Fast Sparse Regression and Classification

Jerome H. Friedman

Stanford University

PREDICTION (Regression/Classification)

$y =$ outcome/response variable

$\mathbf{x} = \{x_1, \cdots, x_n\}$ predictors

Goal: $\hat{y} = F(\mathbf{x})$

Linear model:

$$F(\mathbf{x}; \mathbf{a}) = a_0 + \sum_{j=1}^{n} a_j x_j$$

Cost for error: $L(y, F)$

$(y - F)^2$: least–squares regression

$\log(1 + e^{-yF})$: logistic regression

Risk: $R(\mathbf{a}) = E_{\mathbf{x},y} L(y, F(\mathbf{x}; \mathbf{a}))$

Optimal solution: $\mathbf{a}^* = \arg\min_{\mathbf{a}} R(\mathbf{a})$

$p(\mathbf{x}, y)$ unknown $\Rightarrow \mathbf{a}^*$ unknown

STATISTICAL LEARNING

Training data: $\{y_i, \mathbf{x}_i\}_1^N \sim p(\mathbf{x}, y)$

$\hat{R}(\mathbf{a}) = \frac{1}{N}\sum_{i=1}^N L\left(y_i,\, a_0 + \sum_{j=1}^n a_j x_{ij}\right)$

$\hat{\mathbf{a}} = \arg\min_{\mathbf{a}} \hat{R}(\mathbf{a})$

If not $N >> n$, not very good!

$R(\hat{\mathbf{a}}) >> R(\mathbf{a}^*)$ (high variance)

REGULARIZATION (biased learning)

$$\hat{\mathbf{a}}(\lambda) = \arg\min_{\mathbf{a}} [\hat{R}(\mathbf{a}) + \lambda \cdot P(\mathbf{a})]$$

$P(\mathbf{a}) \geq 0$ penalty

$0 \leq \lambda \leq \infty$ strength parameter

$\lambda \uparrow \Longrightarrow$ bias $\uparrow$ & variance $\downarrow$

$\hat{\mathbf{a}}(\lambda) \sim$ 1–dim. path of solutions $\in S^{n+1}$

$S^{n+1} =$ parameter space

MODEL SELECTION (λ)

$$\lambda^* = \arg\min_{0 \leq \lambda \leq \infty} R(\hat{\mathbf{a}}(\lambda))$$

Model selection criterion:

$\tilde{R}(\mathbf{a}) =$ surrogate for $R(\mathbf{a})$

depends on $L(y, F)$ & $P(\mathbf{a})$

$\hat{\lambda} = \arg\min_{0 \leq \lambda \leq \infty} \tilde{R}(\hat{\mathbf{a}}(\lambda))$

$\hat{\mathbf{a}}(\hat{\lambda}) =$ selected model

Cross–validation: any $L(y, F)$ & $P(\mathbf{a})$

PENALTY SELECTION

$\mathbf{a}^* =$ point in S^{n+1}

Choose penalty that induces paths that

on average come close to $\mathbf{a}^*$

$\{y_i, \mathbf{x}_i\}_1^N \sim p(\mathbf{x}, y)$

Depends on $\mathbf{a}^*$

Choose $P(\mathbf{a})$ based on knowledge of $\mathbf{a}^*$

SPARSITY

Fraction of non influential variables

$$S(\mathbf{a}) = \#(|\, a_k \,| \ << \mathsf{max}_j \,|\, a_j \,|)/n$$

$$\hat{\mathbf{a}} \simeq \mathbf{a}^* \Rightarrow S(\hat{\mathbf{a}}) \ \simeq S(\mathbf{a}^*)$$

$$\text{Choose } P(\mathbf{a}) \text{ s.t. } S(\hat{\mathbf{a}}(\lambda^*)) \simeq S(\mathbf{a}^*)$$

Don't know $S(\mathbf{a}^*)$

Family of penalties $P_\gamma(\mathbf{a})$: γ regulates $S(\hat{\mathbf{a}})$

bridging sparse $\longrightarrow$ dense

Model selection to jointly estimate (γ, λ)

("bridge regression": Frank & Friedman 1993)

POWER FAMILY

$$P_\gamma(\mathbf{a}) = \sum_{j=1}^{n} |\, a_j \,|^\gamma$$

With $L(y, F) = (y - F)^2$:

$\gamma = 2$: ridge–regression (dense)

$\gamma = 1$: lasso (moderately sparse)

$\gamma = 0$: (all) subsets selection (sparsest)

$0 \leq \gamma \leq 2$ bridges subset $\rightarrow$ ridge

Note: $\gamma \geq 1 \Rightarrow$ convex, $\gamma < 1 \Rightarrow$ non convex

Generalized Elastic Net

$1 \leq \beta \leq 2$ (convex: lasso $\rightarrow$ ridge):

Elastic Net (Zou & Hastie 2005)

$$P_\beta(\mathbf{a}) = \sum_{j=1}^{n} (\beta - 1)\, a_j^2/2 + (2-\beta)\,|\, a_j \,|$$

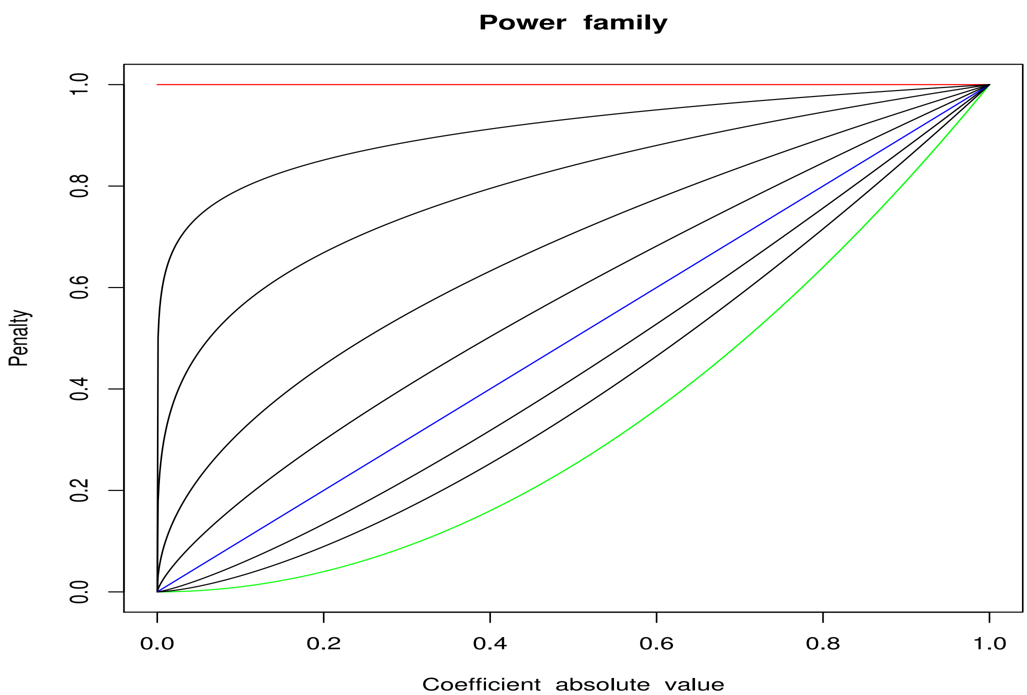
$0 \leq \beta < 1$ (non convex: subset selection $\rightarrow$ lasso):

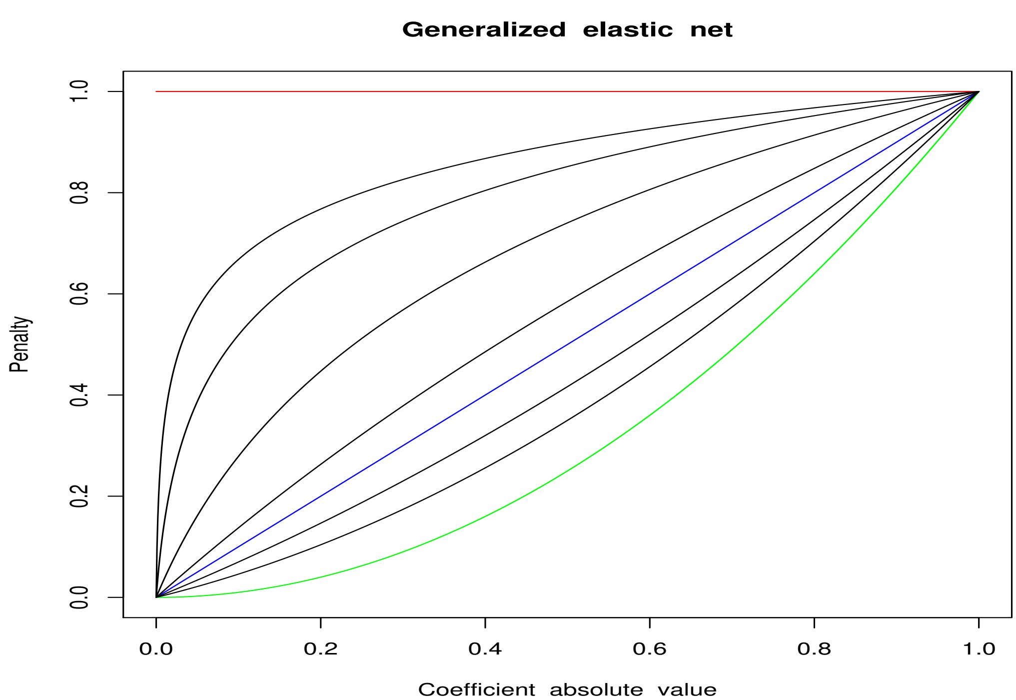
$$P_\beta(\mathbf{a}) = \sum_{j=1}^{n} \log((1-\beta)\,|\, a_j \,| + \beta)$$

$0 \leq \beta \leq 2$ bridges subset $\rightarrow$ ridge

Better statistical & computational properties

Method works for both


Power family
Penalty
Coefficient absolute value
0.0
0.2
0.4
0.6
0.8
1.0


Generalized elastic net
Penalty
Coefficient absolute value
0.0
0.2
0.4
0.6
0.8
1.0

BRIDGE REGRESSION

(1) Repeatedly solve:

$$\hat{\mathbf{a}}_{\beta}(\lambda) = \text{arg min}_{\mathbf{a}}[\hat{R}(\mathbf{a}) + \lambda \cdot P_{\beta}(\mathbf{a})]$$

$$0 \leq \beta \leq 2, \quad 0 \leq \lambda \leq \infty$$

(2) $(\hat{\beta}, \hat{\lambda}) \leftarrow$ model selection criterion

(3) $\hat{\mathbf{a}}_{\hat{\beta}}(\hat{\lambda}) =$ solution

Big challenge: fast enough algorithm for (1)

Especially for $P_{\beta}(\mathbf{a}) =$ non convex

DIRECT PATH SEEKING

Goal: rapidly produce path $\simeq$ given $P(\mathbf{a})$

without repeatedly optimizing

EXAMPLES

$L(y, F) = (y - F)^2$:

PLS $\simeq$ ridge–regression ($\beta = 2$)

LAR $\simeq$ lasso ($\beta = 1$)

Forward stepwise $\simeq$ all–subsets ($\beta = 0$)

Any convex $L(y, F)$:

Gradient boosting $\simeq$ lasso ($\beta = 1$)

Want bridge regression: $0 \leq \beta \leq 2$

Generalized Path Seeking (GPS)

Fast algorithm for:

(1) any $P(\mathbf{a})$ s.t. $\frac{\partial P(\mathbf{a})}{\partial \,|\, a_j \,|} \geq 0$

i.e. $P(\mathbf{a})$ monotone $\uparrow \,|\, a_j \,|$

power family

generalized elastic net family

SCAD (Fan & Li)

many more

(2) any convex $L(y, F)$ (some non convex)

Definitions

$\nu \geq 0$: path length

$\Delta\nu > 0$: small increment

$$g_j(\nu) = -\left[\frac{\partial \hat{R}(\mathbf{a})}{\partial a_j}\right]_{\mathbf{a}=\hat{\mathbf{a}}(\nu)}$$

$$p_j(\nu) = \left[\frac{\partial P(\mathbf{a})}{\partial \,|\, a_j \,|}\right]_{\mathbf{a}=\hat{\mathbf{a}}(\nu)}$$

$$\lambda_j(\nu) = g_j(\nu) \,/ p_j(\nu)$$

GPS algorithm

Initialize: $\nu = 0$; $\{\hat{a}_j(0) = 0\}_1^n$

Loop $\{$

  Compute $\{\lambda_j(\nu)\}_1^n$

  $S = \{j \mid \lambda_j(\nu) \cdot \hat{a}_j(\nu) < 0\}$

  if $(S = \text{ empty})$ $j^* = \arg\max_j |\lambda_j(\nu)|$

  else $j^* = \arg\max_{j \in S} |\lambda_j(\nu)|$

  $\hat{a}_{\,j^*}(\nu + \Delta\nu) = \hat{a}_{\,j^*}(\nu) + \Delta\nu \cdot sign(\lambda_{j^*}(\nu))$

  $\{\hat{a}_{\,j}(\nu + \Delta\nu) = \hat{a}_{\,j}(\nu)\}_{j \neq j^*}$

  $\nu \leftarrow \nu + \Delta\nu$

$\}$ Until $\lambda(\nu) = 0$

THEOREM

$\hat{\mathbf{a}}(\lambda) =$ exact path

$\hat{\mathbf{a}}(\nu) =$ GPS path

If for all $\lambda > \lambda_0$

all $\{\hat{a}_j(\lambda)\}_1^n$ are continuous and monotone

Then for all $\lambda > \lambda_0$

$\hat{\mathbf{a}}(\nu) = \hat{\mathbf{a}}(\lambda)$, as $\boldsymbol{\Delta}\nu \to 0$

i.e. GPS produces exact path

Otherwise: $\hat{a}_j(\nu) \simeq \hat{a}_j(\lambda)$

When $\hat{a}_j(\lambda)$ becomes non monotone:

$\hat{a}_j(\nu)$ tends to slightly delay becoming non monotone

When $\hat{a}_j(\lambda)$ discontinuous ($\gamma < 1$, $\beta < 1/2$):

$\hat{a}_j(\nu) =$ continuous

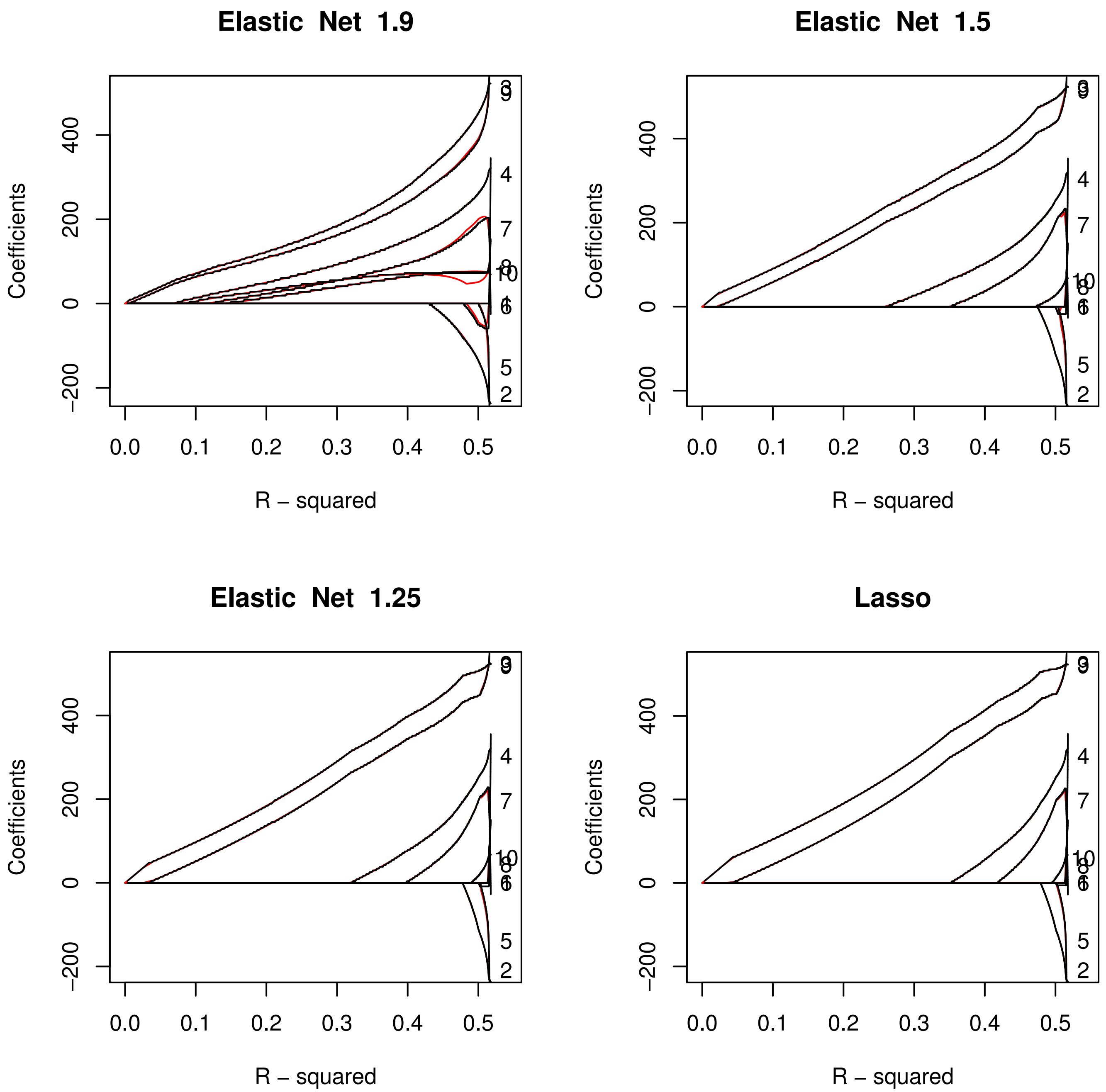
$\sim$ interpolates between $\hat{a}_j(\lambda)$ discontinuities

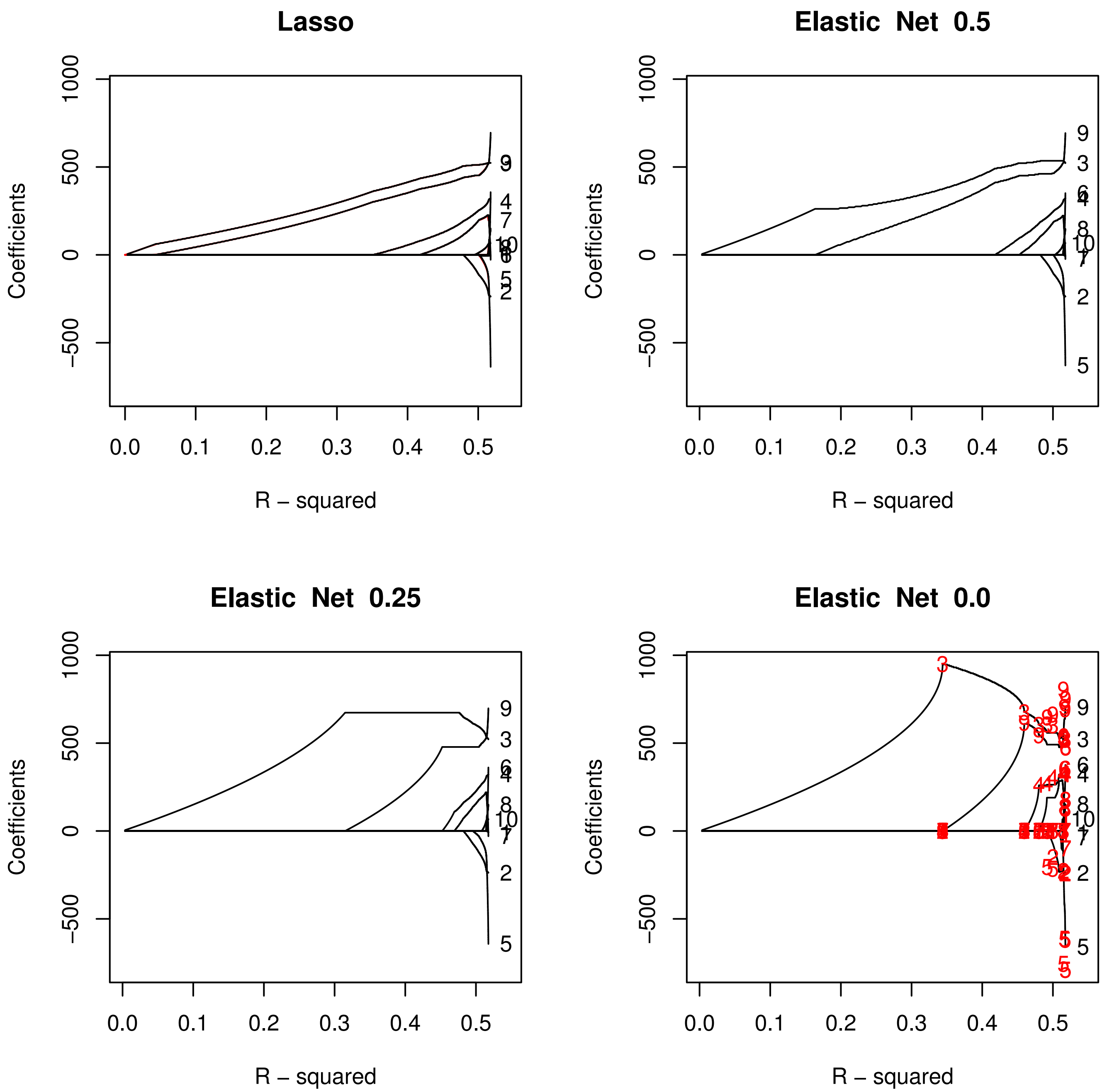
Regression: $L(y, F) = (y - F)^2$

Diabetes data: $n = 10, \;\; N = 442$

Used in LARS (Efron *et al* 2002)

red $=$ exact (convex), black $=$ GPS


Elastic Net 1.9
Coefficients
R − squared
−200 0 200 400
0.0 0.1 0.2 0.3 0.4 0.5
3 9 4 7 8 10 1 6 5 2
Elastic Net 1.5
Coefficients
R − squared
−200 0 200 400
0.0 0.1 0.2 0.3 0.4 0.5
3 9 4 7 10 8 1 6 5 2
Elastic Net 1.25
Coefficients
R − squared
−200 0 200 400
0.0 0.1 0.2 0.3 0.4 0.5
3 9 4 7 10 8 1 6 5 2
Lasso
Coefficients
R − squared
−200 0 200 400
0.0 0.1 0.2 0.3 0.4 0.5
3 9 4 7 10 8 1 6 5 2


Lasso
Elastic Net 0.5
Elastic Net 0.25
Elastic Net 0.0
Coefficients
R − squared
1000
500
0
−500
0.0
0.1
0.2
0.3
0.4
0.5

Regression: under-determined example

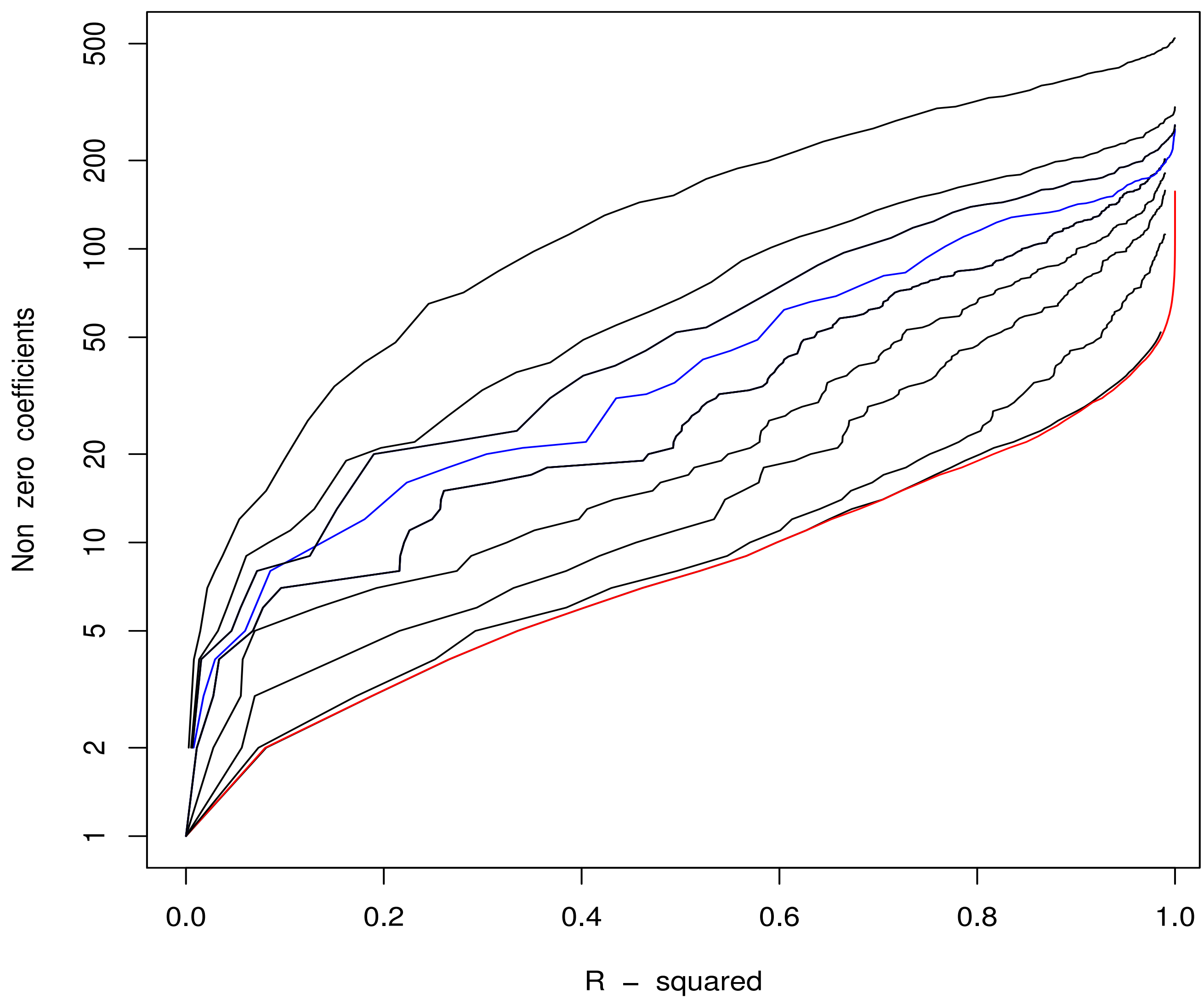
$$n = 10000, \quad N = 200$$

$$\mathbf{x}_i \sim N(0, \mathbf{C}); \quad C_{jj} = 1, \; C_{jk} = 0.4$$

$$y_i = \sum_{j=1}^{n} a_j^* x_{ij} + \varepsilon_i$$

$$\varepsilon_i \sim N(0, \sigma^2); \quad \sigma \sim 3/1 \;\; \text{signal/noise}$$

$$|\, a_j^*| = [31 - j]_+, \quad sign(a_{j+1}^*) = -sign(a_j^*)$$



$\beta \in \{$1.9, 1.7, 1.5, 1.0 (lasso, blue), 0.5, 0.3, 0.2, 0.1, 0.0, step (red)$\}$

THEREFORE

$P_\beta(\mathbf{a}) =$ generalized elastic net

$\beta \downarrow \Rightarrow S(\hat{\mathbf{a}}) \uparrow$ monotonically

at all path points

Penalty Selection (β)

Regression: under-determined example

$n = 10000, \; N = 200$

50 data sets $\sim p(\mathbf{x}, y)$

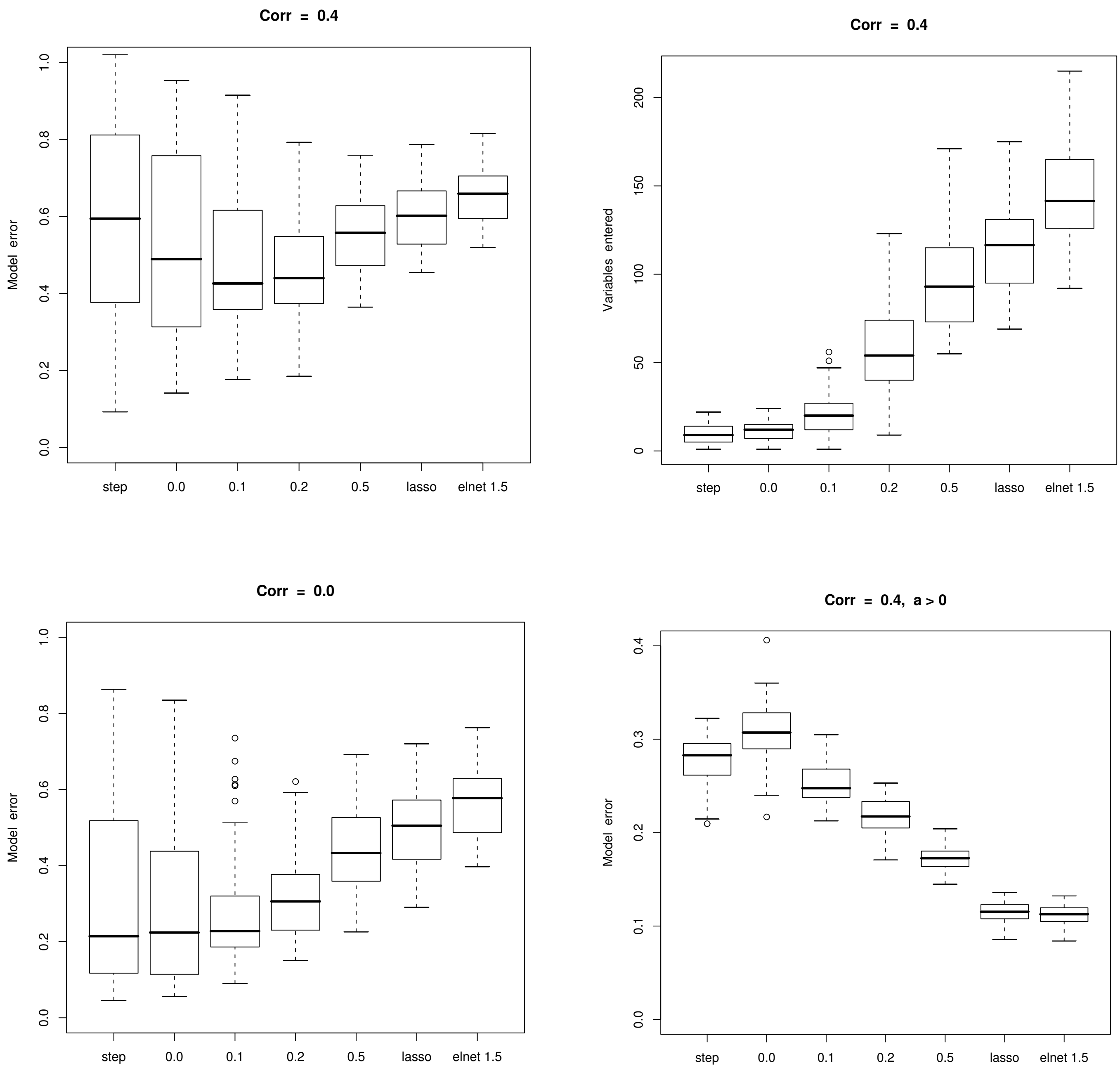
Distribution of closest "distance" to truth $\mathbf{a}^*$ (risk)

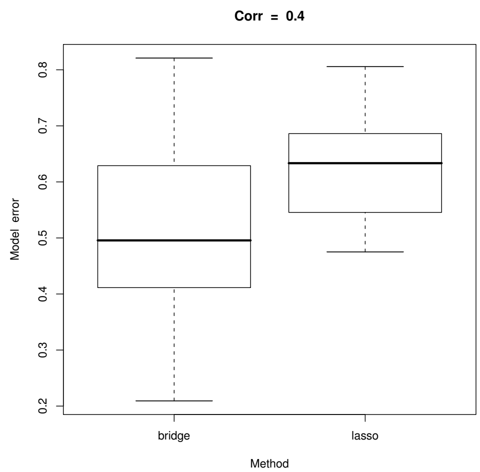
Methods:

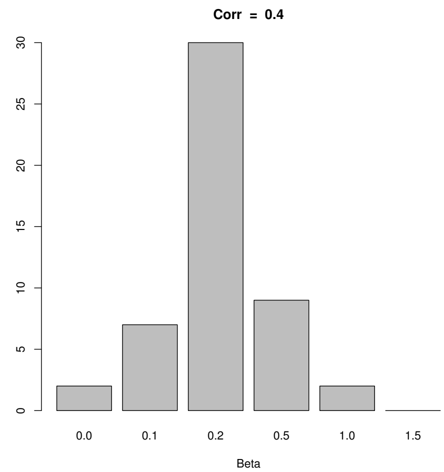
forward stepwise

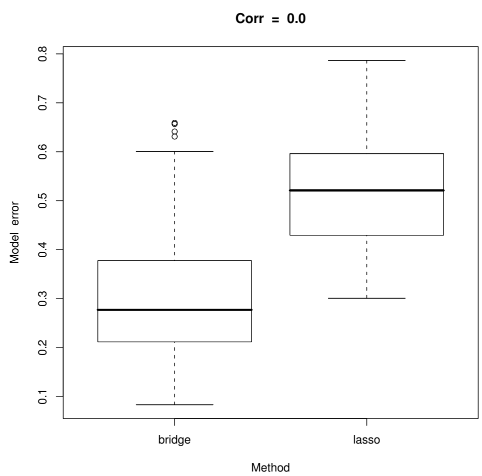
GPS: $\beta \in \{0.0, 0.1, 0.2, 0.5\}$

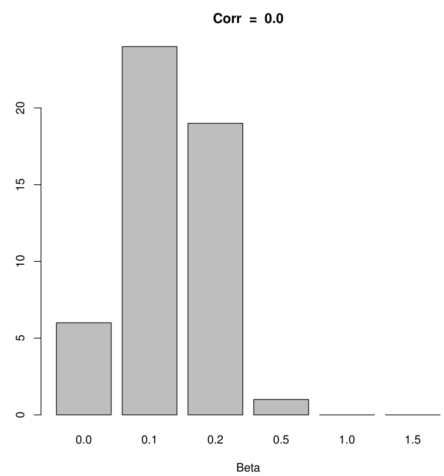
lasso, elastic net (1.5)

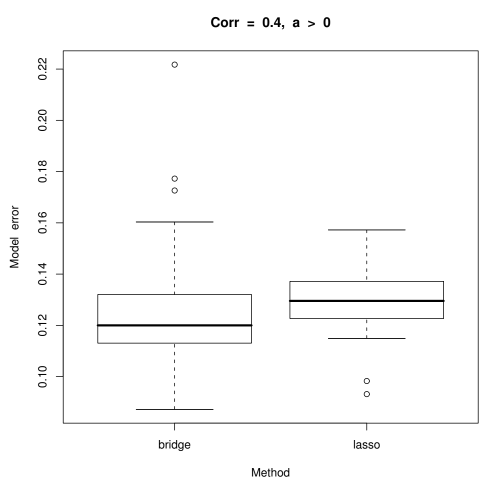

Corr = 0.4
Model error
step
0.0
0.1
0.2
0.5
lasso
elnet 1.5
Corr = 0.4
Variables entered
Corr = 0.0
Model error
Corr = 0.4, a > 0
Model error

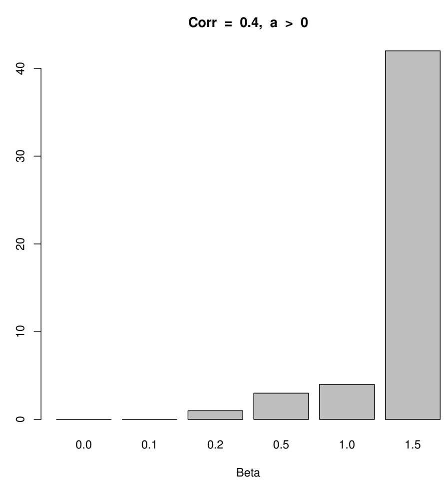

Corr = 0.4
Model error
0.2
0.3
0.4
0.5
0.6
0.7
0.8
bridge
lasso
Method


Corr = 0.4
0
5
10
15
20
25
30
0.0
0.1
0.2
0.5
1.0
1.5
Beta


Corr = 0.0
Model error
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
bridge
lasso
Method


Corr = 0.0
0
5
10
15
20
0.0
0.1
0.2
0.5
1.0
1.5
Beta


Corr = 0.4, a > 0
Model error
0.10
0.12
0.14
0.16
0.18
0.20
0.22
bridge
lasso
Method


Corr = 0.4, a > 0
0
10
20
30
40
0.0
0.1
0.2
0.5
1.0
1.5
Beta

Post-processing Selectors

(1) $\tilde{\mathbf{a}}(\lambda) = \arg\min_{\mathbf{a}} \hat{R}(\mathbf{a}) + \lambda\, P(\mathbf{a})$

$$P(\mathbf{a}) = \text{convex (lasso)}$$

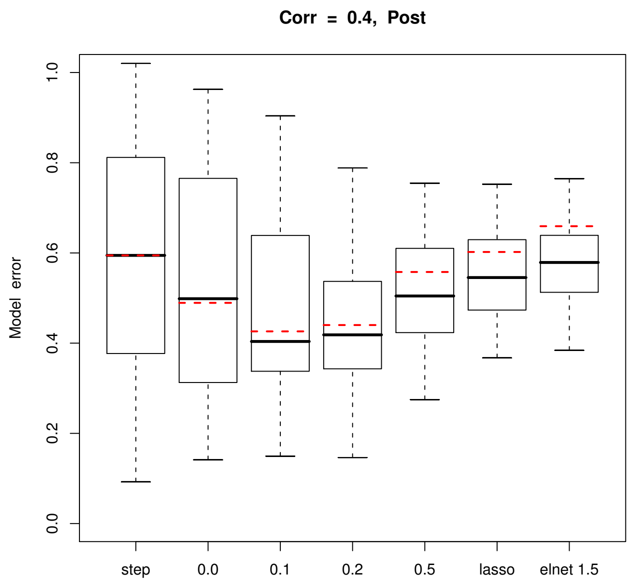
(2) $A(\lambda) = \{j\}_{\tilde{a}_j(\lambda) \neq 0}$ (active variables at λ)

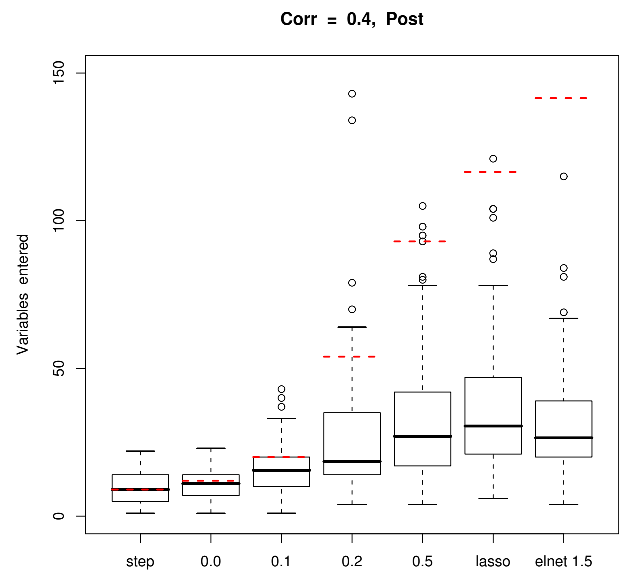
(3) $\hat{\mathbf{a}}(\lambda) = \arg\min_{\mathbf{a}} \hat{R}(\mathbf{a})$ s.t. $\{a_j = 0\}_{j \notin A(\lambda)}$

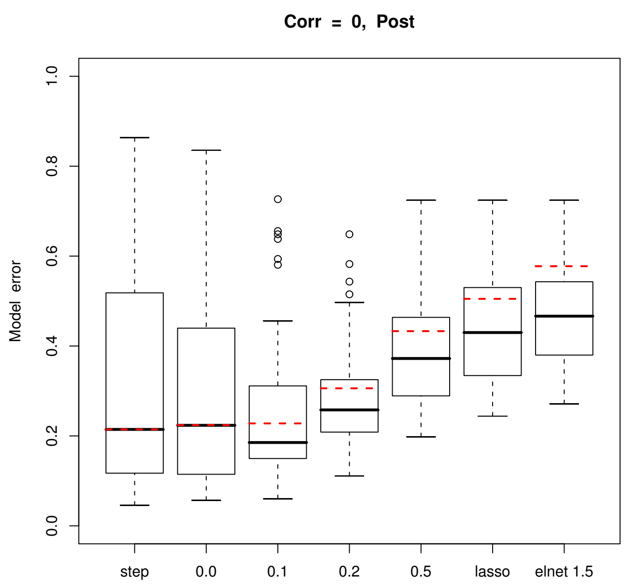
Intuition (sparse problems):

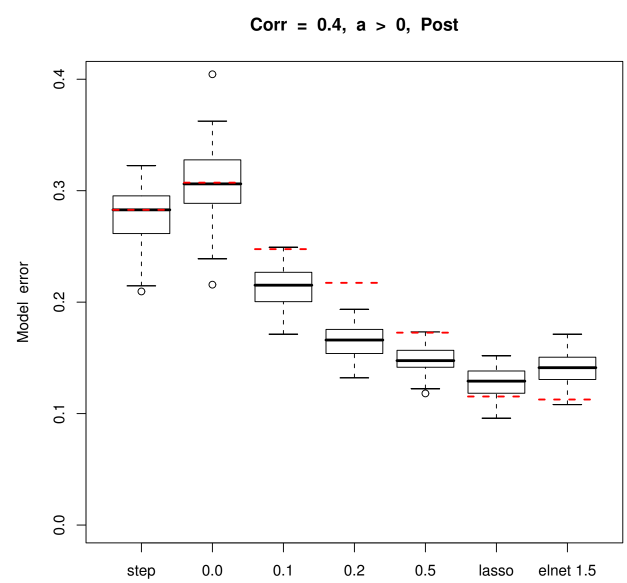
$\tilde{\mathbf{a}}(\lambda) \simeq$ selects correct variables

but over shrinks their values


Corr = 0.4, Post
Model error
0.0
0.2
0.4
0.6
0.8
1.0
step
0.0
0.1
0.2
0.5
lasso
elnet 1.5


Corr = 0.4, Post
Variables entered
0
50
100
150
step
0.0
0.1
0.2
0.5
lasso
elnet 1.5


Corr = 0, Post
Model error
0.0
0.2
0.4
0.6
0.8
1.0
step
0.0
0.1
0.2
0.5
lasso
elnet 1.5


Corr = 0.4, a > 0, Post
Model error
0.0
0.1
0.2
0.3
0.4
step
0.0
0.1
0.2
0.5
lasso
elnet 1.5

CONCLUSIONS

(1) best penalty (prior) for $\{|\, a_j \,|\}$ depends on

$\{|\, a_j^* \,|, sign(a_j^*)\}$ and $\mathbf{x}$ – distribution

(2) need bridge regression to chose $(\hat{\beta}, \hat{\lambda})$

(3) when sparse non convex $P(\mathbf{a})$ is best:

better variable *selection* & shrinkage

(4) best direct methods $\longrightarrow$ best selectors

(5) results same for logistic regression

Generalized Path Seeking

For same $L(y, F)$ & $P(\mathbf{a})$:

paths close to exact solutions

same sparseness properties

Can be applied with:

non convex $P(\mathbf{a}) \Rightarrow$ sparser than lasso

any convex $L(y, F)$, some non convex

Used as selector $\rightarrow$ further improvement

Multinomial regression

Speed

$n = 10000$, $N = 200$

Solutions at 500 path points:

$\simeq$ 0.5 sec. ($\beta \sim$ non/convex)

Bridge regression:

6 $\beta-$ values $\times$ 10–fold xval: ˜ 30 sec

equivalent to solving 30000 optimization problems

(most non convex)

Computation scales $\sim$ linearly with n & N ($n >> N$)

TALK

http://www-stat.stanford.edu/~jhf/talks/GPStalk.pdf

PAPER

http://www-stat.stanford.edu/~jhf/ftp/GPSpaper.pdf