambda,\gamma}(t) = \begin{cases} -t^2/(2\gamma) + \lambda t, & \text{if } t \leq \gamma\lambda, \\ \gamma\lambda^2/2, & \text{if } t > \gamma\lambda. \end{cases}$$

for $\lambda \geq 0$ and $\gamma > 1$. We begin by noting the KKT conditions for the penalized problem (2),

$$\begin{aligned} \mathbf{x}_j'\mathbf{r}/n &= \nabla J_\lambda(|\hat{\beta}_j|) && \text{for all } j \in \mathcal{A}, \\ |\mathbf{x}_j'\mathbf{r}/n| &< \lambda && \text{for all } j \notin \mathcal{A} \end{aligned} \tag{7}$$

where $\mathcal{A} = \{j : \hat{\beta}_j \neq 0\}$ is the active set.

Variables in $\mathcal{A}$ are continuously changing as a function of λ, but many variables in $\mathcal{A}^c$ remain zero from one λ value to the next. Our aim, then, is to develop a screening rule to can discard the variables in the inactive set $\mathcal{A}^c$ that are likely to remain zero. In high dimensions, doing so should yield substantial computational savings. From the KKT conditions (7), we have the form of $c_j(\lambda)$,

$$c_j(\lambda) = \begin{cases} 0, & \text{if } |\hat{\beta}_j| > \gamma\lambda, \\ -\hat{\beta}_j/\gamma + \lambda \text{sign}(\hat{\beta}_j) & \text{if } |\hat{\beta}_j| \leq \gamma\lambda,\ \hat{\beta}_j \neq 0 \\ \mathbf{x}_j'\mathbf{X}_{\mathcal{A}}(\mathbf{X}_{\mathcal{A}}'\mathbf{X}_{\mathcal{A}})^{-1}\nabla J_\lambda(|\hat{\boldsymbol{\beta}}_{\mathcal{A}}|) + (Const), & \text{if } \hat{\beta}_j = 0, \end{cases} \tag{8}$$

where $\boldsymbol{\beta}_{\mathcal{A}} = (\beta_j, j \in \mathcal{A})$ and $\mathbf{X}_{\mathcal{A}} = (\mathbf{x}_j, j \in \mathcal{A})$ denote the subvector and submatrix of $\boldsymbol{\beta}$ and $\mathbf{X}$, respectively, and $(Const)$ stands for constant terms not depending on λ. Unlike the LASSO, the above expression for $c_j(\lambda)$ does not permit a closed-form expression for $\nabla c_j(\lambda)$ for variables in the active set. Hence, we investigate an approximation for $\nabla c_j(\lambda)$ based on an orthogonal design matrix. In this case, the coefficient estimates have closed form solution $\hat{\beta}_j = \frac{\gamma}{\gamma-1}\text{sign}(z_j)(|z_j| - \lambda)_+$, where $z_j = \mathbf{x}_j'\mathbf{y}/n$ is the ordinary least squares estimator, and the second term of (8) is $c_j(\lambda) = \text{sign}(z_j)\left\{\lambda - \frac{1}{\gamma-1}(|z_j| - \lambda)_+\right\}$. This suggests the bound $|\nabla c_j(\lambda)| \leq 1 + 1/(\gamma - 1)$. This slope bound is larger than the corresponding bound for the LASSO, as the nonconvexity of MCP allows the solution path – and thus, $c_j(\lambda)$ – to change more rapidly as a function of λ than it does for LASSO. Note that in the limiting case $\gamma \to \infty$, MCP is equal to the lasso penalty, and the bounds coincide. Conversely, as $\gamma \to 1$, MCP is equivalent to hard thresholding. The bound diverges in this case, and there is no limit to the rate at which the solution path may change and no possibility of discarding variables based on this argument.

As in the LASSO case, a slope bound for variables in the active set does not necessarily extend to variables in the inactive set. Nevertheless, it is reasonable to expect that the correlation with the residuals is changing more rapidly for variables in the active set than variables in the inactive set. This line of thinking that allows us to establish an explicit rule for screening predictors during optimization.

If, for $j = 1, \ldots, p$, the bound

$$|c_j(\lambda) - c_j(\tilde{\lambda})| \leq \frac{\gamma}{\gamma-1}|\lambda - \tilde{\lambda}|, \text{ for any } \lambda \text{ and } \tilde{\lambda}, \tag{9}$$

holds, we can obtain the following rule, which we call the (sequential) strong rule for MCP:

$$|\mathbf{x}_j'\mathbf{r}_{k-1}/n| < \lambda_k + \frac{\gamma}{\gamma-1}(\lambda_k - \lambda_{k-1}). \tag{10}$$

Note that, for any variable j satisfying (9) and (10), we have

$$\begin{aligned} |c_j(\lambda_k)| &\leq |c_j(\lambda_k) - c_j(\lambda_{k-1})| + |c_j(\lambda_{k-1})| \\ &< \frac{\gamma}{\gamma-1}(\lambda_{k-1} - \lambda_k) + \lambda_k + \frac{\gamma}{\gamma-1}(\lambda_k - \lambda_{k-1}) \\ &= \lambda_k, \end{aligned}$$

and thus, $\hat{\beta}_j(\lambda_k) = 0$.

Indeed, as we shall see, the heuristic argument that residual correlation changes more rapidly in the active set than the inactive set holds up quite well in practice. Nevertheless, violations are possible, and thus it is necessary to check the discarded variables against the KKT conditions (7) as a final step in the optimization algorithm.

2.2 SCAD

The SCAD penalty proposed by Fan and Li (2001) is defined as

$$J_{\lambda,\gamma}(t) = \begin{cases} \lambda t, & \text{if } t \leq \lambda, \\ \{\gamma\lambda(t-\lambda) - (t^2 - \lambda^2)/2\}/(\gamma-1), & \text{if } t \leq \gamma\lambda, \\ (\gamma-1)\lambda^2/2 + \lambda^2, & \text{if } t > \gamma\lambda. \end{cases}$$

for $\lambda \geq 0$ and $\gamma > 2$. From the KKT conditions (7), we have

$$c_j(\lambda) = \begin{cases} 0, & \text{if } |\hat{\beta}_j| > \gamma\lambda \\ (\gamma\lambda\text{sign}(\hat{\beta}_j) - \hat{\beta}_j)/(\gamma-1), & \text{if } \lambda < |\hat{\beta}_j| \leq \gamma\lambda \\ \lambda\text{sign}(\hat{\beta}_j), & \text{if } |\hat{\beta}_j| \leq \lambda, \ \hat{\beta}_j \neq 0 \\ \mathbf{x}_j'\mathbf{X}_{\mathcal{A}}(\mathbf{X}_{\mathcal{A}}^T\mathbf{X}_{\mathcal{A}})^{-1}\nabla J_\lambda(|\hat{\boldsymbol{\beta}}_{\mathcal{A}}|) + (Const), & \text{if } \hat{\beta}_j = 0, \end{cases}$$

where $(Const)$ stands for constant terms not depending on λ. For SCAD, the orthogonal design solution is $\hat{\beta}_j = \text{sign}(z_j)(\frac{\gamma-1}{\gamma-2})\{|z_j| - \lambda\gamma/(\gamma-1)\}_+$. Applying the same reasoning as in Section 2.1, we obtain the approximate slope bound $|\nabla c_j(\lambda)| \leq 1 + 2/(\gamma-2)$ and the sequential strong rule for SCAD

$$|\mathbf{x}_j'\mathbf{r}_{k-1}/n| < \lambda_k + \frac{\gamma}{\gamma-2}(\lambda_k - \lambda_{k-1}). \tag{11}$$

Like MCP, the SCAD solution path is capable of changing more rapidly with respect to λ than LASSO, and thus requires a larger bound for its strong rule.

2.3 Numerical illustrations

We now provide an illustration of the how the strong rules perform using a simulated example. The design of the simulation, which we also use for the simulation study in Section 4, is as follows. All covariates marginally follow standard Gaussian distributions, with a common correlation ρ between any two covariates. The response variable y is generated from the linear model (1) with errors drawn from the standard Gaussian distribution. For each independently generated data set, we set $n = 200$ and $p = 2,000$, with 20 nonzero coefficients set to be ± 1 for linear regression and the remaining $1,980$ coefficients equal to zero. Throughout this paper, we fix $\gamma = 3$ for MCP and $\gamma = 4$ for SCAD, roughly in line with recommendations suggested in Fan and Li (2001) and Zhang (2010), respectively.

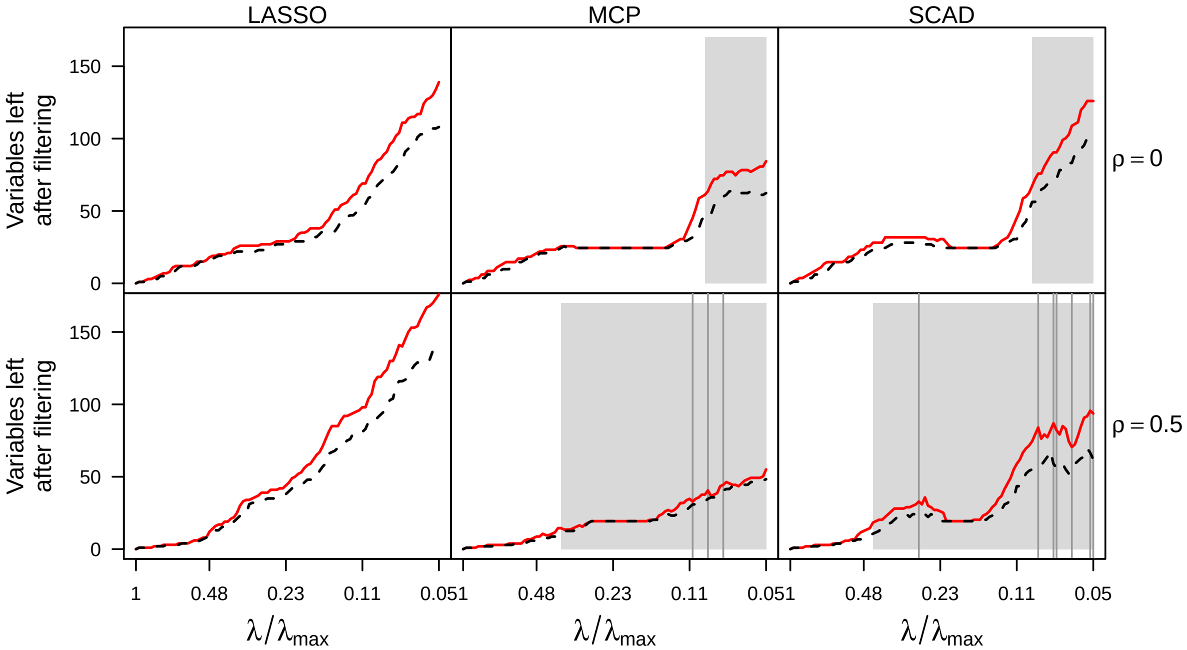

Figure 1: Application of the strong rules (5, 10, and 11) for two simulated data sets, one with $\rho = 0$ (top panel) and the other with $\rho = 0.5$ (bottom panel). The solid line is the number of variables left after filtering by strong rules and the dotted line is the actual number of active variables for each λ. The region of the coefficient path that does not satisfy local convexity is shaded gray. Vertical lines are drawn for any value of λ at which a violation of the strong rules occurred.

Figure 1 displays the performance of the strong rules for LASSO, MCP, and SCAD on two simulated data sets, one with uncorrelated covariates, the other with a pairwise correlation of $\rho = 0.5$. The figure displays the number of variables remaining (i.e., p minus the number of discarded variables) after applying the strong rules for a decreasing sequence of λ values, alongside the actual number of nonzero coefficients in the model for those λ values. Vertical lines are drawn for each value of λ for which a violation of the strong rules occurred. So in this example, there were no violations for any of the methods when $\rho = 0$, but we observe 3 violations for MCP and 7 violations for SCAD when $\rho = 0.5$.

The strong rules perform remarkably well here, especially for $\rho = 0$. The vast majority of the $p = 2,000$ variables are discarded by the strong rules. In fact, nearly all of the variables that should be discarded are discarded, across the entire path of λ values. With so many variables discarded by the strong rules, it is surprising how rare it is for a variable to be erroneously discarded.

Nevertheless, violations do occur, and are more common for nonconvex penalties than for the LASSO, as we discuss in the next section. Violations are important, but not fatal – an algorithm based on dimension reduction through discarding variables can always check the validity of the dimension reduction by inspecting the KKT conditions for $\hat{\boldsymbol{\beta}}(\lambda)$ upon convergence for each value of λ, and include any variables that were erroneously discarded. In this manner, we ensure that all solutions $\hat{\boldsymbol{\beta}}$ returned by the algorithm are indeed a (local) minimum of the objective function. Details for constructing algorithms based on strong rules are given in Section 5.

Although none occurred in this example, violations are also possible for the lasso, and a similar KKT-checking step is required in the LASSO algorithm proposed by Tibshirani et al. (2012). A systematic numerical study of the frequency of violations for MCP and SCAD is provided in Section 4.

2.4 Local convexity

Unlike the LASSO solution path, for nonconvex penalties $\hat{\boldsymbol{\beta}}$ is not necessarily a continuous function of λ. It is possible for the objective function to possess multiple local minima, and for $\hat{\boldsymbol{\beta}}(\lambda)$ to "jump" from one local minimum to a different local minimum between λ_{k-1} and λ_k. Such a discontinuity undermines the entire premise of strong rules.

It is possible, however, to characterize the regions of the solution path where such discontinuities may and may not occur. The portion of the solution path guaranteed to be continuous was referred to in Breheny and Huang (2011) as the locally convex region. Letting $\mathcal{A}(\lambda) = \{j : \hat{\beta}_j(\lambda) \neq 0\}$ denote the active set of variables at λ, and $\tau_{\min}(\lambda)$ denote the minimum eigenvalue of $\mathbf{X}'_{\mathcal{A}(\lambda)}\mathbf{X}_{\mathcal{A}(\lambda)}/n$, the solution path is said to be locally convex at λ if $\gamma > 1/\tau_{\min}(\lambda)$ for MCP, and $\gamma > 1 + 1/\tau_{\min}(\lambda)$ for SCAD. Correspondingly, the locally convex region is defined as $(\lambda_{\max}, \lambda^*)$, where λ^* is the first (i.e., largest) value of λ for which the solution is no longer locally convex. As demonstrated in Breheny and Huang (2011), coefficient paths for nonconvex penalties are smooth and well behaved in the locally convex region, but may be discontinuous and erratic in the non-locally convex region.

We would therefore expect strong rules to be less reliable in the non-locally convex region, and this is precisely what we see in Figure 1, where all the observed violations occur in the non-locally convex region. Indeed, although this is not apparent in the figure, several violations tend to occur simultaneously when a discontinuity arises in the solution path. For example, at $\lambda = 0.0745$, there were 11 variables excluded by the strong rules that were discovered during the KKT check to be nonzero at the new local minimum $\hat{\boldsymbol{\beta}}(\lambda)$.

The presence of discontinuities in the solution paths for nonconvex penalties places an inherent limitation on the use of sequential rules to improve optimization efficiency during model fitting. Nevertheless, as we will see, even in highly correlated settings, only a small number of λ values experience violations, and strong rules may be profitably incorporated into optimization algorithms for nonconvex penalized models despite these violations, solving for the solution path $\hat{\boldsymbol{\beta}}(\lambda)$ substantially faster than cyclic coordinate descent approaches.

3 Extensions to other nonconvex penalized models

3.1 ℓ_2-stabilization

To stabilize the solution path for nonconvex penalties, especially in $p > n$ problems with highly correlated predictors, Huang et al. (2013) proposed the Mnet estimator, which is defined as the minimizer of

$$Q_{\lambda,\gamma}(\boldsymbol{\beta}) = \frac{1}{2n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \sum_{j=1}^{p} J_{\lambda_1,\gamma}(|\beta_j|) + \frac{1}{2}\lambda_2 \sum_{j=1}^{p} \beta_j^2, \tag{12}$$

where $J_{\lambda_1,\gamma}(\cdot)$ is the MCP. The logic behind the estimator is the same as that of the elastic net (or Enet, Zou and Hastie, 2005), but with MCP replacing the LASSO in the penalty. Let

$$\tilde{\mathbf{X}} = \begin{pmatrix} \mathbf{X} \\ \sqrt{n\lambda_2}\,\mathbf{I}_p \end{pmatrix}, \ \tilde{\mathbf{y}} = \begin{pmatrix} \mathbf{y} \\ \mathbf{0}_p \end{pmatrix},$$

where $\mathbf{I}_p$ is the $p \times p$ identity matrix and $\mathbf{0}_p$ is the p-dimensional vector whose all elements are zero. Then the criterion (12) may be rewritten as

$$\frac{1}{2n}\|\tilde{\mathbf{y}} - \tilde{\mathbf{X}}\boldsymbol{\beta}\|_2^2 + \sum_{j=1}^{p} J_{\lambda_1,\gamma}(|\beta_j|). \tag{13}$$

Hence, we can directly apply the sequential strong rule (10) to discard variables. Reparameterizing the problem in terms of $\lambda_1 = \alpha\lambda$ and $\lambda_2 = (1-\alpha)\lambda$, the strong rule for Mnet becomes

$$\left|-\mathbf{x}_j'\mathbf{r}_{k-1}/n + \lambda_k(1-\alpha)\hat{\beta}_j(\lambda_{k-1})\right| < \alpha\left\{\lambda_k + \frac{\gamma}{\gamma-1}(\lambda_k - \lambda_{k-1})\right\},$$

since $\tilde{\mathbf{x}}_j'\tilde{\mathbf{y}} = \mathbf{x}_j'\mathbf{y}$ and $\tilde{\mathbf{x}}_j'\tilde{\mathbf{X}}\hat{\boldsymbol{\beta}} = \mathbf{x}_j'\mathbf{X}\hat{\boldsymbol{\beta}} + n\lambda(1-\alpha)\hat{\beta}_j$. For inactive variables ($\hat{\beta}_j = 0$), the above rule reduces to

$$\left|\mathbf{x}_j'\mathbf{r}_{k-1}/n\right| < \alpha\left\{\lambda_k + \frac{\gamma}{\gamma-1}(\lambda_k - \lambda_{k-1})\right\}. \tag{14}$$

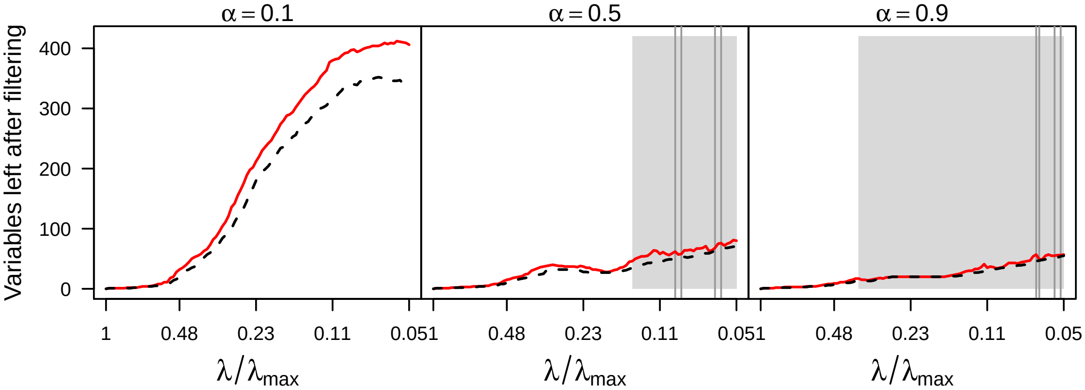

Figure 2: Application of strong rule (14) to a simulated data set with $\rho = 0.5$ for different values of α. As in Figure 1, the solid line is the number of variables left after filtering by strong rules, the dotted line is the actual number of active variables for each λ, the region of the coefficient path that does not satisfy local convexity is shaded gray, and vertical lines are drawn for any value of λ for which a violation of the strong rule occurred.

Figure 2 illustrates the application of strong rules to the Mnet estimator for a simulated data set with $\rho = 0.5$ as we vary the parameter α that controls the MCP/ℓ_2 balance in the penalty. As in Section 2.4, one may characterize the locally convex region; for the Mnet estimator, this consists of the values of λ satisfying $\gamma > 1/\{\tau_{\min}(\lambda) + (1-\alpha)\lambda\}$.

From Figure 2, we can see that as we decrease α and thereby increase the ℓ_2 proportion of the penalty, the locally convex region is extended, the model becomes less sparse, and fewer issues with discontinuities and strong rule violations arise. Indeed, at $\alpha = 0.1$, the objective function is locally convex over the entire solution path and no violations occurred. For all α values, the strong rules were successful in discarding a large proportion of the inactive variables.

3.2 Generalized linear models

Suppose that the distribution of $\mathbf{y}|\mathbf{X}$ falls within the framework of the generalized linear model (GLM), with link function $\eta_i = g(\mu_i)$, where $\mu_i = E(y_i|x_{i1}, \ldots, x_{ip})$ and

$$\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta} = \beta_0 + \mathbf{x}_1\beta_1 + \cdots + \mathbf{x}_p\beta_p, \tag{15}$$

where $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_n)' \in \mathbb{R}^n$ is the vector of linear predictors. The nonconvex penalized estimator for a GLM is defined as the minimizer of the negative log-likelihood plus the penalty term. For example, for logistic regression,

$$-\frac{1}{n}\sum_{i=1}^{n}\{y_i \log \mu_i + (1-y_i)\log(1-\mu_i)\} + \sum_{j=1}^{p} J_{\lambda,\gamma}(|\beta_j|). \tag{16}$$

The extension of strong rules to GLMs is straightforward. Indeed, both the KKT conditions (7) and the strong rules themselves (10, 11) are the same as in the linear case, although the residual vector must now incorporate the link function: $\mathbf{r} = \mathbf{y} - \boldsymbol{\mu}(\boldsymbol{\eta})$. For example, $\boldsymbol{\mu}(\boldsymbol{\eta}) = \exp(\boldsymbol{\eta})/(1+\exp(\boldsymbol{\eta}))$ for logistic regression and $\boldsymbol{\mu}(\boldsymbol{\eta}) = \exp(\boldsymbol{\eta})$ for Poisson regression.

3.3 Nonconvex group penalized estimation

Nonconvex penalties have also been proposed in the context of group variable selection. Suppose that the covariates may be grouped into G groups, with the grouping structure non-overlapping and known in advance:

$$\mathbf{y} = \sum_{g=1}^{G}\mathbf{X}_g\boldsymbol{\beta}_g + \boldsymbol{\varepsilon}. \tag{17}$$

where $\mathbf{X}_g$ is the $n \times p_g$ design matrix corresponding to the gth group and $\boldsymbol{\beta}_g \in \mathbb{R}^{p_g}$ is the vector of corresponding regression coefficients of the gth group. The nonconvex group penalized estimator is defined as the minimizer of

$$Q_{\lambda,\gamma}(\boldsymbol{\beta}) = \frac{1}{2n}\|\mathbf{y} - \sum_{g=1}^{G}\mathbf{X}_g\boldsymbol{\beta}_g\|_2^2 + \sum_{g=1}^{G} J_{\lambda_g,\gamma}(\|\boldsymbol{\beta}_g\|_2), \tag{18}$$

where $J_{\lambda_g,\gamma}(\cdot)$ is a penalty function applied to the ℓ_2-norm of $\boldsymbol{\beta}_g$. It is common practice (Yuan and Lin, 2006; Simon and Tibshirani, 2011) to adjust the regularization parameter for each group using $\lambda_g = \lambda\sqrt{p_g}$ to account for differences in group size. By the KKT conditions, the local minimizer $\hat{\boldsymbol{\beta}} \in \mathbb{R}^p$ satisfies

$$\begin{aligned} \mathbf{X}_g'\mathbf{r}/n &= \lambda\sqrt{p_g}\hat{\boldsymbol{\beta}}_g/\|\hat{\boldsymbol{\beta}}_g\|_2, && g \in \mathcal{A}, \\ \left|\left|\mathbf{X}_g'\mathbf{r}/n\right|\right|_2 &< \lambda\sqrt{p_g}, && g \in \mathcal{A}^c, \end{aligned} \tag{19}$$

where $\mathcal{A} = \{g : \|\hat{\boldsymbol{\beta}}_g\|_2 \neq 0\}$ is the index set of nonzero groups. Similar to standard variable selection, we can derive strong rules to discard the gth group for group MCP and group SCAD as follows:

$$\text{group MCP} : \left\|\mathbf{X}_g'\mathbf{r}_{k-1}/n\right\|_2 < \sqrt{p_g}\left\{\lambda_k + \frac{\gamma}{\gamma-1}(\lambda_k - \lambda_{k-1})\right\} \tag{20}$$

$$\text{group SCAD} : \left\|\mathbf{X}_g'\mathbf{r}_{k-1}/n\right\|_2 < \sqrt{p_g}\left\{\lambda_k + \frac{\gamma}{\gamma-2}(\lambda_k - \lambda_{k-1})\right\}. \tag{21}$$

Although we framed this derivation in the linear regression setting, note that these rules apply to the generalized linear model case as well, provided that the link function is included in the calculation of $\mathbf{r}_{k-1}$, as in Section 3.2.

4 Simulation study

In this section, we carry out a more thorough investigation of the illustration presented in Figure 1 in terms of the frequency of strong rule violations. We first consider linear and logistic regression for MCP and SCAD. The simulation design follows the description in Section 2 for the linear regression case; for logistic regression, the design is the same except that y_i follows a Bernoulli distribution with the logistic link function and the nonzero regression coefficients equal ±0.5. For 100 independently generated data sets, we record the number of eliminated variables, the number of λ values at which a violation occurred, and the total number of erroneously discarded variables. For the violations, we also record whether the violation occurred in the locally convex region of the solution path or not.

Table 1: Simulation results for MCP and SCAD strong rules with $n = 200$ and $p = 2,000$. Results averaged over 100 independent data sets.

Model	*Method*	ρ	*Average # of eliminated variables*	*Number of violated λ values*	*Number of violated variables*	*Number of violated λ (convex region)*
Linear regression	MCP	0	1971.17	1.23	4.13	0.01
		0.5	1973.76	6.28	12.74	0.01
	SCAD	0	1958.19	0.16	0.62	-
		0.5	1958.77	7.69	36.37	-
Logistic regression	MCP	0	1970.81	2.36	6.37	-
		0.5	1982.41	2.23	3.98	-
	SCAD	0	1935.91	3.72	18.23	-
		0.5	1966.88	4.51	16.96	-

Table 1 presents the results of this simulation, averaged over the 100 replications. Overall, the table reflects the earlier observations made concerning Figure 1. The strong rules discard a large proportion of variables in the inactive sets and thereby achieve considerable dimension reduction in $p > n$ problems. The rules are not foolproof: violations occur regularly, and the problem is exacerbated by high correlation among the covariates, although logistic regression is less sensitive to correlation than is linear regression.

Nevertheless, violations only occur at a small number of the 100 λ values along the solution path, and almost always occur in the non-locally convex region. For example, in the linear regression case with $\rho = 0$, a single violation in 100 data sets was observed for MCP in the locally convex region, which accounts for less than 1% of the observed violations. In many of the scenarios, no violations in the locally convex region occurred.

We also study the performance of strong rules for group variable selection using group MCP and group SCAD using the following simulation design. The covariates follow a standard Gaussian distribution with a block-diagonal correlation structure such that within-block correlation is 0.5. The design matrix consists of 500 groups (blocks), each with 4 elements. The coefficients for the first 6 groups are equal to ± 1 for linear regression and ± 0.5 for logistic regression; the coefficients in the other 494 groups are all zero. We fixed the sample size at 200 (i.e., $n = 200$, $G = 500$ and $p = 2,000$).

Table 2: Simulation results for group MCP and group SCAD strong rules with $n = 200$, $G = 500$ and $p = 2,000$. Within-block correlation ρ is 0.5. Results averaged over 100 independent data sets.

Model	*Method*	*Average # of eliminated groups*	*Number of violated λ values*	*Number of violated groups*	*Number of violated λ (convex region)*
Linear regression	gMCP	492.99	-	-	-
	gSCAD	492.44	-	-	-
Logistic regression	gMCP	492.07	0.11	0.11	0.01
	gSCAD	487.30	1.12	1.78	-

Table 2 shows the number of discarded groups, as well as the number of strong rule violations, averaged over 100 independent data sets. As in the non-grouped case, violations occur only for a small fraction of the 100 λ values along the solution path, and almost always in the non-locally convex region.

5 Incorporation of strong rules into model-fitting algorithms

In this section, we discuss the incorporation of strong rules into the coordinate descent (CD) algorithm of Breheny and Huang (2011) for fitting nonconvex penalized regression models. The algorithms we propose may be viewed as modifications of the idea behind coordinate descent: rather than cycling over the full set of variables with every iteration, the availability of strong rules and other heuristics allow one to carry out *targeted cycling* in which computational effort is concentrated on the variables most likely to be nonzero and therefore change from one λ value to the next. We consider three targeted cycling algorithms: one based on strong rules, one based on active set cycling, and a hybrid algorithm combining the two heuristics.

Algorithm 1 describes the incorporation of strong rules into the coordinate descent algorithm; we refer to this approach as the *strong rule algorithm*. The algorithm relies on computing the *strong set* $\mathcal{S}(\lambda)$, which we define as the set of variables remaining after discarding variables according to the strong rules

Algorithm 1 Dimension reduction using strong rules for targeted cycling

```
for $k = 1, 2, \ldots, m$
    Calculate the strong set $\mathcal{S}(\lambda_k)$ and let $\mathcal{T} = \mathcal{S}(\lambda_k)$
    repeat
        Find the solution $\hat{\boldsymbol{\beta}}_{\mathcal{T}}(\lambda_k)$ using only the variables in $\mathcal{T}$
        Find $\mathcal{V} = \{j \in \mathcal{T}^c : |\mathbf{x}_j'\mathbf{r}/n| \geq \lambda_k\}$
        Update $\mathcal{T}$ by $\mathcal{T} \cup \mathcal{V}$
    until $\mathcal{V} = \emptyset$
```

(10, 11), and then using this set as the target set $\mathcal{T}$ that we cycle over until convergence. As discussed previously, it is possible for strong rules to be violated, and therefore necessary to calculate the set of violations $\mathcal{V}$ in order to ensure that all solutions $\hat{\boldsymbol{\beta}}$ satisfy the KKT conditions at convergence.

An alternative approach is to use the active set $\mathcal{A}(\lambda_{k-1})$ as the target set for calculating the solution at the next step in the solution path, $\hat{\boldsymbol{\beta}}(\lambda_k)$. The algorithm, which we refer to as *active set cycling* (Friedman et al., 2010), is the same as Algorithm 1 with the active set $\mathcal{A}(\lambda_{k-1})$ replacing the strong set $\mathcal{S}(\lambda_k)$.

Algorithm 2 Dimension reduction using a hybrid of strong rules and active set cycling

```
for $k = 1, 2, \ldots, m$
    Set $\mathcal{T} = \mathcal{A}(\lambda_{k-1})$ and $\mathcal{S} = \mathcal{S}(\lambda_k)$
    repeat
        repeat
            Find the solution $\hat{\boldsymbol{\beta}}_{\mathcal{T}}(\lambda_k)$ using only the variables in $\mathcal{T}$
            Find $\mathcal{V}_1 = \{j \in \mathcal{S} \setminus \mathcal{T} : |\mathbf{x}_j'\mathbf{r}/n| \geq \lambda_k\}$
            Update $\mathcal{T}$ by $\mathcal{T} \cup \mathcal{V}_1$
        until $\mathcal{V}_1 = \emptyset$
        Find $\mathcal{V}_2 = \{j \in \mathcal{T}^c \setminus \mathcal{S} : |\mathbf{x}_j'\mathbf{r}/n| \geq \lambda_k\}$
        Update $\mathcal{T}$ by $\mathcal{T} \cup \mathcal{V}_2$
    until $\mathcal{V}_2 = \emptyset$
```

The final approach we consider combines the active set and strong sets into an algorithm that involves two-stage targeted cycling. The details are provided in Algorithm 2, which we refer to as the *hybrid algorithm*.

Contrasting the four algorithms (cyclic, strong, active, and hybrid), there is a tradeoff between how aggressive the algorithms are in terms of discarding variables and how often violations involving erroneously discarded variables occur. Discarding variables naturally increases the speed of optimization over the target set; however, violations introduce a computational cost as well, since the iterative targeted cycling procedure must be restarted and the KKT conditions re-checked. At one extreme, active set cycling discards the largest number of variables, but its targeted cycling rule is violated every time a new variable enters the active set. On the other extreme, full cyclic coordinate descent does not have to contend with violations or re-check any KKT conditions, but must contend with the full set of variable at every step. The strong and hybrid algorithms attempt to occupy a middle ground between these two extremes, reducing dimensionality as much as possible without introducing a large number of violations.

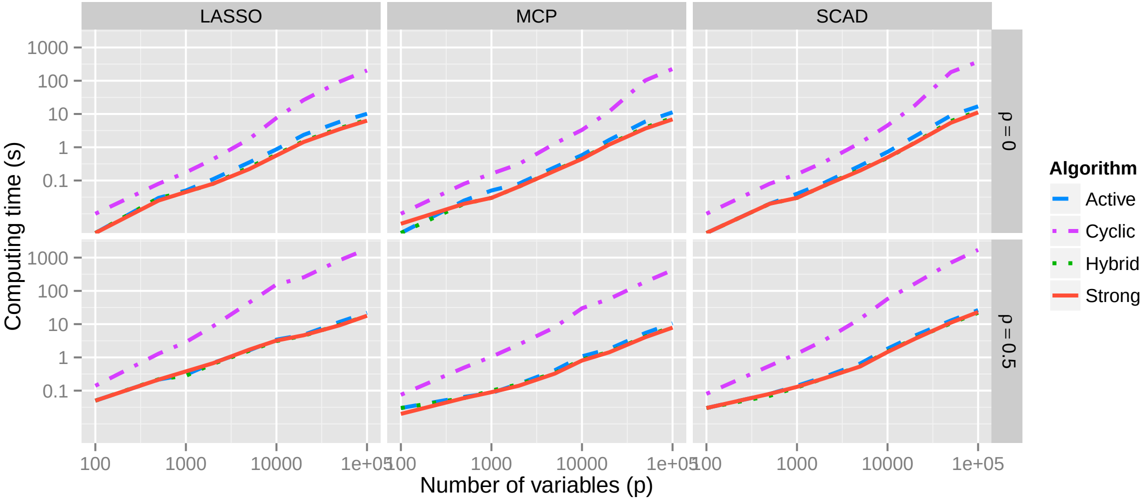

Figure 3: Comparison of the cyclic CD algorithm and targeted cycling algorithms in terms of computing time required to fit the entire coefficient path down to $\lambda_{\min}/\lambda_{\max} = 0.05$ for linear regression as a function of the number of covariates p. Both axes are on the log scale. Median times over 20 replications are displayed.

Figure 3 demonstrates that all three targeted cycling algorithms are considerably faster than full cyclic coordinate descent, and that the magnitude of the difference is substantial for high dimensional problems. For example, the median time required to fit a SCAD model with $\rho = 0.5$ and $p = 100,000$ was 1,711 seconds using cyclic coordinate descent and just 23 seconds using the strong rule algorithm. It is worth noting that even though strong rules are more likely to be violated as correlation increases, the fact that optimization algorithms must go through a larger number of iterations in this case results in an even greater advantage for targeted cycling in the correlated case than in the uncorrelated case.

In Figure 3, it is clear that targeted cycling is more efficient than full cyclic CD, but it is unclear how the target cycling algorithms compare to each other. In Figure 4, we compare the speed of the three targeted cycling algorithms for linear, logistic, and Poisson regression. In each case, 20 variables are set

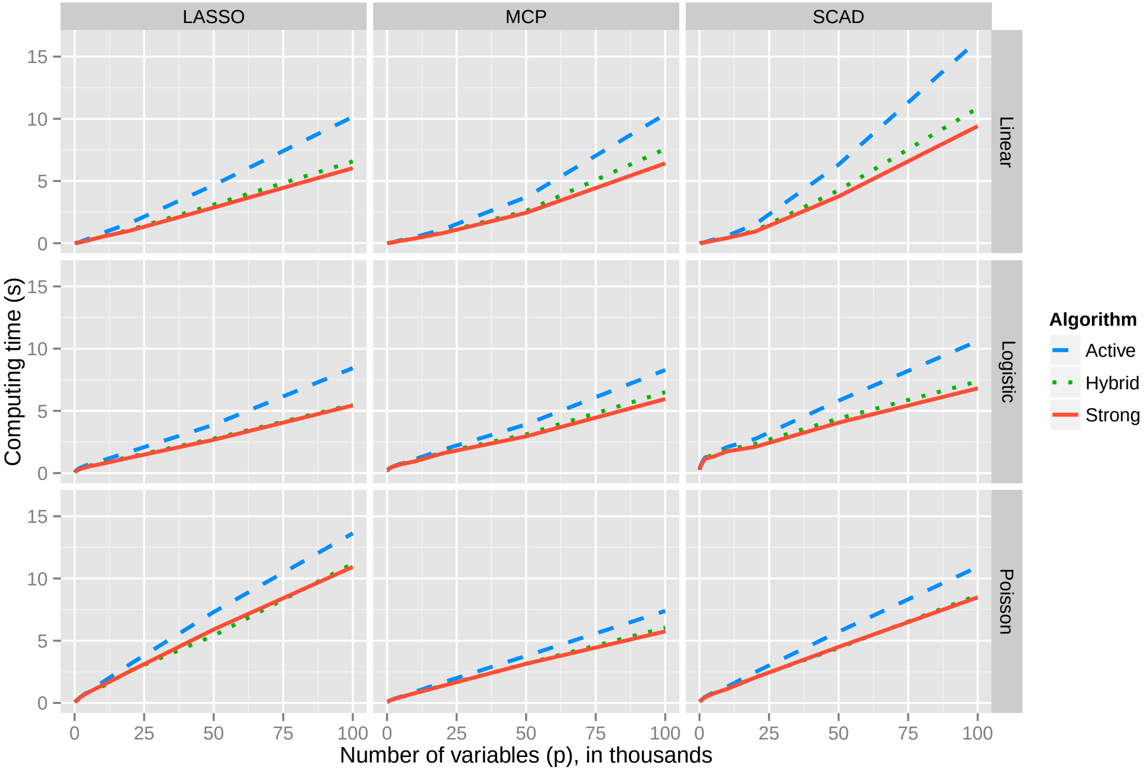

Figure 4: Comparison of targeted cycling algorithms in terms of computational time required to fit the entire coefficient path for linear (top), logistic (middle) and Poisson regression (bottom). Median computing times over 20 replications are displayed.

to ±0.5 with the remaining variables set to zero, and the outcome follows the distribution assumed in the GLM. In all cases, the strong rule and hybrid algorithms were seen to be more efficient than active set cycling. Although the difference in computing time between the targeted cycling algorithms is minor for small p, active set cycling can take almost twice as long for high-dimensional models. For example, fitting a SCAD-penalized logistic regression model with $p = 100,000$ required a median computing time of 11 seconds for active cycling and only 7 seconds for the strong rule and hybrid algorithms.

Although the strong rule algorithm was slightly faster than the hybrid algorithm in Figure 4, we have found that there are situations in which the hybrid algorithm offers considerably better performance than the strong rule approach. We depict one such situation for linear regression in Figure 5. The top panel ("Case 1") of Figure 5 is similar to the situations we have examined so far, with $n = 200$, $p = 20,000$, $\rho = 0$, and 20 nonzero coefficients equal to ± 1. Here, the variance of Gaussian noise was chosen so that the signal-to-noise ratio was equal to 3. The setting for the bottom panel ("Case 2") is the same, except that the nonzero coefficients all have coefficients equal to $+1$. In Case 1, the size of the target set for

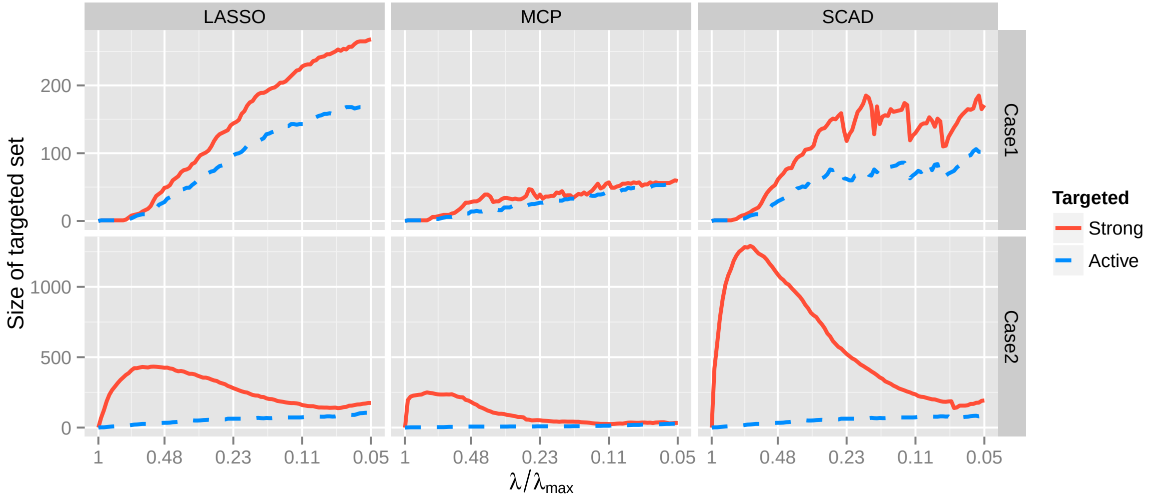

Figure 5: Comparison of the size of strong and active sets for each λ for simulated Gaussian data with $\rho = 0$, $n = 200$ and $p = 20,000$. In "Case 1" (top panel), nonzero coefficients are equal to ±1; in "Case 2" (bottom panel), all nonzero coefficients equal +1.

the strong rule algorithm matches the active set quite closely, and nearly all the variables that can be eliminated are eliminated by the strong rules. In Case 2, however, although the strong rules are still valid and rarely violated, they do not yield a target set that closely matches the active set, and fail to discard hundreds of variables that remained inactive.

In Case 1, there were minimal differences between the computing time of the three targeted cycling algorithms (all were within 1 second of each other). In Case 2, however, the strong rule algorithm was substantially slower than active cycling and the hybrid algorithm due to the much larger size of its target set. For SCAD, where the difference in target set size was most dramatic, the strong rule algorithm required 19 seconds, while active cycling required just 5. As it is designed to do, the hybrid algorithm utilizes the best features of each heuristic and requires just 4 seconds to compute the solution path.

In summary, we find the hybrid algorithm to be the most robust of the targeted cycling approaches – never much slower than the strong rule algorithm, and in some cases much faster. For this reason, we have implemented the hybrid algorithm for lasso, SCAD, and MCP-penalized linear, logistic, and Poisson regression in the `ncvreg` package.

6 Application to genome-wide association studies

In this section, we apply the algorithms described in Section 5 to real data from a genome-wide association study (GWAS) of preeclampsia. The data were collected during the Study of Pregnancy Hypertension in

Iowa (SOPHIA), a population-based case-control study. We provide a brief description of the data here; the study is described in greater detail in Zhao et al. (2012).

The sample consists of 177 mothers diagnosed with preeclampsia according to National Heart, Lung and Blood Institute guidelines and 115 mothers with normal blood pressure to serve as controls. All 292 mothers were genotyped using the Affymetrix Genome-Wide Human SNP Array 6.0 (Affymetrix, Santa Clara, CA). After applying quality control procedures and eliminating monomorphic markers, we were left with 810,198 single-nucleotide polymorphisms (SNPs) to serve as potential predictors of preeclampsia risk.

We analyzed this data using MCP-penalized logistic regression with case-control status as the response variable. Allele effects were assumed to be additive and independent, thereby yielding a design matrix with $n = 292$ and $p = 810,198$. Due to the fact that $p \gg n$, we fit the penalized regression model over a relatively small portion of the coefficient path, down to $\lambda_{\min}/\lambda_{\max} = 0.8$, at which point 22 SNPs had entered the model. Despite the large number of features in the design matrix, the penalized regression models could be fit very rapidly: using the strong rule algorithm, the solution path could be fit in just 4.7 seconds on a standard desktop computer (3.60GHz Intel Xeon processor, 16 GB RAM). The active cycling algorithm took somewhat longer, at 7.9 seconds, while the performance of the hybrid algorithm was similar to that of the strong rule approach (4.9 seconds to fit the solution path).

The SNPs selected by the penalized regression model are consistent with the top-ranked SNPs in terms of univariate hypothesis testing using Fisher's exact test, as reported in Zhao et al. (2012). However, we reach the same conclusion that the authors of the previous study reached – namely, that there is insufficient evidence in the data to perform variable selection with any meaningful degree of reliability. In particular, when we carry out 10-fold cross-validation for the purposes of selecting λ, we find that the optimal model is the intercept-only model.

Although this particular study was negative in terms of identifying genetic risk factors for preeclampsia, it illustrates the feasibility of fitting penalized regression models to very high-dimensional data. The current genome-wide association literature is overwhelmingly focused on univariate tests, which have many shortcomings compared to multivariate modeling: inefficiency, increased risk of confounding, and limited predictive inference, among others. Several authors have recommended penalized regression as an alternative, and discussed its benefits in comparison with univariate testing (Zhou et al., 2010; Wu et al., 2009). Others, however, have judged the problem to be computationally impractical for the very high dimensions that prevail at the genome-wide scale and developed multi-stage or iterative screening proposals to reduce the dimensionality of the problem (Fan and Lv, 2008; Shi et al., 2011; Zhao and Chen, 2012). We demonstrate here that such approaches are not necessary – or, depending on your perspective, that screening is indeed a very useful idea, but it can be incorporated directly into coordinate descent algorithms through targeted cycling.

7 Discussion

Concern over the computational burden of penalized regression in very high dimensions has prevented its use in many fields, particularly in genetics. This concern, in turn, has led many researchers to pre-screening procedures to reduce the dimensionality of the problem before fitting the penalized regression model. At best, this complicates both the theoretical study of such procedures and the practical implementation of procedures such as cross-validation. At worst, it opens the door for bad statistical practice by obfuscating the multiple comparison problem. For example, if pre-screening is used to select candidate variables on the full data set, and then cross-validation is used to select a tuning parameter λ, the resulting inference is heavily biased by the fact that the external validation data is not truly external, as it has already been used for screening.

It is possible to carry out unbiased cross-validation in the presence of screening, but it is also very easy for a well-intentioned investigator to make a mistake (a thorough discussion of this issue may be found in Hastie et al., 2009). In contrast, cross-validation is both straightforward and computationally feasible, and already implemented existing software such as `glmnet` and `ncvreg`. In particular, for the analysis in Section 6, ten-fold cross validation was carried out in under a minute despite fitting nonconvex penalized logistic regression models with $p = 810,198$ variables.

With this work, we have demonstrated that fitting high-dimensional nonconvex penalized regression models can be made computationally feasible through the use of targeted cycling and strong rules to achieve dimension reduction. Furthermore, by sharing implementations of these algorithms in the publicly available `R` package `ncvreg`, we hope to encourage researchers to adopt these methods with greater regularity for analyzing high-dimensional data.

References

Breheny, P. and Huang, J. (2011). Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *The Annals of Applied Atatistics*, **5** 232.

Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004). Least angle regression. *Annals of Statistics*, **32** 407–451.

El Ghaoui, L., Viallon, V. and Rabbani, T. (2011). Safe feature elimination for the lasso and sparse supervised learning problems. *arXiv preprint arXiv:1009.4219.*

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

Fan, J. and Lv, J. (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society Series B*, **70** 849–911.

Friedman, J., Hastie, T., Höfling, H. and Tibshirani, R. (2007). Pathwise coordinate optimization. *Annals of Applied Statistics*, **1** 302–332.

Friedman, J. H., Hastie, T. and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, **33** 1–22.

Hastie, T., Tibshirani, R. and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

Huang, J., Breheny, P., Lee, S., Ma, S. and Zhang, C.-H. (2013). Balancing stability and bias reduction in variable selection with the mnet estimator.

Kim, Y., Choi, H. and Oh, H.-S. (2008). Smoothly clipped absolute deviation on high dimensions. *Journal of the American Statistical Association*, **103** 1665–1673.

Shi, G., Boerwinkle, E., Morrison, A. C., Gu, C. C., Chakravarti, A. and Rao, D. (2011). Mining gold dust under the genome wide significance level: a two-stage approach to analysis of gwas. *Genetic Epidemiology*, **35** 111–118.

Simon, N. and Tibshirani, R. (2011). Standardization and the group lasso penalty. *Statistica Sinica*, **22** 983–1001.

Tibshirani, R., Bien, J., Friedman, J., Hastie, T., Simon, N., Taylor, J. and Tibshirani, R. J. (2012). Strong rules for discarding predictors in lasso-type problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **74** 245–266.

Wu, T., Chen, Y., Hastie, T., Sobel, E. and Lange, K. (2009). Genome-wide association analysis by lasso penalized logistic regression. *Bioinformatics*, **25** 714.

Wu, T. and Lange, K. (2008). Coordinate descent algorithms for lasso penalized regression. *Annals of Applied Statistics*, **2** 224–244.

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

ZHANG, C. (2010). Nearly unbiased variable selection under minimax concave penalty. *Annals of Statistics*, **38** 894–942.

ZHAO, J. and CHEN, Z. (2012). A two-stage penalized logistic regression approach to case-control genome-wide association studies. *Journal of Probability and Statistics*, **2012** 1–15.

ZHAO, L., TRICHE, E., WALSH, K., BRACKEN, M., SAFTLAS, A., HOH, J. and DEWAN, A. (2012). Genome-wide association study identifies a maternal copy-number deletion in psg11 enriched among preeclampsia patients. *BMC Pregnancy and Childbirth*, **12** 61.

ZHOU, H., SEHL, M., SINSHEIMER, J. and LANGE, K. (2010). Association screening of common and rare genetic variants by penalized regression. *Bioinformatics*, **26** 2375.

ZOU, H. and HASTIE, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B*, **67** 301–320.

ZOU, H. and LI, R. (2008). One-step sparse estimates in nonconcave penalized likelihood models. *Annals of statistics*, **36** 1509.