

Strong Rules for Discarding Predictors in Lasso-type Problems

Robert Tibshirani

Joint work with Jacob Bien, Jerry Friedman, Trevor Hastie, Noah Simon, Jon Taylor, Ryan Tibshirani

7 authors!- generated lots of discussion.

Thanks to Stephen Boyd, Emmanuel Candes, Laurent El Ghaoui, Rahul Mazumder for helpful discussion.

Email: `tibs@stanford.edu`

`http://www-stat.stanford.edu/~tibs`

“God knows the last thing we need is another algorithm for the lasso”

Stephen Boyd, Sept 28, 2010

This is not quite a talk about algorithms for the lasso– but ideas for speeding up existing algorithms.

Also reveals interesting aspects of convex statistical problems.

Top 7 reasons why this Lasso/L1 stuff may have gone too far

1. One of Tibshirani's students just wrote a paper on the "Generalized adaptive doubly sparse grouped relaxed lasso"
2. One of Candes's students just wrote a paper on the "**Near-optimality** of the generalized adaptive doubly sparse grouped relaxed lasso"
3. One of Donoho's students just wrote a paper on "**Higher criticism** for near-optimality of the generalized adaptive doubly sparse grouped relaxed lasso"
4. Papers are now being rejected out-of-hand from *Statistica Sinica* if they don't mention L1 penalties at least three times in the abstract
5. There are now more Lasso algorithms than millionaires at FaceBook
6. Someone discovered a computational fact about the lasso that's NOT an exercise in Boyd's *Convex Optimization* book
7. The Bayesians are getting really pissed off!

The Lasso

Usual regression setting $N \times p$ matrix of predictors \mathbf{X} , N -vector of outcomes \mathbf{y} .

$$\hat{\boldsymbol{\beta}} = \operatorname{argmin}_{\boldsymbol{\beta}} \frac{1}{2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1, \quad (1)$$

“ ℓ_1 - regression”, “Basis pursuit” (Chen, Donoho and Saunders, 1997). Delivers a sparse solution vector $\hat{\boldsymbol{\beta}}$.

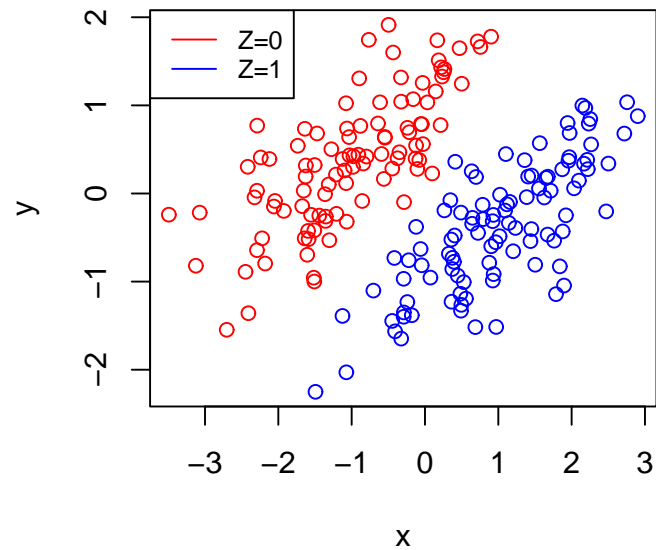
Let \mathbf{x}_j be the j th column of \mathbf{X} .

Question: before we fit the lasso for a fixed value λ , can we safely discard a predictor j (set $\hat{\beta}_j = 0$) based just on the inner product $\mathbf{x}_j^T \mathbf{y}$?

...Seems unlikely

Think of linear regression analogy- where marginal correlation can be zero while partial correlation is not.

Lasso becomes linear regression as $\lambda \rightarrow 0$.



Surprise! (to me, at least)

- El Ghaoui et al. (2010) recently derived “SAFE” rules that **guarantee** that a predictor will have a coefficient of zero in the lasso fit with parameter λ , based just on the inner product $\mathbf{x}_j^T \mathbf{y}$.
- They are useful for saving time and space in lasso-type computations, but they are limited in what they can achieve
- In this talk we propose **strong rules** that are **sequential** and discard many more predictors than the SAFE rules.
- BUT the strong rules **are not foolproof**— sometimes they discard predictors that have a non-zero coefficient. Fortunately they can be combined with simple checks of the KKT conditions to make them safe.

Outline

- The SAFE rules and their derivation from the dual
- Strong rules for lasso
- Implementation and Timings
- Strong rules for general problems

Related work

- Sure independence screening “SIS” (Fan and Lv 2008) screens predictors based on inner products, and then fits on remaining predictors; asymptotic justification; goal is not to obtain the exact lasso solution
- This talk is about **exact computation** for the lasso, not asymptotic approximation. But it may have consequences for SIS as well.
- Wu, Chen, Hastie, Sobel and Lange (2009) proposed “swindles” for screening predictors in ℓ_1 -penalized logistic regression, based on the univariate inner products. Similar idea to this talk, but not as principled and not sequential.

SAFE rules for the Lasso

$$\hat{\boldsymbol{\beta}} = \operatorname{argmin}_{\boldsymbol{\beta}} \frac{1}{2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1, \quad (2)$$

SAFE rule: discard predictor j if

$$|\mathbf{x}_j^T \mathbf{y}| < \lambda - \|\mathbf{x}_j\|_2 \|\mathbf{y}\|_2 \frac{\lambda_{\max} - \lambda}{\lambda_{\max}}, \quad (3)$$

where $\lambda_{\max} = \max_j |\mathbf{x}_j^T \mathbf{y}|$ is the smallest λ for which all coefficients are zero.

More on the SAFE rule

-

$$|\mathbf{x}_j^T \mathbf{y}| < \lambda - \|\mathbf{x}_j\|_2 \|\mathbf{y}\|_2 \frac{\lambda_{\max} - \lambda}{\lambda_{\max}}, \quad (4)$$

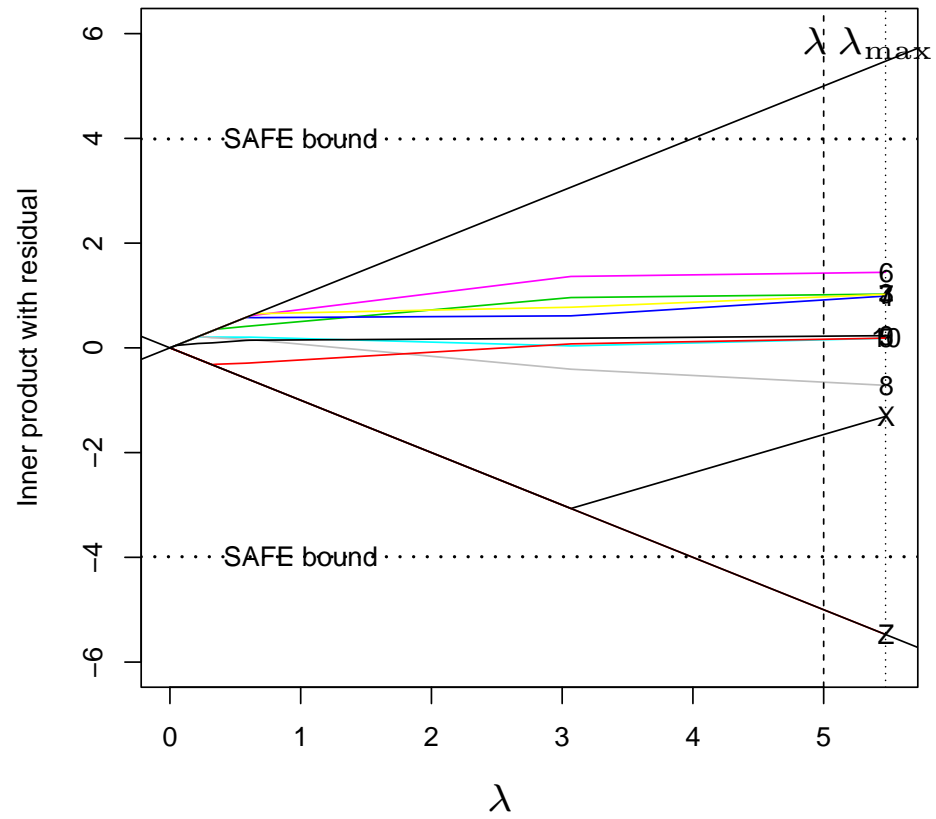
- Necessary and sufficient (KKT) conditions for solution:

$$\mathbf{x}_j^T (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) = \lambda \cdot s(\beta_j) \quad \text{subgradient equations}$$

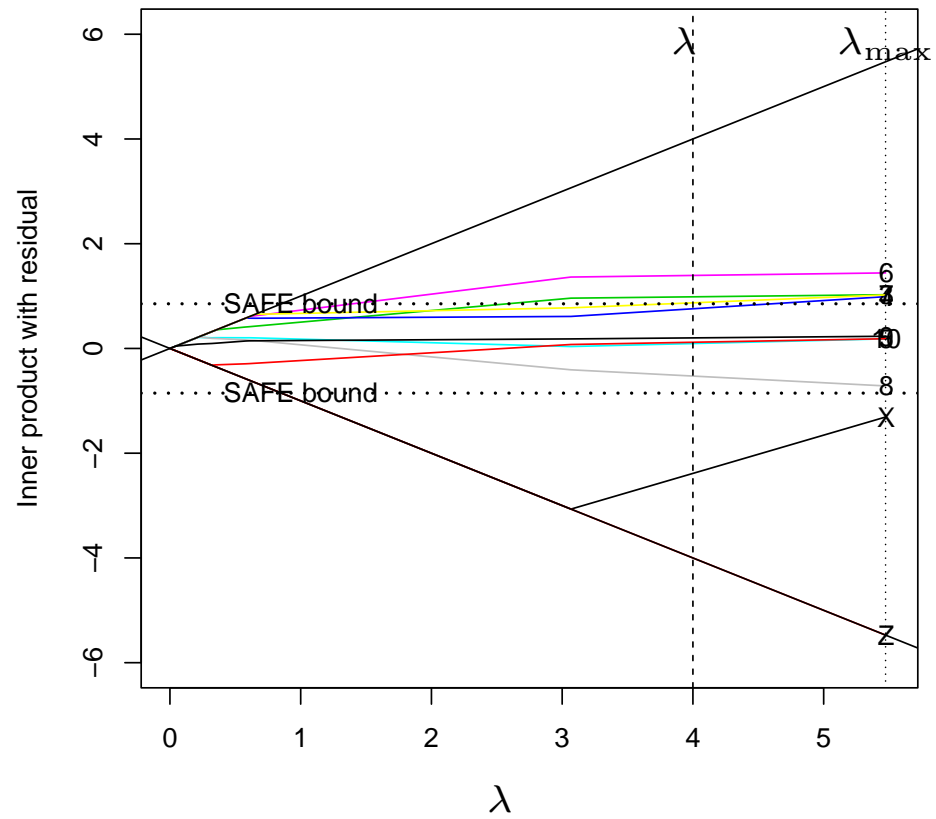
$$s(\beta_j) = \text{sign}(\beta_j) \text{ if } \beta_j \neq 0, \quad s(\beta_j) \in [-1, 1] \text{ if } \beta_j = 0$$

- At $\boldsymbol{\beta} = 0$, KKT conditions are simply $\mathbf{x}_j^T \mathbf{y} = \pm \lambda_{\max}$
- SAFE rule adjusts the bound downward, to account for the fact that we have moved from λ_{\max} to λ

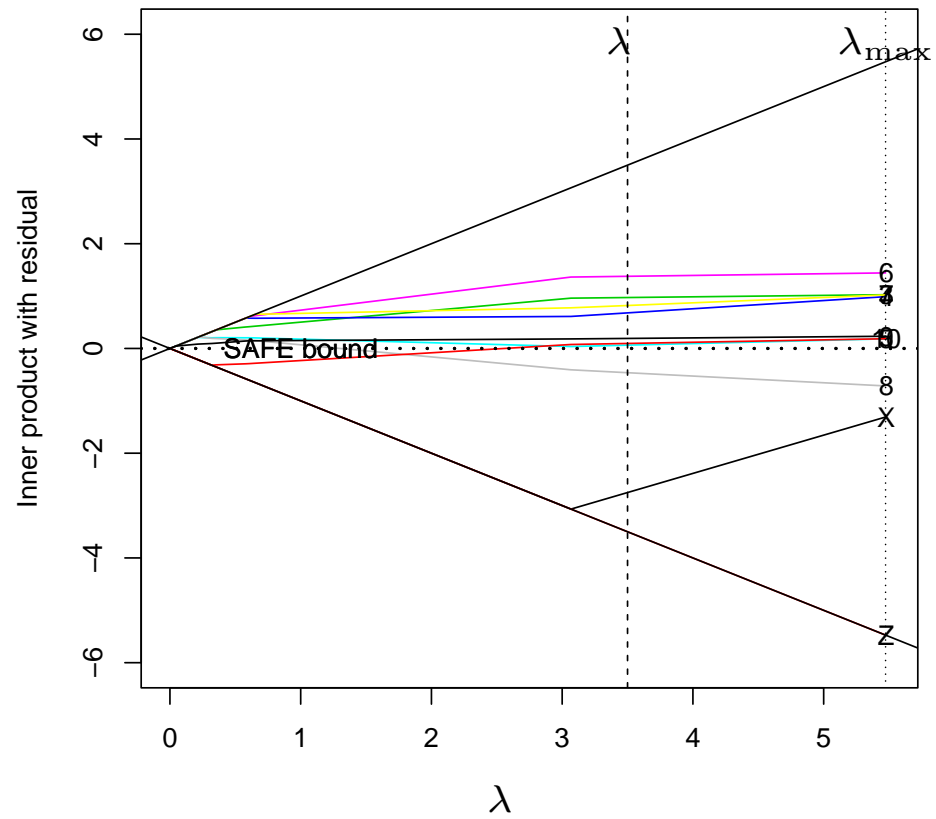
SAFE bound for the X, Z example



...with a smaller lambda



...with an even smaller lambda



Derivation/proof for the SAFE bound

Start with lasso: $\operatorname{argmin}_{\boldsymbol{\beta}} \frac{1}{2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda \|\boldsymbol{\beta}\|_1$

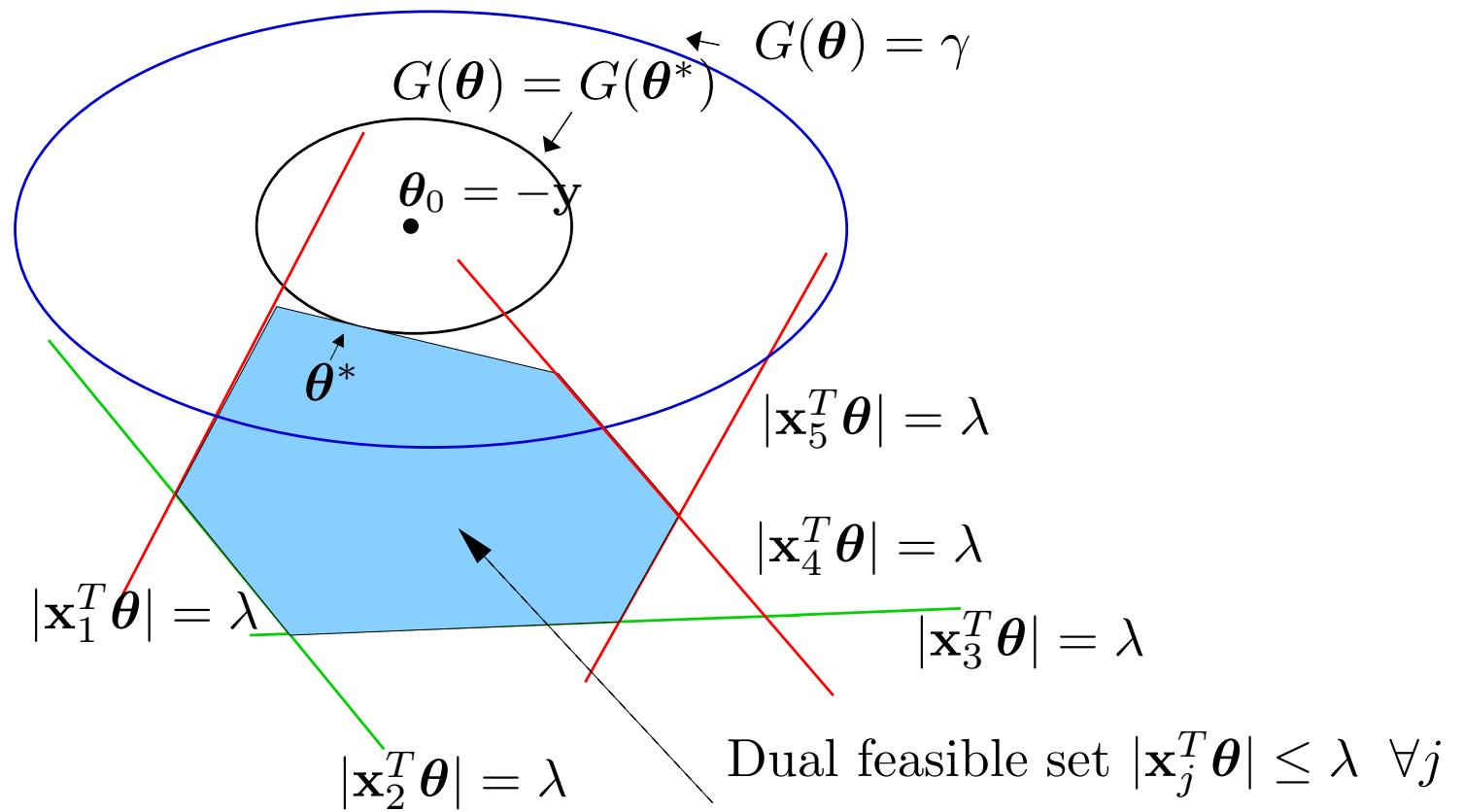
Focus on the equivalent Lagrange dual problem:

$$\begin{aligned} \hat{\boldsymbol{\theta}} &= \operatorname{argmax}_{\boldsymbol{\theta}} G(\boldsymbol{\theta}) = \frac{1}{2} \|\mathbf{y}\|_2^2 - \frac{1}{2} \|\mathbf{y} + \boldsymbol{\theta}\|_2^2 \\ &\text{subject to } |\mathbf{x}_j^T \boldsymbol{\theta}| \leq \lambda \text{ for } j = 1, \dots, p. \end{aligned} \quad (5)$$

The relationship between the primal and dual solutions is $\hat{\boldsymbol{\theta}} = \mathbf{X}\hat{\boldsymbol{\beta}} - \mathbf{y}$.

$|\mathbf{x}_j^T \hat{\boldsymbol{\theta}}| < \lambda$ means $\hat{\beta}_j = 0$.

The dual of the lasso



Discard predictor j if $m_j = \max_{\boldsymbol{\theta}} |\mathbf{x}_j^T \boldsymbol{\theta}|$ over the set $G(\boldsymbol{\theta}) \geq \gamma$ is less than λ

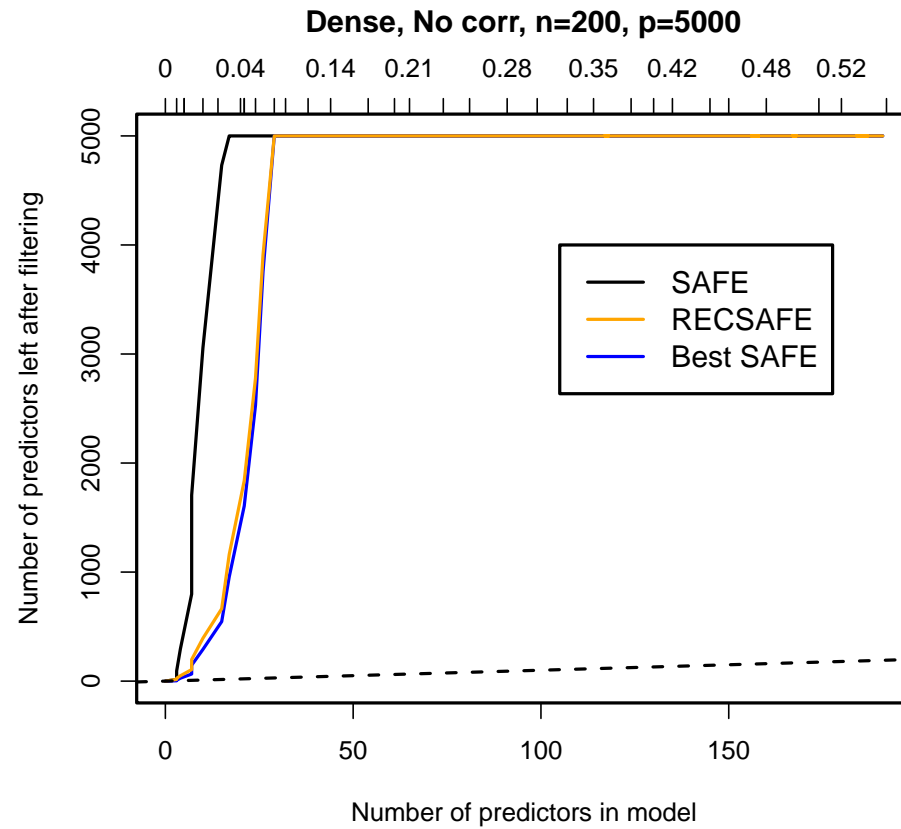
Derivation of the SAFE rule

- Find a dual feasible point θ' with dual value $\gamma = G(\theta')$.
- Find

$$m_j = \max_{\theta} |\mathbf{x}_j^T \theta| \text{ over the set } G(\theta) \geq \gamma. \quad (6)$$

- If $m_j < \lambda$, discard predictor j
- How to find θ' ? They suggest: start with $\theta_0 = \mathbf{y}$ and scale it so that it is feasible. Then the optimization in (6) can be done algebraically and this leads to the basic SAFE rule.
- They also derive a **recursive SAFE** rule which starts with θ_0 equal to the dual solution for some $\lambda' > \lambda$.
- For reference, we also compute the **Best SAFE rule** obtained by setting θ_0 equal to the actual dual solution at λ .

Numerical example



(Doesn't depend on much on the correlation between predictors)

Summary of SAFE rules

- provably correct, somewhat useful
- can be extended to other settings (messy)
- to improve them significantly (discard more predictors), have to allow occasional violations

Strong rules

- **Basic (global) rule**

$$\begin{aligned} |\mathbf{x}_j^T \mathbf{y}| &< 2\lambda - \lambda_{\max} \\ &= \lambda - (\lambda_{\max} - \lambda), \end{aligned} \tag{7}$$

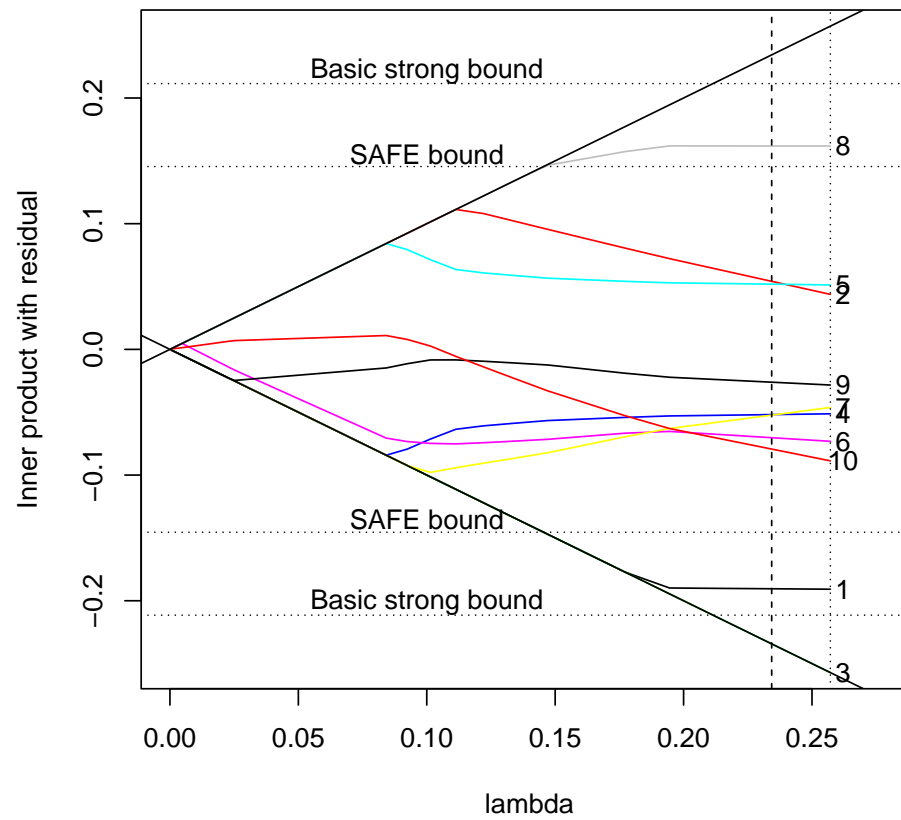
where as before $\lambda_{\max} = \max_j |\mathbf{x}_j^T \mathbf{y}|$.

- When the predictors are standardized ($\|\mathbf{x}_j\|_2 = 1$ for each j), we have $\lambda_{\max} \leq \|\mathbf{y}\|_2$, so that

$$\lambda - \|\mathbf{y}\|_2 \frac{\lambda_{\max} - \lambda}{\lambda_{\max}} \leq \lambda - (\lambda_{\max} - \lambda) = 2\lambda - \lambda_{\max}.$$

Hence strong bound \geq SAFE bound (discards more predictors).

Basic (global) SAFE and strong bounds



****The take-home slide****

Sequential strong rule

- Suppose that we have already computed the solution $\hat{\beta}(\lambda_0)$ at λ_0 , and wish to discard predictors for a fit at $\lambda < \lambda_0$. Defining the residual $\mathbf{r} = \mathbf{y} - \mathbf{X}\hat{\beta}(\lambda_0)$, our *strong sequential rule* discards predictor j if

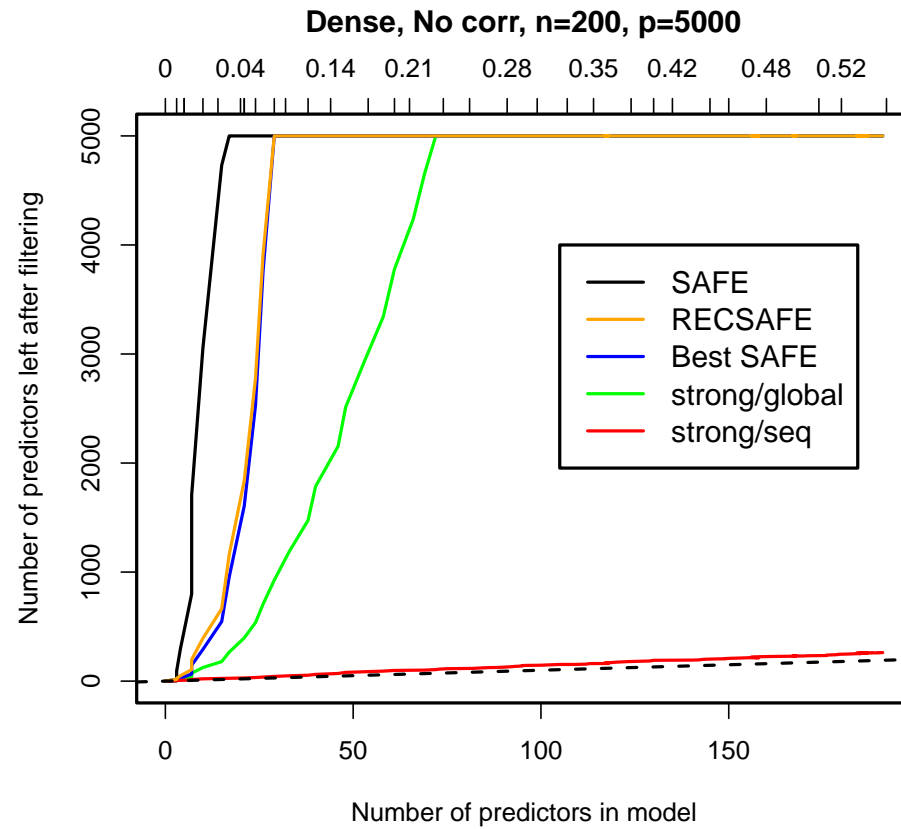
$$\begin{aligned} |\mathbf{x}_j^T \mathbf{r}| &< 2\lambda - \lambda_0 \\ &= \lambda - (\lambda_0 - \lambda) \end{aligned} \tag{8}$$

- Recall that at λ , the active variables satisfy $|\mathbf{x}_j^T (\mathbf{y} - \mathbf{X}\hat{\beta}(\lambda))| = \lambda$. We think of $(\lambda_0 - \lambda)$ as a “buffer”, to allow the inner product to rise as we move from λ_0 to λ .

Strong rules are not foolproof

- For quite a while, we thought they were!
- They can be violated (but rarely)
- Violations seem to occur when $p \approx N$, but never when $p \gg N$, where they are most needed!
- There are **no violations** in any of the numerical examples in our talk/paper ($p \gg N$)
- **Good news**- can be complemented with simple checks of the KKT conditions, to ensure exact solution is obtained (details later)

Numerical example again

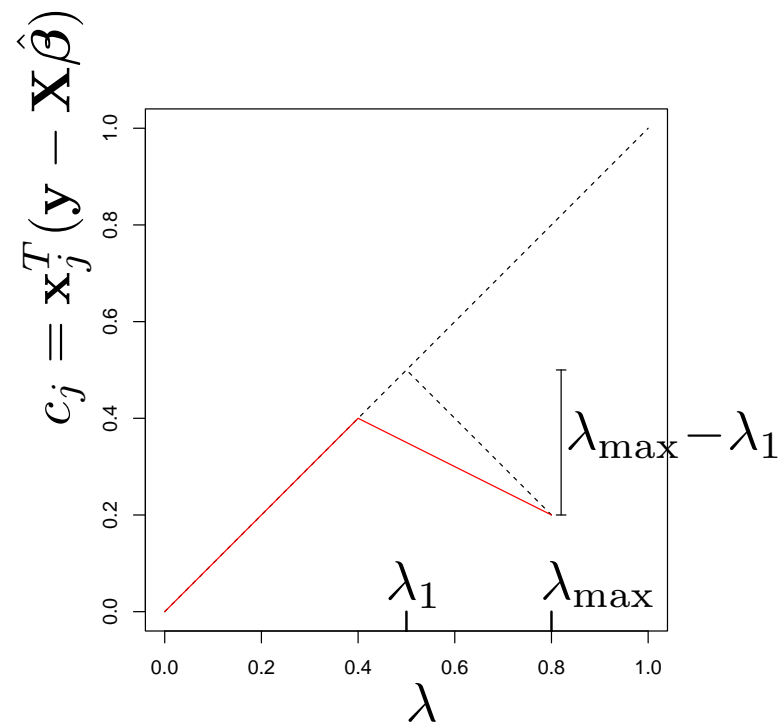


Decreasing $\lambda \rightarrow$

(Doesn't depend on much on the correlation between predictors)

Motivation for strong rules

If the slope of inner product as a function of λ is **less than or equal to 1 in absolute value**, then it can't change more than $\lambda_{\max} - \lambda_1$ as we move from λ_{\max} to λ_1 . (Proof omitted).



Where does unit slope condition come from?

or why it can pay to be sloppy at math!!

Recall the KKT conditions

-

$$c_j(\lambda) = \mathbf{x}_j^T (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) = \lambda \cdot s_j \quad (9)$$

where $s_j \in \text{sign}(\beta_j)$. Then

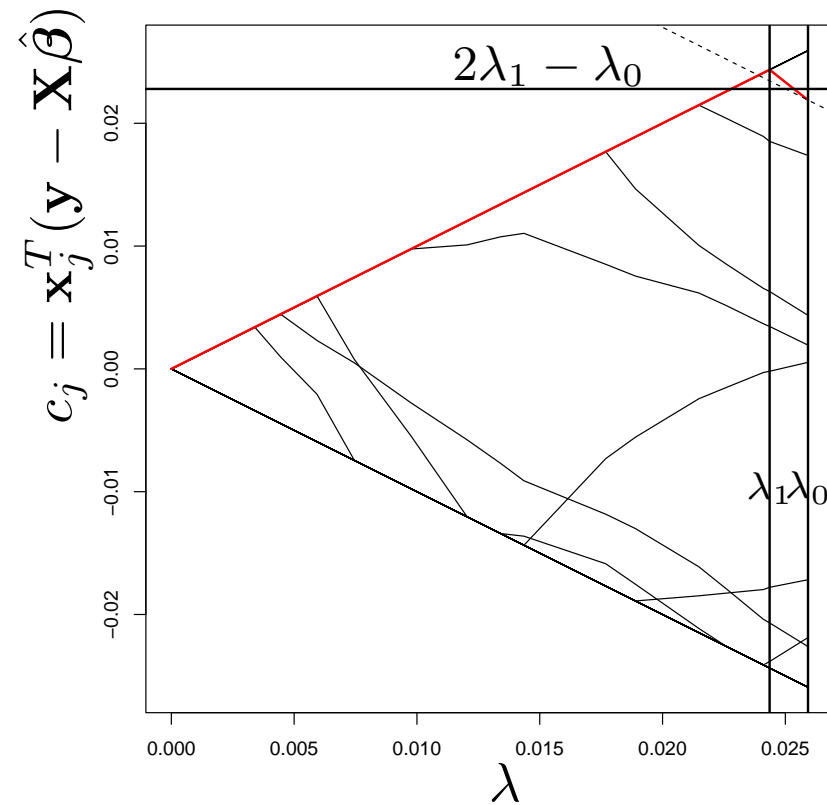
$$c'_j(\lambda) = s_j(\lambda) \in [-1, 1] ???$$

- Actually, $c_j(\lambda) = \mathbf{x}_j^T (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) = \lambda \cdot s_j(\lambda)$ and by the product rule

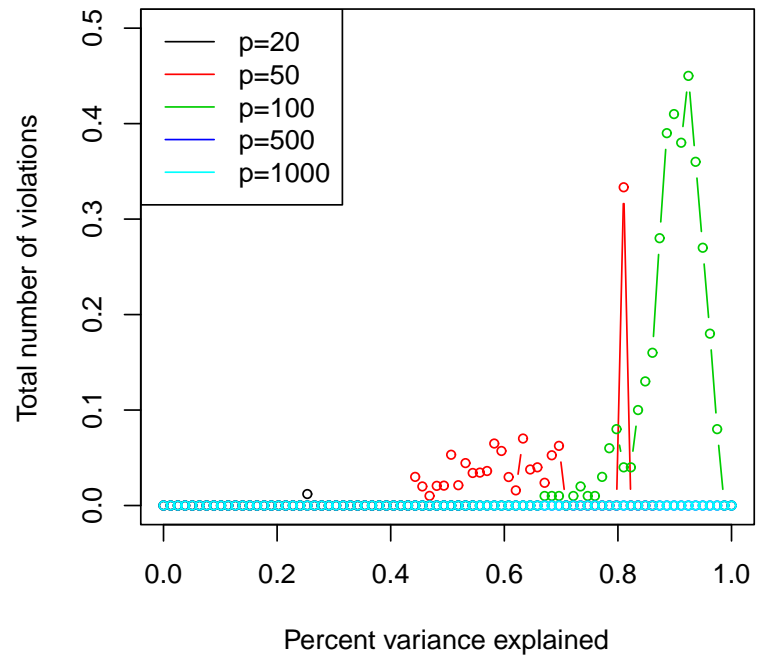
$$c'_j(\lambda) = s_j(\lambda) + \lambda \cdot s'_j(\lambda), \quad (10)$$

- for active variables, second term is zero and $c'_j(\lambda) = 1$.
- for inactive variables, slope bound is a very good heuristic

A violation of the sequential strong rule (Ryan)



Frequency of Violations



A sufficient condition for unit slope bound

- - $(\mathbf{X}^T \mathbf{X})^{-1}$ is diagonally dominant, (11)
- Follows from the “boundary lemma” in generalized lasso dual path (Ryan Tibshirani and Jon Taylor)
- Interestingly, it is this same condition under which lasso and Dantzig Selector (Candes & Tao) paths are identical (Meinshausen, Rocha, Yu)
- For general \mathbf{X} , we found that for the Dantzig Selector, the strong rules work well in sparse part of the path, but badly in the dense part.

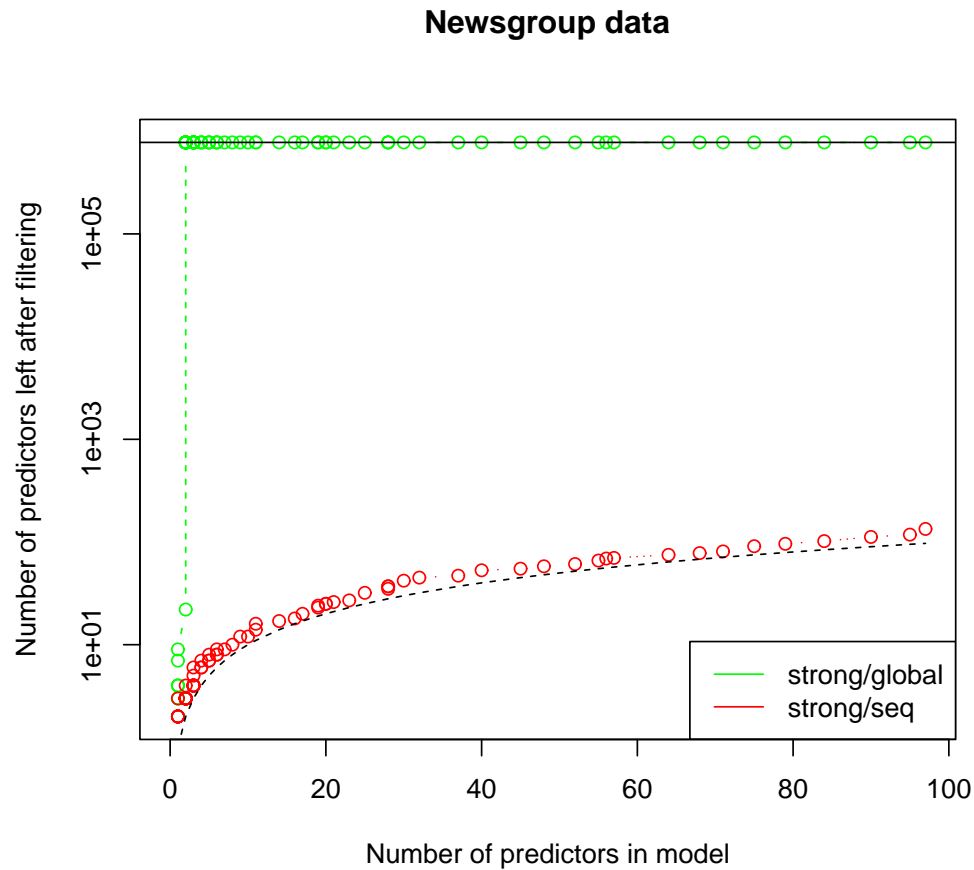
Logistic regression

$$\mathbf{p}_0 = \mathbf{p}(\hat{\beta}_0(\lambda_0), \hat{\beta}(\lambda_0)):$$

$$|\mathbf{x}_j^T (\mathbf{y} - \mathbf{p}_0)| < 2\lambda - \lambda_0. \quad (12)$$

Newsgroup data

$N = 11,314, p = 777,811, \mathbf{X}$ sparse



Implementing the sequential strong rule

- For use with any generic optimizer; have to guard against possible violations
- Given a solution $\hat{\beta}(\lambda_0)$ and considering a new value $\lambda < \lambda_0$, let $S(\lambda)$ be the indices of the predictors that survive the screening rule (8): we call this the **strong set**. Computational approach:
 1. Start with just the strong set $S(\lambda)$.
 2. Solve the problem at value λ using only the predictors in $S(\lambda)$
 3. Check the KKT conditions at the solution for all predictors. If there are no violations, we are done. Otherwise add the predictors that violate the KKT conditions to the strong set and repeat (2),(3).

Application to generalized gradient algorithm

- For lasso, basic iteration is

$$\hat{\beta} \leftarrow S_{t,\lambda}(\beta + t \cdot \mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta))$$

where S is the soft-threshold operator, t is a stepsize.

- Speedup: when $p > N$, reduces Np per iteration to $\approx N^2$
- Numerical example: generalized gradient with approximate backtracking applied to lasso, $N = 100$. 100 values of λ spanning the entire relevant range.

p	500	1000	5000	10000	p=500, N=1000
speedup factor	2.7	4.6	9.6	15.5	1.9

Similar results for Nesterov's momentum method

Implementation in glmnet

- Our **glmnet** program uses coordinate descent to solve the lasso, logistic/lasso and related problems. It is one of the fastest computational approaches available.
- It uses coordinate descent over a path of decreasing λ values with warm starts and an active set strategy- iteration is first done over variables that have ever had a non-zero coefficient for some earlier value of λ
- active set strategy is very effective
- Typically
$$\text{active set} \subset \text{strong set},$$
and so we use them both.

Glmnet timings

Lasso, $p = 100,000$ predictors, $N = 200$ observations, 30 nonzero coefficients; In the rightmost column, 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	Population correlation				
	0.0	0.25	0.5	0.75	Sparse
glmnet	4.07	6.13	9.50	17.70	4.14
with seq-strong	2.50	2.54	2.62	2.98	2.52

Average time in seconds

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%.

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

Strong rules for general problems

- Suppose that we have a convex problem of the form

$$\text{minimize}_{\boldsymbol{\beta}} \left[f(\boldsymbol{\beta}) + \lambda \cdot \sum_{k=1}^K g(\boldsymbol{\beta}_k) \right] \quad (13)$$

where f and g are convex functions, f is differentiable and $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \dots, \boldsymbol{\beta}_K)$ with each $\boldsymbol{\beta}_k$ being a scalar or vector.

- Then following the same logic as before, we can derive the general strong rule

$$\left\| \frac{f(\hat{\boldsymbol{\beta}}_{0k})}{d\boldsymbol{\beta}_k} \right\|_q < (1 + A)\lambda - A\lambda_0 \quad (14)$$

where A is a bound on the subgradient variable for g .

Graphical lasso for sparse covariance estimation

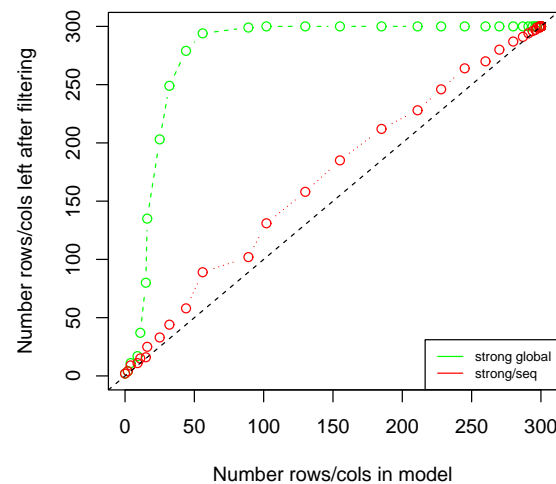
Observed covariance matrix \mathbf{S} . Maximize the penalized log-likelihood:

$$\log \det \Sigma + \text{tr}(\mathbf{S} \Sigma^{-1}) - \lambda \cdot |\Sigma^{-1}|_1$$

Graphical lasso uses blockwise coordinate descent, one row/col at a time.

Strong rule discards an entire row/column at once:

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



Final comments

- Can apply these ideas to other problems, eg. matrix completion (Rahul Mazunder, Trevor Hastie), group lasso, adaptive lasso etc.
- Application to other model selection problems?
- Why does strong rule “never” fail when $p \gg N$?

Predicted questions from audience

1. What exactly is plotted on the horizontal axis?
2. Does this have something to do with a result of Vapnik and Chernovenkis? (first part of a multi-part question)
3. Isn't this just a restatement of the ergodic theorem?
4. I'm sure this must all be an exercise in my Convex Optimization book!

References

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.