Strong rules for discarding predictors in lasso-type problems

Robert Tibshirani[*], Ryan J. Tibshirani[†], Jerome Friedman [‡], and Trevor Hastie [§]

November 11, 2010

Abstract

We consider rules for discarding predictors in lasso regression and related problems, for computational efficiency. El Ghaoui et al. (2010) propose "SAFE" rules that guarantee that a coefficient will be zero in the solution, based on the inner products of each predictor with the outcome. In this paper we propose *strong rules* that are not foolproof but rarely fail in practice. These can be complemented with simple checks of the Karush-Kuhn-Tucker (KKT) conditions to provide safe rules that offer substantial speed and space savings in a variety of statistical convex optimization problems.

1 Introduction

We consider the problem of prediction using a linear model, with ℓ_1-type regularization. In particular we consider a problem with N observations and p predictors. Denote by $\mathbf{y}$ the N-vector of predictors, and let $\mathbf{X}$ be the $N \times p$ matrix of predictors with jth column $\mathbf{x}_j$ and ith row x_i. We assume that the predictors and outcome are centered, and so we can omit an intercept from the model.

The lasso (Tibshirani 1996) minimizes the criterion

$$\frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2 + \lambda||\boldsymbol{\beta}||_1 \tag{1}$$

where $\lambda \geq 0$ is a tuning parameter. There has been considerable work in the past few years deriving fast algorithms for solving this problem, for large values of N and p. Some methods like coordinate descent deliver the solution over a grid of λ values, using warm starts along the way. This is implemented for example in the `glmnet` package (Friedman et al. 2008).

Wu et al. (2009) proposed screening rules for penalized logistic regression based on the inner products $|\mathbf{x}_j^T\mathbf{y}|$. A lasso fit is computed using a small selected subset of the predictors with largest inner products, and then the KKT conditions are checked for violations. This same predictor ordering is proposed and studied by Fan & Lv (2008) in their "Sure Independence Screening" method. El Ghaoui et al. (2010) use a clever argument to derive a surprising rule for discarding predictors in the lasso and related problems. Their "SAFE" rule discards predictors if $|\mathbf{x}_j^T\mathbf{y}|$ is less that a certain bound that depends on λ. They prove that their rule is safe, that is, if the predictor is discarded then $\hat{\beta}_j$ is guaranteed to be zero for the lasso solution at λ. They show that the SAFE rule can save both time and memory in the overall computation.

In this paper we propose *strong rules* for discarding predictors. These rules discard more predictors than the SAFE rules, but are not foolproof and can fail in practice. However these failures are rare and the new rules can be combined with simple checks of the Karush-Kuhn-Tucker (KKT) conditions to provide

[*] Departments of Health, Research & Policy, and Statistics, Stanford University, tibs@stanford.edu

[†] Department of Statistics, Stanford University, ryantibs@stanford.edu

[‡] Department of Statistics, Stanford University, jhf@stanford.edu

[§] Departments of Statistics, and Health, Research & Policy, Stanford University, hastie@stanford.edu safe rules. As a result, they offer a substantial speed and space savings in a variety of statistical convex optimization problems.

In Section 2 we review the SAFE rules of El Ghaoui et al. (2010). The *strong rules* are introduced and studied in Section 3, for the lasso and elastic net, and a condition for exactness of the rule is also given. We discuss logistic regression in Section 4. A form of the strong rules for general convex optimization problems is given in Section 5 and applied to the graphical lasso method. We discuss and illustrate implementation of the sequential strong rule in our `glmnet` algorithm in Section 6 and finally Section 7 contains some final discussion.

2 A review of SAFE rules for discarding variables

The basic SAFE rule of El Ghaoui et al. (2010) is defined as follows: fitting at λ, can discard predictor j if

$$|\mathbf{x}_j^T\mathbf{y}| < \lambda - ||\mathbf{y}||||\mathbf{x}_j||\frac{\lambda_{max}-\lambda}{\lambda_{max}} \tag{2}$$

where $\lambda_{max} = \max_j|\mathbf{x}_j^T\mathbf{y}|$, is the smallest λ at for which all coefficients are zero. This is the basic SAFE bound. They also derive a more complicated, somewhat better bound called "recursive SAFE" (RECSAFE).

The (basic) SAFE bound is derived by looking at a dual of the lasso. Here is a sketch of their argument. One version of the dual problem is to maximize

$$G(\boldsymbol{\theta}) = \mathbf{y}^T\mathbf{y}/2 - (\mathbf{y}+\boldsymbol{\theta})^T(\mathbf{y}+\boldsymbol{\theta})/2 \tag{3}$$

subject to $|\mathbf{x}_j^T\boldsymbol{\theta}| \leq \lambda\ \forall j$. The relationship between the primal and dual variables at the solution is $\hat{\boldsymbol{\theta}} = \mathbf{X}\hat{\boldsymbol{\beta}} - \mathbf{y}$. They find a dual feasible point of the form $\boldsymbol{\theta} = s\mathbf{y}$, ($s$ is a scalar) and hence $\gamma = G(s\mathbf{y})$ represents a lower bound for the value of G at the solution. Then for each predictor j they find the maximum m_j of $|\boldsymbol{\theta}^T\mathbf{x}_j|$ over the set $G(\boldsymbol{\theta}) \geq \gamma$; if this maximum is less than λ, this tells us that the quantity $|\boldsymbol{\theta}^T\mathbf{x}_j|$ must be $< \lambda$ at the solution, i.e. β_j must be zero at the solution. Hence we can discard predictor j. Finally, rewriting the condition $m_j < \lambda$ yields condition (2).

Figure 1 shows some examples. There are four scenarios with various values of N and p; in the first three panels, the $\mathbf{X}$ matrix is dense, while it is sparse in the bottom right panel. The population correlation among the feature is zero, positive, negative and zero in the four panels. Finally, 25% of the coefficients are non-zero, with a standard Gaussian distribution. In the plots, we are fitting along a path of decreasing λ values and the plots show the number of predictors left after screening at each stage. We see that the SAFE and RECSAFE bounds only exclude predictors near the beginning of the path. The "strong" rules (orange and red lines) are discussed in section 3.

The RECSAFE method uses the solution at a given point λ_0 to derive a rule for discarding predictors at $\lambda < \lambda_0$. Here is another way to (potentially) apply the SAFE rule in a sequential manner. Suppose that we have $\hat{\boldsymbol{\beta}}_0 = \hat{\boldsymbol{\beta}}(\lambda_0)$, and $\mathbf{r} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_0$, and we consider the fit at $\lambda < \lambda_0$, with $\mathbf{r} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_0$. Defining

$$\lambda_0 \quad = \quad \max_j(|\mathbf{x}_j^T\mathbf{r}|); \tag{4}$$

we discard predictor j if

$$|\mathbf{x}_j^T\mathbf{r}| < \lambda - ||\mathbf{r}||||\mathbf{x}_j||\frac{\lambda_0-\lambda}{\lambda_0} \tag{5}$$

We have been unable to prove the correctness of this rule, and do not know if it is infallible. At the same time, we have been not been able to find a numerical example in which it fails.

3 Strong screening rules

Here we propose some alternative screening rules in the lasso setting, and then consider extensions to the elastic net.

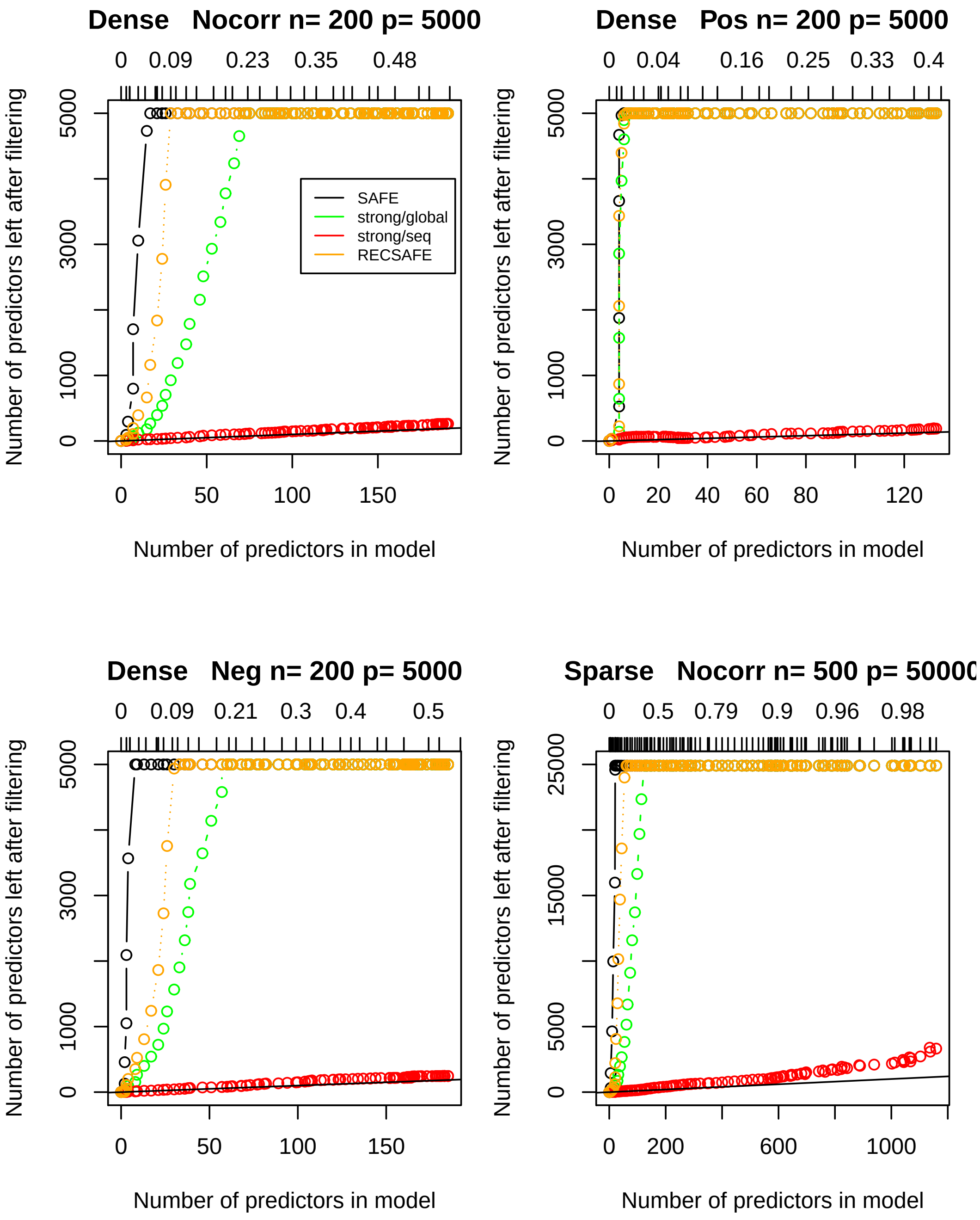

Figure 1: *Lasso regression: results of different discarding rules applied to four different scenarios. There are four scenarios with various values of N and p; in the first three panels the $\mathbf{X}$ matrix is dense, while it is sparse in the bottom right panel. The population correlation among the feature is zero, positive, negative and zero in the four panels. Finally, 25% of the coefficients are non-zero, with a standard Gaussian distribution. In the plots, we are fitting along a path of decreasing λ values and the plots show the number of predictors left after screening at each stage. The proportion of variance explained by the model is shown along the top of the plot is shown.*

3.1 The lasso

First note that the subgradient equation for the lasso is

$$\mathbf{X}^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) = \lambda \cdot \mathbf{s}(\lambda) \tag{6}$$

where $\mathbf{s}(\lambda) = (s_1(\lambda), s_2(\lambda), \dots s_p(\lambda))$ and $s_j(\lambda) = \mathrm{sign}(\beta_j(\lambda))$ if $\beta_j(\lambda) \neq 0$ and $s_j(\lambda) \in [-1,1]$ otherwise. Let $\lambda_{max} = \max_j |\mathbf{x}_j^T\mathbf{y}|$ and $c_j(\lambda) = \mathbf{x}_j^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}(\lambda))$.

Suppose that predictor j is not in the model at λ_{max} so that $|c_j(\lambda_{max})| < \lambda_{max}$. Consider the slope $dc_j(\lambda)/d\lambda$. For an active variable $c_j(\lambda) = \lambda$ and hence $|dc_j(\lambda)/d\lambda| = 1$. Suppose that in general we could assume that

$$\Big|\frac{dc_j(\lambda)}{d\lambda}\Big| \leq 1 \tag{7}$$

for all variables, active and inactive, except for the finite set of λ values where $dc_j(\lambda)/d\lambda$ is not differentiable. This is plausible since

$$\frac{dc_j(\lambda)}{d\lambda} = s_j(\lambda) + \lambda \cdot \frac{ds_j(\lambda)}{d\lambda} \tag{8}$$

For variables that remain active at λ $ds_j(\lambda)/d\lambda = 0$; if we can ignore the second term above for all variables, then (7) would follow since $|s_j(\lambda)| \leq 1$ Note also in the orthonormal design case ($\mathbf{X}^T\mathbf{X} = \mathbf{I}$), it is easy to show that $dc_j(\lambda)/d\lambda = -1, +1$, or 0, where this derivative exists.

Hence assume for the moment that (7) holds. We know that predictor j enters the model at λ if

$$c_j(\lambda) = c_j(\lambda_{max}) + [c_j(\lambda) - c_j(\lambda_{max})] = \lambda \tag{9}$$

Since $|\frac{dc_j(\lambda)}{d\lambda}| \leq 1$, we have

$$\begin{aligned} |c_j(\lambda) - c_j(\lambda_{max})| &= |\int_{\lambda_{max}}^{\lambda} \frac{dc_j(\lambda)}{d\lambda} d\lambda| \\ &\leq \int_{\lambda_{max}}^{\lambda} |\frac{dc_j(\lambda)}{d\lambda}| d\lambda \\ &= \lambda - \lambda_{max}. \end{aligned} \tag{10}$$

Here we have integrated over the piecewise linear segments between the points of discontinuity. Therefore we can discard predictor j if

$$|\mathbf{x}_j^T\mathbf{y}| < \lambda - (\lambda_{max} - \lambda) = 2\lambda - \lambda_{max}. \tag{11}$$

We call this the *strong* screening rule. Figure 2 illustrates the SAFE and strong rules in an example. It shows the inner product of the active and inactive predictors with the residual as λ decreases. If we can assume that the absolute slope of each inner-product curve is at most one, then we can bound the amount that any such inner-product rises as we move from λ_{max} to the value λ. Hence if the initial inner-product starts too far below the maximal achieved inner-product, then it can't "catch up" in time.

It turns out be more effective to apply the strong rule sequentially. Suppose that we have a solution at λ_0 yielding solution $\hat{\boldsymbol{\beta}}(\lambda_0)$ and residual $\mathbf{r} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}(\lambda_0)$, and wish to discard predictors for a fit at $\lambda < \lambda_0$. The sequential version of the strong rule (11) is to discard predictor j if

$$|\mathbf{x}_j^T\mathbf{r}| < 2\lambda - \lambda_0 \tag{12}$$

We call this the *strong sequential* rule. Note that as $\lambda_0 \to \lambda$, the sequential rule becomes

$$|\mathbf{x}_j^T\mathbf{r}| < \lambda \tag{13}$$

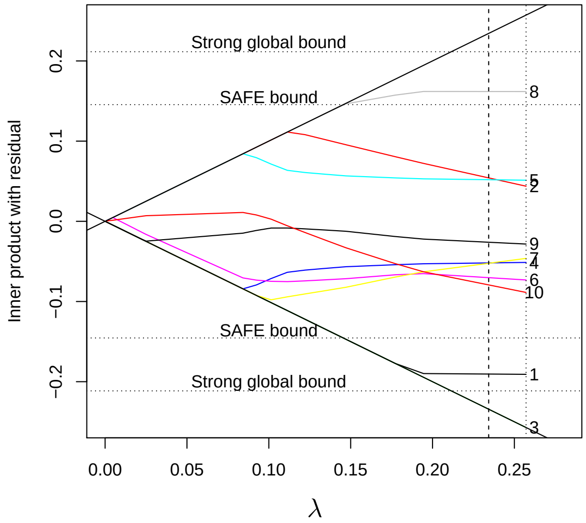

Figure 2: *SAFE and strong global bounds in an example with 10 predictors. The dotted vertical line is drawn at λ_{max}; the broken vertical line is drawn at λ. The strong rule keeps only predictor #3, while the SAFE bound keeps predictors #8 and #1 as well.*

This is just the KKT condition for excluding a variable in the solution at λ. Hence in effect the sequential rule (12) at λ_0 "anticipates" the KKT conditions at λ.

To compare the SAFE and strong rules, note that the SAFE bound (2) is unchanged if we standardize the predictors and response. Then comparing (2) to (11) in the standardized case (i.e the outcome and each feature having unit norm), since $(\lambda_{max} - \lambda)/\lambda_{max} \geq (\lambda_{max} - \lambda)$, we have

$$2\lambda - \lambda_{max} \geq \lambda - (\lambda_{max} - \lambda)/\lambda_{max}. \tag{14}$$

Therefore the bound in (11) is larger than that in (2) and will discard more predictors.

There is another way to view the SAFE bound (2). With standardized data, the bound says that the inner product $\mathbf{x}_j^T\mathbf{r}$ cannot change more than $(\lambda_{max} - \lambda)/\lambda_{max}$, as we move from λ_{max} to λ. This means that the average slope of the inner product $\mathbf{X}_j^T\mathbf{r}$ is bounded by $1/\lambda_{max}$ in absolute value. Since the data are standardized, $\lambda_{max} \leq 1$ and so this bound on the slope is ≥ 1. Thus it is perhaps not surprising that an upper bound for the slopes of one does not hold in general, as we show next.

3.2 Violation of the slope condition

It turns out that the key slope condition (7) is nearly true, but can be violated for short stretches, especially when $p \approx N$ and for small values of λ in the "overfit" regime of a lasso problem. Figure 3 shows an example where it is violated. Using this kind of example, we can easily construct a problem in which rule (12) erroneously discards predictors. We suspect that a counter-example for the global rule (11) can also be constructed but have not yet found one. Such a counter example would require that the average slope exceed one from λ_{max} to λ, rather than just for a short stretch of λ values.

In section 3.4 we derive a condition for the data matrix $\mathbf{X}$ under which (7) is guaranteed to hold; but this condition will not be true in general. Nonetheless the strong rules (11) and (12) can very useful in practice. In fact, the rules never made an error in any of the numerical examples in this paper, and hence the name *strong.*

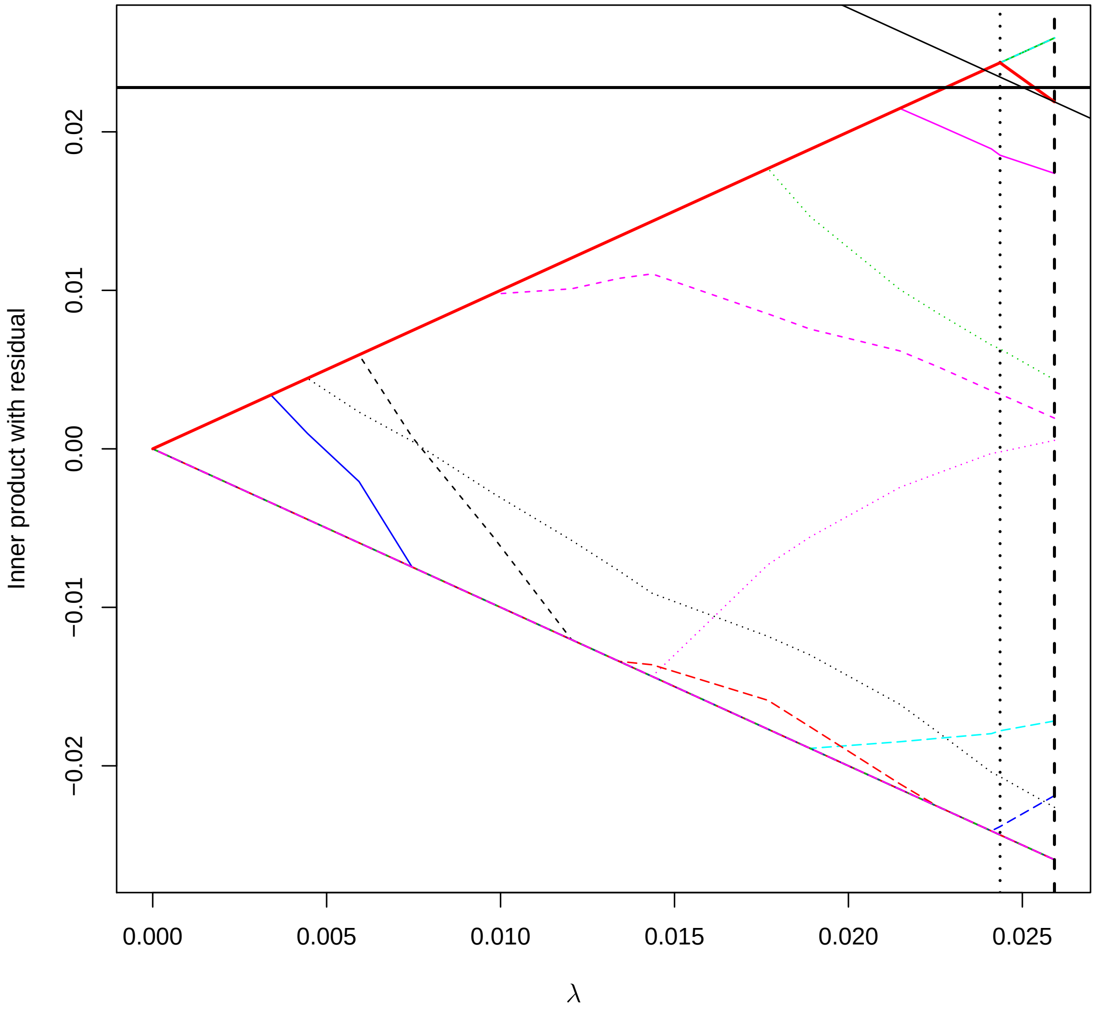

Figure 3: *Example of a violation in the sequential rule (12), where absolute value of correlation slope exceeds one. The data were generated as standard Gaussian* $N = 100$, $p = 80$ *and no signal. The long red lines are the envelop of maximal inner products achieved by predictors in the model for each* λ. *For clarity we only show the profiles for a subset of the predictors. The vertical broken line is drawn at* λ_0, *and we are considering the new value* $\lambda < \lambda_0$ *(dotted vertical line). The horizontal black line is the bound (12). In the top right part of the plot, inner product for the predictor depicted in red starts below the bound but enters the model at* λ. *The slope of the red segment between* λ_0 *and* λ *exceeds one. A black line with slope -1 is drawn beside the red segment for reference.*

But counter-examples do exist, as shown in Figure 3. Hence to use the strong rules in practice, one has to check for violations. Consider in particular the sequential rule (12). At a given λ we apply this rule and discard the corresponding predictors. We fit on the remaining variables, and then finally check the KKT condition of the resulting solution. If they are satisfied, we are done; otherwise we add the variables that violate the KKT conditions into the current model and refit. In principle we we might have to repeat this sequence many times, although the total number is bounded by the number of predictors that can ever enter the model, which is $\min(N, p)$. We implement a strategy like this in our `glmnet` algorithm in Section 6.

3.3 Example: continued

Figure 1 show the results for approximate global and sequential rules (orange and red lines). There were no violations in any of these figures, that is no predictor was discarded that had a non-zero coefficient at the actual solution. We see that the strong sequential rule performs extremely well, leaving only a small number of excess predictors at each stage. The lack of violations is due to the fact that $p \gg N$: we discuss in Section 3.2. Fortuitously, the large p setting is one where discarding predictors is especially attractive.

3.4 A condition for the unit slope bound

Tibshirani & Taylor (2010) provide a general result that can be used to give the following sufficient condition for the unit slope bound (7). Recall that a matrix $\mathbf{A}$ is diagonally dominant if $|A_{ii}| \geq \sum_{j \neq i} |A_{ij}|$ for all i. Their results give us the following:

Theorem. Suppose $\mathbf{X}$ is $N \times p$ and of full rank, with $N \geq p$. If

$$(\mathbf{X}^T\mathbf{X})^{-1} \text{ is diagonally dominant} \tag{15}$$

then the unit slope condition (7) holds at all points where $c_j(\lambda)$ is differentiable.

Note that (15) is a weaker condition than the positive cone condition used in Efron et al. (2004): the positive cone condition implies (15). A sketch of the proof is given in the Appendix.

One example where diagonal dominance holds is the equi-correlation model where $\text{corr}(X_j, X_k) = r$ for all j, k. Assuming standardized features, the inverse of $\mathbf{X}^T\mathbf{X}$ is

$$(\mathbf{X}^T\mathbf{X})^{-1} = \mathbf{I} \cdot \frac{1}{1-r} - \frac{1}{1-r}\Big(\frac{\mathbf{e}\mathbf{e}^T}{1 + r(p-1)}\Big) \tag{16}$$

where $\mathbf{e}$ is a vector of ones. This is diagonally dominant as long as $r \geq 0$.

Another example is the Haar basis, in which the jth column of $\mathbf{X}$ has the form $I(z_i > t_j)$. Here z_i $i = 1, 2, \ldots N$ is a scalar variable and $t_j, j = 1, 2, \ldots p$ are a set of cutpoints. This arises, for example in the one-dimensional fused lasso where we minimize

$$\frac{1}{2}\sum_{i=1}^{N}(y_i - \beta_i)^2 + \lambda \sum_{i=1}^{N-1} |\beta_{i+1} - \beta_i|. \tag{17}$$

(We have left out the usual $\lambda_2 \sum |\beta_i|$ term, which can be dealt with separately.) If we transform this problem to the parameters $\theta_i = \beta_{i+1} - \beta_i$, we get a lasso problem with design matrix $\mathbf{L}$ being a lower triangular matrix of ones, and $(\mathbf{L}^T\mathbf{L})^{-1}$ is diagonally dominant.

3.5 A numerical investigation of the strong sequential rule violations

We generated Gaussian data with $N = 100$, varying values of the number of predictors p and pairwise correlation 0.5 between the predictors. One quarter of the coefficients were non-zero, with the indices of the nonzero predictors randomly chosen and their values equal to ±2. We fit the lasso for 80 equally spaced values of λ from λ_{max} to 0, and recorded the number of violations of the strong sequential rule. Figure

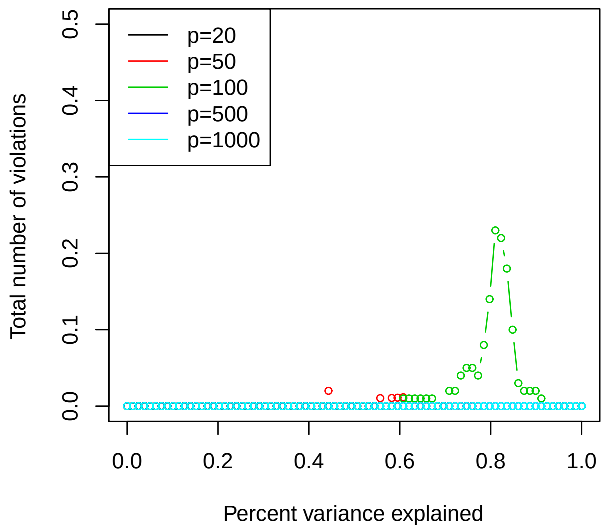

Figure 4: *Total number of violations (out of p predictors) of the strong sequential rule, for simulated data with $N = 100$ and varying values of p. A sequence of models is fit, with decreasing values of λ as we move from left to right. The features are uncorrelated. The results are averages over 100 simulations.*

4 shows the number of violations (out of p predictors) averaged over 100 simulations: we plot versus the percent variance explained instead of λ, since the former is more meaningful. Since the vertical axis is the total number of violations (averaged over 100 simulations), we see that violations are quite rare in general never averaging more than 0.3 out of p predictors. They are more common near the right end of the path. They also tend to occur when p is fairly close to N. When $p \gg N$ ($p = 500$ or 1000 here), there were no violations. Not surprisingly, then, there were no violations in the numerical examples in this paper since they all have $p \gg N$.

Looking at (13), it suggests that if we take a finer grid of λ values, there should be fewer violations of the rule. However we have not found this to be true numerically: the average number of violations at each grid point λ stays about the same.

3.6 Screening rules for the elastic net

In the elastic net we solve the problem [1]

$$\text{minimize } \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2 + \frac{1}{2}\lambda_2||\boldsymbol{\beta}||^2 + \lambda_1||\boldsymbol{\beta}||_1 \tag{18}$$

Letting

$$\mathbf{X}^* = \begin{pmatrix} \mathbf{X} \\ \sqrt{\lambda_2}\cdot\mathbf{I} \end{pmatrix}; \;\; \mathbf{y}^* = \begin{pmatrix} \mathbf{y} \\ 0 \end{pmatrix}, \tag{19}$$

[1] This differs from the original form of the "naive" elastic net in Zou & Hastie (2005) by the factors of 1/2, just for notational convenience.

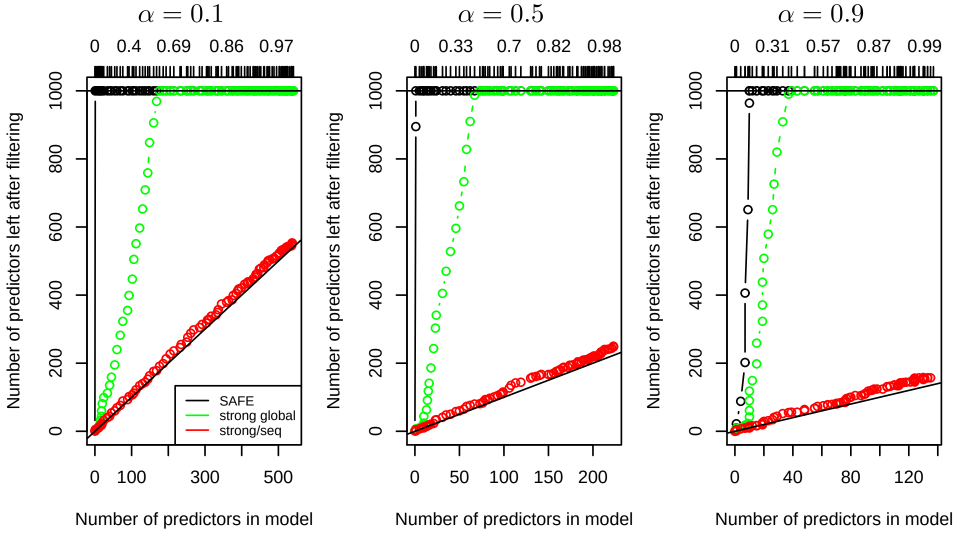

Figure 5: *Elastic net: results for different rules for three different values of the mixing parameter α. In the plots, we are fitting along a path of decreasing λ values and the plots show the number of predictors left after screening at each stage. The proportion of variance explained by the model is shown along the top of the plot is shown.*

we can write (18) as

$$\text{minimize}\frac{1}{2}||\mathbf{y}^* - \mathbf{X}^*\boldsymbol{\beta}||^2 + \lambda_1||\boldsymbol{\beta}||_1. \tag{20}$$

In this form we can apply the SAFE rule (2) to obtain a rule for discarding predictors. Now $|\mathbf{x}_j^{*T}\mathbf{y}^*| = |\mathbf{x}_j^T\mathbf{y}|$, $||\mathbf{x}_j^*|| = \sqrt{||\mathbf{x}_j||^2 + \lambda_2}$, $||\mathbf{y}^*|| = ||\mathbf{y}||$. Hence the global rule for discarding predictor j is

$$|\mathbf{x}_j^T\mathbf{y}| < \lambda_1 - ||\mathbf{y}|| \cdot \sqrt{||\mathbf{x}_j||^2 + \lambda_2} \cdot \frac{\lambda_{1max} - \lambda_1}{\lambda_{1max}} \tag{21}$$

Note that the `glmnet` package uses the parametrization $((1-\alpha)\lambda, \alpha\lambda)$ rather than (λ_2, λ_1). With this parametrization the basic SAFE rule has the form

$$|\mathbf{x}_j^T\mathbf{y}| < \Big(\alpha\lambda - ||\mathbf{y}|| \cdot \sqrt{||\mathbf{x}_j||^2 + (1-\alpha)\lambda} \cdot \frac{\lambda_{max} - \lambda}{\lambda_{max}}\Big) \tag{22}$$

The strong screening rules turn out to be the same as for the lasso. With the `glmnet` parametrization the global rule is simply

$$|\mathbf{x}_j^T\mathbf{y}| < \alpha(2\lambda - \lambda_{max}) \tag{23}$$

while the sequential rule is

$$|\mathbf{x}_j^T\mathbf{r}| < \alpha(2\lambda - \lambda_0). \tag{24}$$

Figure 5 show results for the elastic net with standard independent Gaussian data, $n = 100, p = 1000$, for 3 values of α. There were no violations in any of these figures, i.e. no predictor was discarded that had a non-zero coefficient at the actual solution. Again we see that the strong sequential rule performs extremely well, leaving only a small number of excess predictors at each stage.

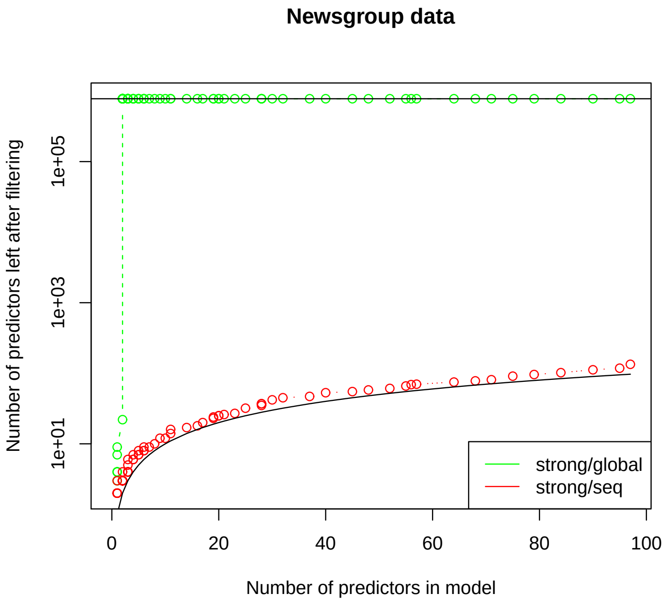

Figure 6: *Logistic regression: results for newsgroup example, using the new global rule (28) and the new sequential rule (29). The black curve is the 45^o line, drawn on the log scale.*

4 Logistic regression

Here we have a binary response $y_i = 0, 1$ and we assume the logistic model

$$\Pr(Y = 1|x) = 1/(1 + \exp(-\beta_0 - x^T\beta)) \tag{25}$$

Letting $p_i = \Pr(Y = 1|x_i)$, the penalized log-likelihood is

$$\ell(\beta_0, \boldsymbol{\beta}) = -\sum_i [y_i \log p_i + (1 - y_i)\log(1 - p_i)] + \lambda||\beta||_1 \tag{26}$$

El Ghaoui et al. (2010) derive an exact global rule for discarding predictors, based on the inner products between $\mathbf{y}$ and each predictor, using the same kind of dual argument as in the Gaussian case.

Here we investigate the analogue of the strong rules (11) and (12). The subgradient equation for logistic regression is

$$\mathbf{X}^T(\mathbf{y} - \mathbf{p}(\boldsymbol{\beta})) = \lambda \cdot \text{sign}(\boldsymbol{\beta}) \tag{27}$$

This leads to the global rule: letting $\bar{\mathbf{p}} = \mathbf{1}\bar{y}$, $\lambda_{max} = \max|\mathbf{x}_j^T(\mathbf{y} - \bar{\mathbf{p}})|$, we discard predictor j if

$$|\mathbf{x}_j^T(\mathbf{y} - \bar{\mathbf{p}})| < 2\lambda - \lambda_{max} \tag{28}$$

The sequential version, starting at λ_0, uses $\mathbf{p}_0 = \mathbf{p}(\hat{\beta}_0(\lambda_0), \hat{\boldsymbol{\beta}}(\lambda_0))$:

$$|\mathbf{x}_j^T(\mathbf{y} - \mathbf{p}_0)| < 2\lambda - \lambda_0. \tag{29}$$

Figure 6 show the result of various rules in an example, the newsgroup document classification problem (Lang 1995). We used the training set cultured from these data by (Koh et al. 2007). The response is binary, and indicates a subclass of topics; the predictors are binary, and indicate the presence of particular tri-gram sequences. The predictor matrix has 0.05% nonzero values. [2] Results for are shown for the new global rule (28) and the new sequential rule (29). We were unable to compute the logistic regression global SAFE rule for this example, using our R language implementation, as this had a very long computation time. But in smaller examples it performed much like the global SAFE rule in the Gaussian case. Again we see that the strong sequential rule (29), after computing the inner product of the residuals with all predictors at each stage, allows us to discard the vast majority of the predictors before fitting. There were no violations of either rule in this example.

Some approaches to penalized logistic regression such as the `glmnet` package use a weighted least squares iteration within a Newton step. For these algorithms, an alternative approach to discarding predictors would be to apply one of the Gaussian rules within the weighted least squares iteration.

Wu et al. (2009) used $|\mathbf{x}_j^T(\mathbf{y}-\bar{\mathbf{p}})|$ to screen predictors (SNPs) in genome-wide association studies, where the number of variables can exceed a million. Since they only anticipated models with say $k < 15$ terms, they selected a small multiple, say $10k$, of SNPs and computed the lasso solution path to k terms. All the screened SNPs could then be checked for violations to verify that the solution found was global.

5 Strong rules for general problems

Suppose that we have a convex problem of the form

$$\text{minimize}_{\boldsymbol{\beta}}\Big[f(\boldsymbol{\beta})+\lambda\cdot\sum_{j=1}^{p}g(\boldsymbol{\beta}_j)\Big] \tag{30}$$

where f and g are convex functions, f is differentiable and $\boldsymbol{\beta}=(\boldsymbol{\beta}_1,\boldsymbol{\beta}_2,\ldots\boldsymbol{\beta}_p)$ with each $\boldsymbol{\beta}_j$ being a scalar or vector. Suppose further that the subgradient equation for this problem has the form

$$f'(\boldsymbol{\beta})+\lambda\mathbf{s}_j=0;\ j=1,2,\ldots p \tag{31}$$

where each subgradient variable $\mathbf{s}_j$ satisfies $||\mathbf{s}_j||_q \le A$, and $||\mathbf{s}_j||_q = A$ when the constraint $g(\boldsymbol{\beta}_j)=0$ is satisfied (here $||\cdot||_q$ is a norm). Suppose we have two values $\lambda < \lambda_0$, and corresponding solutions $\hat{\boldsymbol{\beta}}(\lambda), \hat{\boldsymbol{\beta}}(\lambda_0)$. Then following the same logic as in Section 3, we can derive the general strong bound

$$\max||f'(\hat{\boldsymbol{\beta}}_{0j})||_q < (1+A)\lambda - A\lambda_0 \tag{32}$$

This can be applied either globally or sequentially. In the lasso regression setting, it is easy to check that this reduces to the rules (11),(12) where $A=1$.

The rule (32) has many potential applications. For example in the graphical lasso for sparse invariance covariance estimation (Friedman et al. 2007), we observe N multivariate normal observations of dimension p, with mean 0 and covariance Σ, with observed empirical covariance matrix S. Letting $\Theta=\Sigma^{-1}$, the problem is to maximize the penalized log-likelihood

$$\log\det\Theta - \operatorname{tr}(S\Theta) - \lambda||\Theta||_1, \tag{33}$$

over non-negative definite matrices Θ. The penalty $||\Theta||_1$ sum the absolute values of the entries of Θ; we assume that the diagonal is not penalized. The subgradient equation is

$$\Sigma - S - \lambda\cdot\Gamma = 0, \tag{34}$$

where $\Gamma_{ij}\in\operatorname{sign}(\Theta_{ij})$. The graphical lasso algorithm proceeds in a blockwise fashion, optimizing one whole row and column at a time. For some row i, denote by $S_{i,-i}$ by s_{12} and similarly σ_{12} and Γ_{12} for Σ and Γ, respectively. Then the subgradient equation for one row has the form

$$\sigma_{12} - s_{12} - \lambda\cdot\Gamma_{12} = 0, \tag{35}$$

[2] This dataset is available as a saved R data object at `http://www-stat.stanford.edu/ hastie/glmnet`

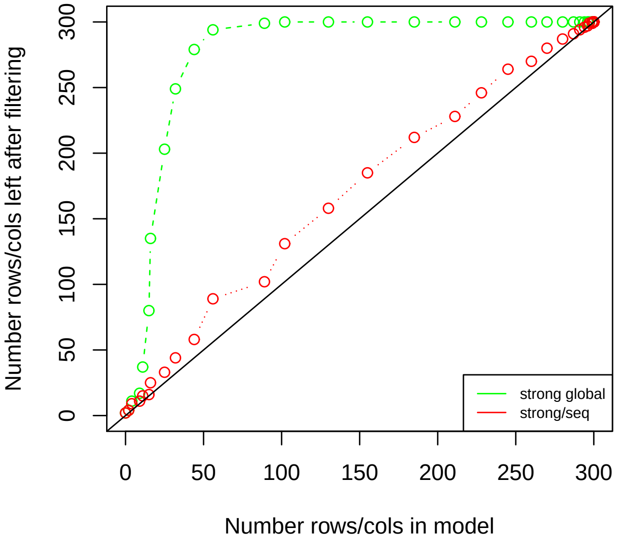

Figure 7: *Strong global and sequential rules applied to the graphical lasso.*

Now given two values $\lambda < \lambda_0$, and solution $\hat{\Sigma}^0$ at λ_0, we have the strong sequential rule

$$\max|\hat{\sigma}^0_{12} - s_{12}| < 2\lambda - \lambda_0 \tag{36}$$

If this rule is satisfied, we discard the entire ith row and column of Θ, and hence set them to zero (but retain the ith diagonal element). Figure 7 shows an example with $N = 100, p = 300$, standard independent Gaussian variates. No violations of the rule occurred.

Finally, we note that strong rules can be derived in a similar way, for other problems such as the group lasso (Yuan & Lin 2007).

6 Implementation and numerical studies

The strong sequential rule (12) can be used to provide potential speed improvements in convex optimization problems. Generically, given a solution $\hat{\boldsymbol{\beta}}(\lambda_0)$ and considering a new value $\lambda < \lambda_0$, let $S(\lambda)$ be the indices of the predictors that survive the screening rule (12): we call this the *strong set*. Denote by E the eligible set of predictors. Then a useful strategy would be

1. Set $E = S(\lambda)$.

2. Solve the problem at value λ using only the predictors in E.

3. Check the KKT conditions at this solution for all predictors. If there are no violations, we are done. Otherwise add the predictors that violate the KKT conditions to the set E, and repeat steps 2 and 3.

Depending on how the optimization is done in step 2, this can be quite effective. Now in the `glmnet` procedure, coordinate descent is used, with warm starts over a grid of decreasing values of λ. In addition, an "ever-active" set of predictors $A(\lambda)$ is maintained, consisting of the indices of all predictors that have a

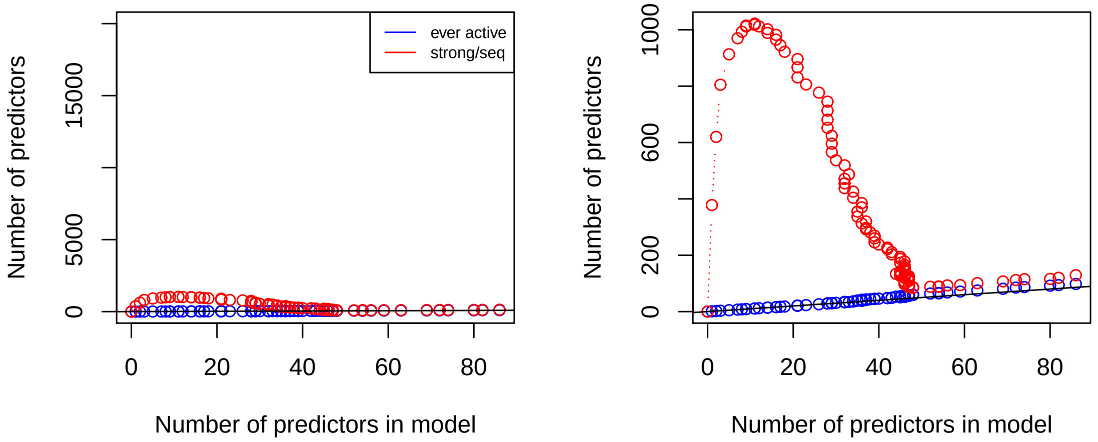

Figure 8: *Gaussian lasso setting, $N = 200, p = 20,000$, pairwise correlation between features of* 0.7*. The first 50 predictors have positive, decreasing coefficients. Shown are the number of predictors left after applying the strong sequential rule (12) and the number that have ever been active (i.e. had a non-zero coefficient in the solution) for values of λ larger than the current value. The right-hand plot is a zoomed version of the left plot.*

non-zero coefficient for some λ' greater than the current value λ under consideration. The solution is first found for this active set: then the KKT conditions are checked for all predictors. if there are no violations, then we have the solution at λ; otherwise we add the violators into the active set and repeat.

The two strategies above are very similar, with one using the strong set $S(\lambda)$ and the other using the ever-active set $A(\lambda)$. Figure 8 shows the active and strong sets for an example. Although the strong rule greatly reduces the total number of predictors, it contains more predictors than the ever-active set; accordingly, violations occur more often in the ever-active set than the strong set. This effect is due to the high correlation between features and the fact that the signal variables have coefficients of the same sign. It also occurs with logistic regression with lower correlations, say 0.2.

In light of this, we find that using both $A(\lambda)$ and $S(\lambda)$ can be advantageous. For `glmnet` we adopt the following combined strategy:

1. Set $E = A(\lambda)$.
2. Solve the problem at value λ using only the predictors in E.
3. Check the KKT conditions at this solution for all predictors in $S(\lambda)$. If there are violations, add these predictors into E, and go back to step 1 using the current solution as a warm start.
4. Check the KKT conditions for all predictors. If there are no violations, we are done. Otherwise add these violators into $A(\lambda)$, recompute $S(\lambda)$ and go back to step (1) using the current solution as a warm start.

Note that violations in step 3 are fairly common, while those in step 4 are rare. Hence the fact that the size of $S(\lambda)$ is $\ll p$ can make this an effective strategy.

We implemented this strategy and compare it to the standard `glmnet` algorithm in a variety of problems, shown in Tables 1–4. Details are given in the table captions. We see that the new strategy offers a speedup factor of five or more in some cases, and never seems to slow things down.

The strong sequential rules also have the potential for space savings. With a large dataset, one could compute the inner products $\{\mathbf{x}_j^T\mathbf{r}\}_1^p$ offline to determine the strong set of predictors, and then carry out the intensive optimization steps in memory using just this subset of the predictors.

7 Discussion

In this paper we have proposed strong global and sequential rules for discarding predictors in statistical convex optimization problems such as the lasso. When combined with checks of the KKT conditions, these can offer substantial improvements in speed while still yielding the exact solution. We plan to include these rules in a future version of the `glmnet` package.

Acknowledgements: We would like to thank Jacob Bien, Noah Simon, Jonathan Taylor and Stephen Boyd for helpful discussions. Finally we thank Laurent El Ghaoui and his co-authors for sharing their paper before publication, and for helpful discussions.

References

Efron, B., Hastie, T., Johnstone, I. & Tibshirani, R. (2004), ‘Least angle regression’, *Annals of Statistics* **32**(2), 407–499. (with discussion).

El Ghaoui, L., Viallon, V. & Rabbani, T. (2010), Safe feature elimination in sparse supervised learning, Technical Report UC/EECS-2010-126, EECS Dept., University of California at Berkeley.

Fan, J. & Lv, J. (2008), ‘Sure independence screening for ultra-high dimensional feature space’, *Journal of the Royal Statistical Society Series B, to appear* .

Friedman, J., Hastie, T., Hoefling, H. & Tibshirani, R. (2007), ‘Pathwise coordinate optimization’, *Annals of Applied Statistics* **2**(1), 302–332.

Friedman, J., Hastie, T. & Tibshirani, R. (2008), ‘Regularization paths for generalized linear models via coordinate descent’, *Journal of Statistical Software* **33**.

Koh, K., Kim, S.-J. & Boyd, S. (2007), ‘An interior-point method for large-scale l1-regularized logistic regression’, *Journal of Machine Learning Research* **8**, 1519–1555.

Lang, K. (1995), Newsweeder: Learning to filter netnews., *in* ‘Proceedings of the Twenty-First International Conference on Machine Learning (ICML)’, pp. 331–339.

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

Tibshirani, R. & Taylor, J. (2010), Regularization paths for least squares problems with generalized ℓ_1 penalties. submitted.
*http://www-stat.stanford.edu/∼ryantibs/papers/dualpath.pdf

Wu, T. T., Chen, Y. F., Hastie, T., Sobel, E. & Lange, K. (2009), ‘Genomewide association analysis by lasso penalized logistic regression’, *Bioinformatics* **25**(6), 714–721.

Yuan, M. & Lin, Y. (2007), ‘Model selection and estimation in regression with grouped variables’, *Journal of the Royal Statistical Society, Series B* **68**(1), 49–67.

Zou, H. & Hastie, T. (2005), ‘Regularization and variable selection via the elastic net’, *Journal of the Royal Statistical Society Series B.* **67**(2), 301–320.

Appendix

A sketch of the proof of Theorem 1: Tibshirani & Taylor (2010) consider a generalized lasso problem of the form

$$\frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2 + \lambda||\mathbf{D}\boldsymbol{\beta}||_1 \tag{37}$$

where $\mathbf{D}$ is a general $m \times p$ penalty matrix. Letting $\mathbf{X}^+ = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ (a pseudo-inverse of $\mathbf{X}$), $\tilde{\mathbf{y}} = \mathbf{X}\mathbf{X}^+\mathbf{y}$, $\tilde{\mathbf{D}} = \mathbf{X}^+\mathbf{D}$, then derive the dual problem with dual variables $\mathbf{u}_\lambda$ satisfying $\max|\mathbf{u}_\lambda| \leq \lambda$. At the solution, the primal and dual variables are related by $\hat{\boldsymbol{\beta}}_\lambda = \mathbf{X}^+(\tilde{y} - \tilde{\mathbf{D}}'\hat{\mathbf{u}}_\lambda)$. Tibshirani & Taylor (2010) discuss a *boundary lemma* which says that once an element of $\hat{\mathbf{u}}_\lambda$ is equal to λ ("on the boundary") for some λ, it will remain there for all $\lambda' < \lambda$. They show that a sufficient condition for this is that $\tilde{\mathbf{D}}\tilde{\mathbf{D}}^T$ be diagonally dominant, and in the process show that this implies that the slopes of all of the $\hat{\mathbf{u}}_\lambda$ variables are less than one in absolute value. Now in the case of the lasso, $\mathbf{D}$ is the identity matrix, an element of $\hat{\mathbf{u}}_\lambda$ being on the boundary means that the variable has a non-zero coefficient, and the slopes of the $\hat{\mathbf{u}}_\lambda$ variable are the slopes $dc_j(\lambda)/d\lambda$. Finally $\tilde{\mathbf{D}}\tilde{\mathbf{D}}^T = (\mathbf{X}^T\mathbf{X})^{-1}$ and we have (15).

Method	Population correlation			
	0.0	0.25	0.5	0.75
glmnet	4.07	6.13	9.50	17.70
with seq-strong	2.50	2.54	2.62	2.98

Table 1: *Glmnet timings (seconds) for fitting a lasso problem in the Gaussian setting. There are* $p = 100,000$ *predictors,* $N = 200$ *observations, 30 nonzero coefficients, with the same value and signs alternating; signal-to-noise ratio equal to 3.*

Method	Time (sec.)
glmnet	4.14
with seq-strong	2.52

Table 2: *Glmnet timings (seconds) fitting a lasso problem in the Gaussian setting. Here the data matrix is sparse, consisting of just zeros and ones, with* 0.1% *of the values equal to 1. There are* $p = 50,000$ *predictors,* $N = 500$ *observations, with 25% of the coefficients nonzero, having a Gaussian distribution; signal-to-noise ratio equal to 4.3.*

Method	α				
	1.0	0.5	0.2	0.1	0.01
glmnet	9.49	7.98	5.88	5.34	5.26
with seq-strong	2.64	2.65	2.73	2.99	5.44

Table 3: *Glmnet timings (seconds) for fitting an elastic net problem. There are* $p = 100,000$ *predictors,* $N = 200$ *observations, 30 nonzero coefficients, with the same value and signs alternating; signal-to-noise ratio equal to 3*

.

Method	Population correlation		
	0.0	0.5	0.8
glmnet	11.71	12.41	12.69
with seq-strong	6.31	9.491	12.86

Table 4: *Glmnet timings (seconds) fitting a lasso/logistic regression problem. Here the data matrix is sparse, consisting of just zeros and ones, with* 0.1% *of the values equal to 1. There are* $p = 50,000$ *predictors,* $N = 800$ *observations, with 30% of the coefficients nonzero, with the same value and signs alternating; Bayes error equal to 3%.*