Least Angle Regression, Forward Stagewise and the Lasso

Brad Efron, Trevor Hastie, Iain Johnstone and Robert Tibshirani
Stanford University

http://www-stat.stanford.edu/∼hastie/Papers/#LARS

Background

- Today's talk is about linear regression
- But the motivation comes from the area of flexible function fitting: "Boosting"— Freund & Schapire (1995)

Least Squares Boosting

Friedman, Hastie & Tibshirani — see *Elements of Statistical Learning (chapter 10)*

Supervised learning: Response y, predictors $x = (x_1, x_2 \ldots x_p)$.

1. Start with function $F(x) = 0$ and residual $r = y$
2. Fit a CART regression tree to r giving $f(x)$
3. Set $F(x) \leftarrow F(x) + \epsilon f(x)$, $r \leftarrow r - \epsilon f(x)$ and repeat step 2 many times

Least Squares Boosting

Linear Regression

Here is a version of least squares boosting for multiple linear regression: (assume predictors are standardized)

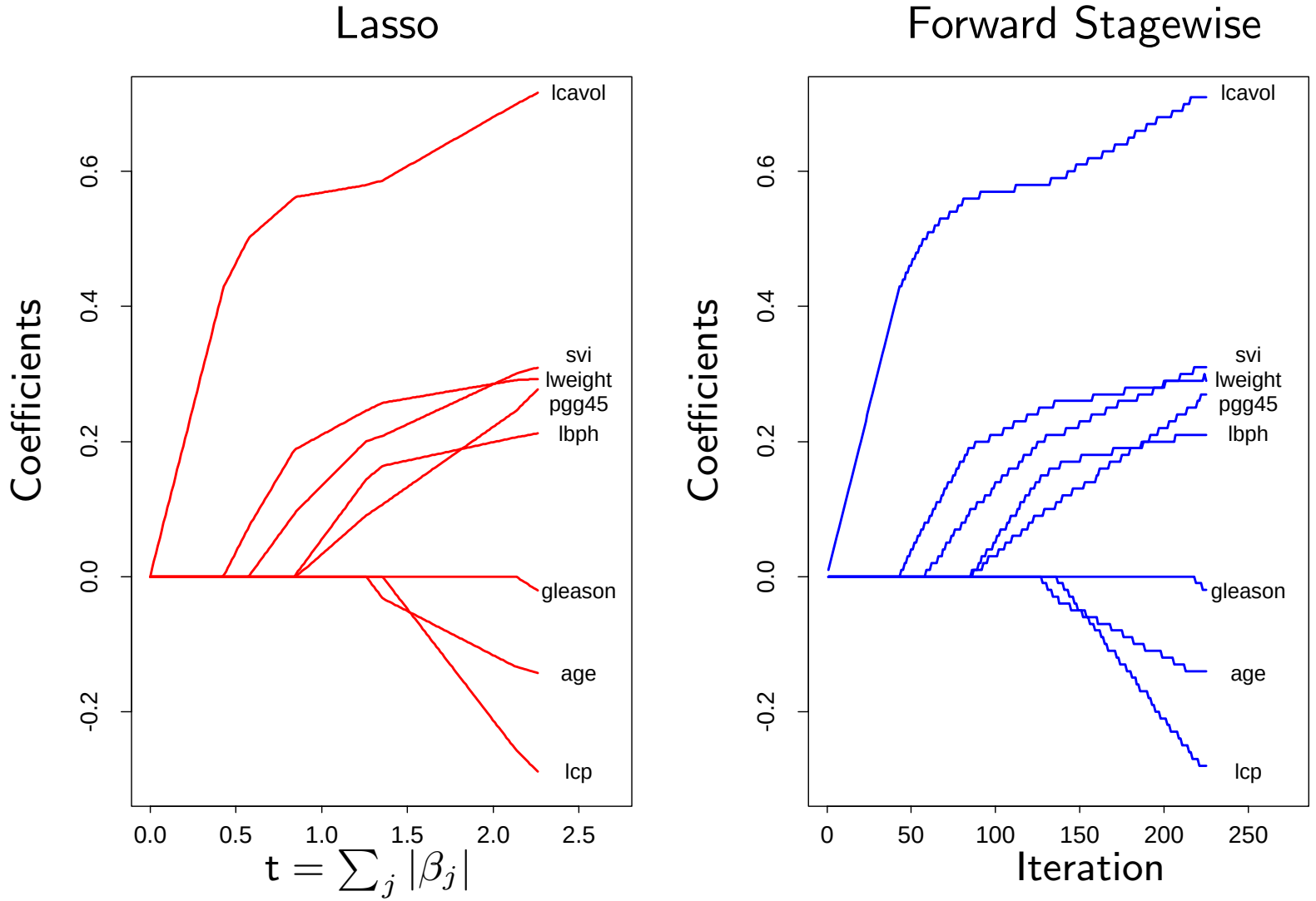
(Incremental) Forward Stagewise

1. Start with $r = y$, $\beta_1, \beta_2, \ldots \beta_p = 0$.
2. Find the predictor x_j most correlated with r
3. Update $\beta_j \leftarrow \beta_j + \delta_j$, where $\delta_j = \epsilon \cdot \text{sign}\langle r, x_j \rangle$
4. Set $r \leftarrow r - \delta_j \cdot x_j$ and repeat steps 2 and 3 many times

$\delta_j = \langle r, x_j \rangle$ gives usual forward stagewise; different from forward stepwise

Analogous to least squares boosting, with *trees=predictors*

Prostate Cancer Data



Linear regression via the Lasso (Tibshirani, 1995)

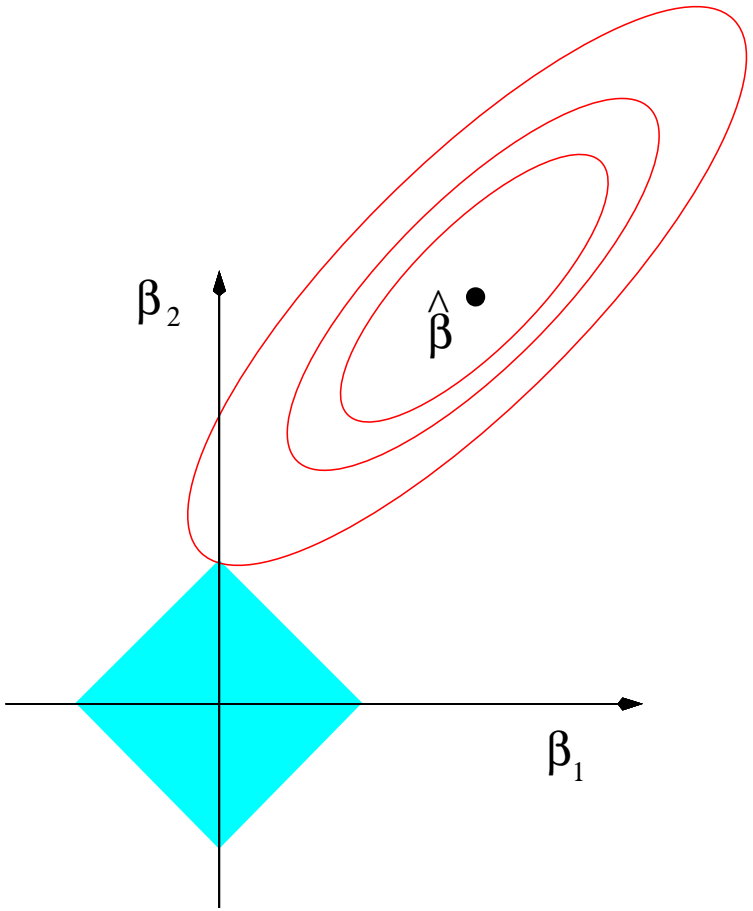
- Assume $\bar{y} = 0$, $\bar{x}_j = 0$, $\text{Var}(x_j) = 1$ for all j.
- Minimize $\sum_i (y_i - \sum_j x_{ij}\beta_j)^2$ subject to $\sum_j |\beta_j| \leq s$
- With orthogonal predictors, solutions are soft thresholded version of least squares coefficients:

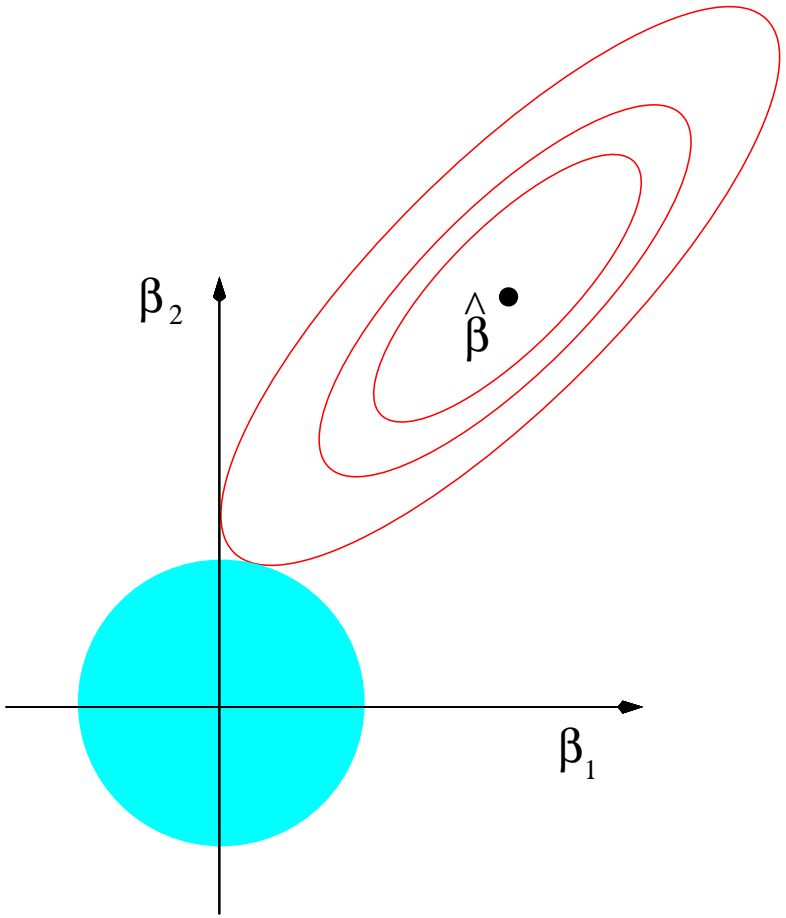
$$\text{sign}(\hat{\beta}_j)(|\hat{\beta}_j| - \gamma)_+$$

(γ is a function of s)

- For small values of the bound s, Lasso does variable selection. See pictures

Lasso and Ridge regression

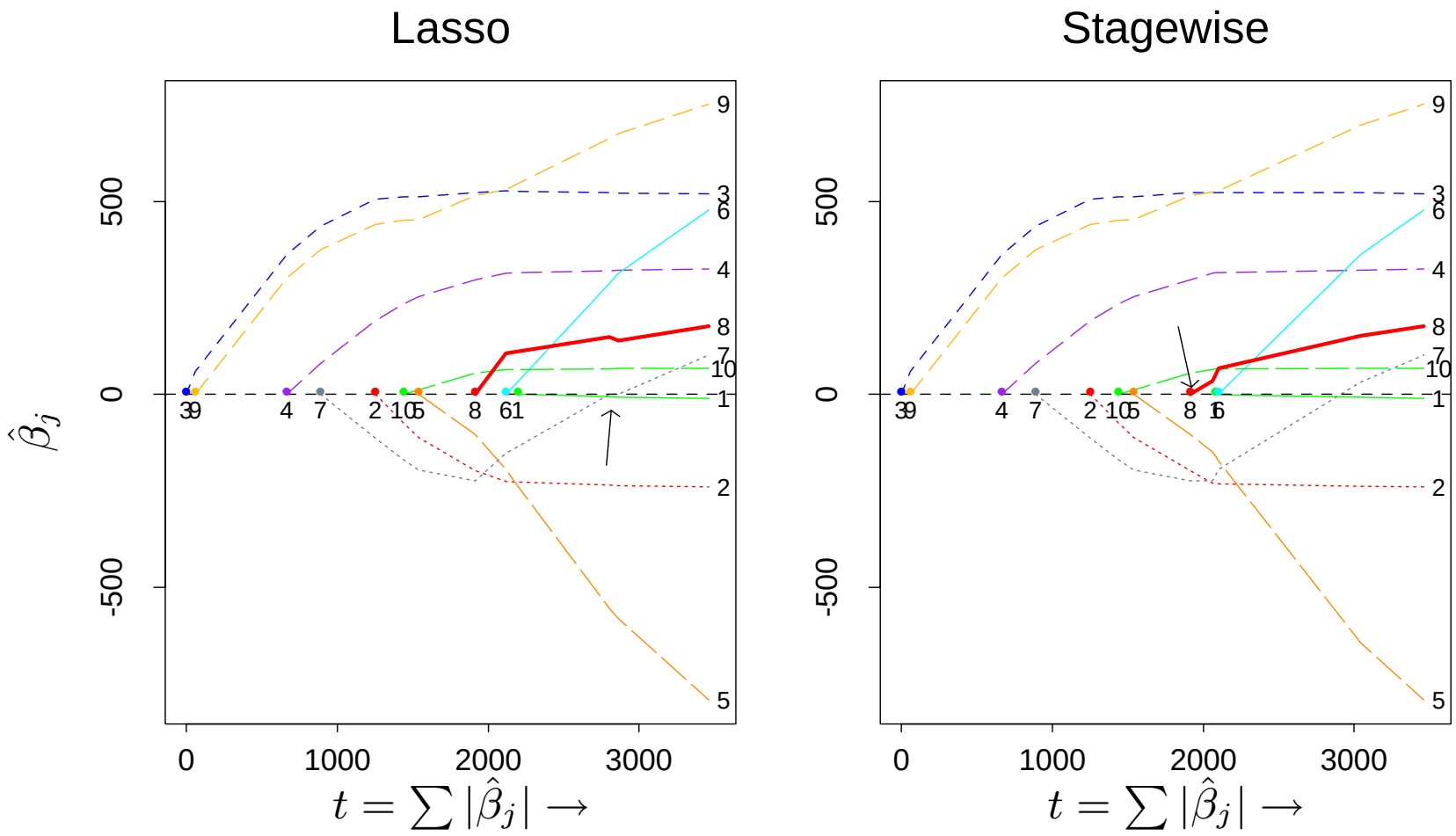




More on Lasso

- Current implementations use quadratic programming to compute solutions
- Can be applied when $p > n$. In that case, number of non-zero coefficients is at most $n - 1$ (by convex duality)
- interesting consequences for applications, eg microarray data

Diabetes Data



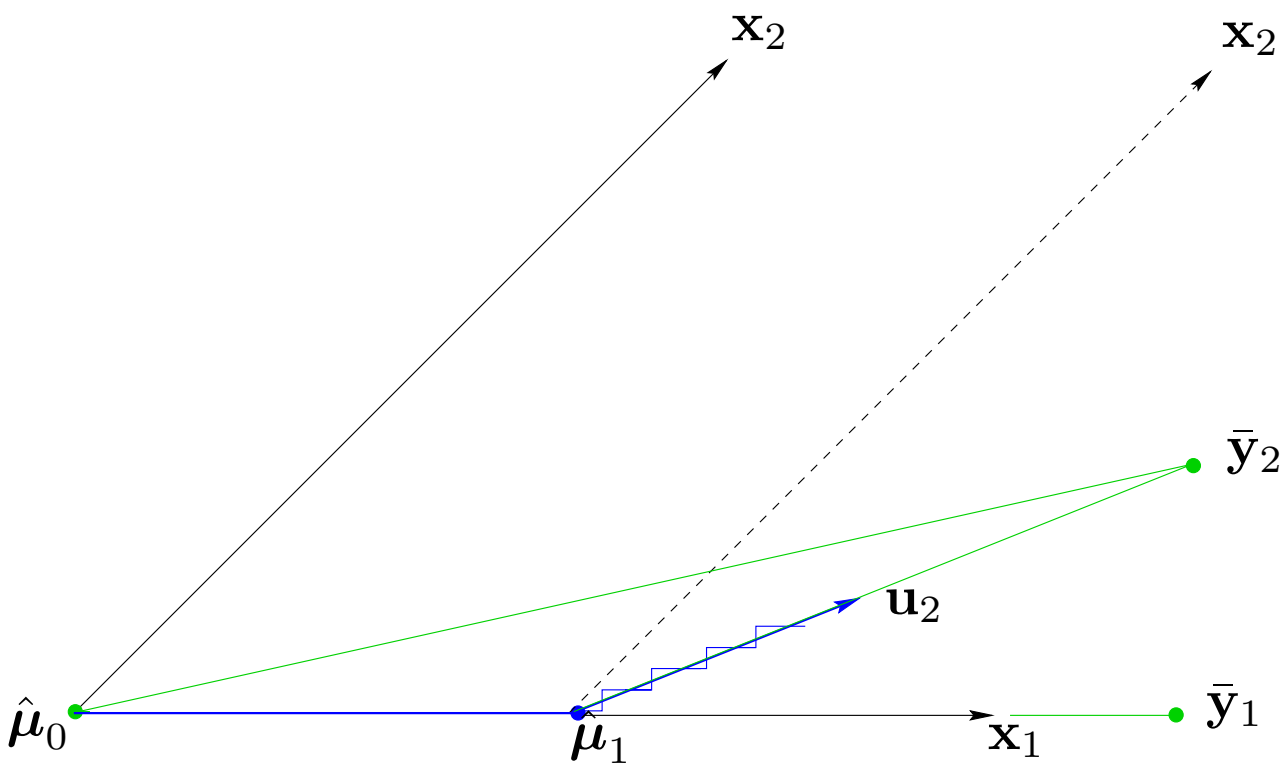
Why are Forward Stagewise and Lasso so similar?

- Are they identical?
- In orthogonal predictor case: *yes*
- In hard to verify case of *monotone* coefficient paths: *yes*
- In general, almost!
- Least angle regression (LAR) provides answers to these questions, and an efficient way to compute the complete Lasso sequence of solutions.

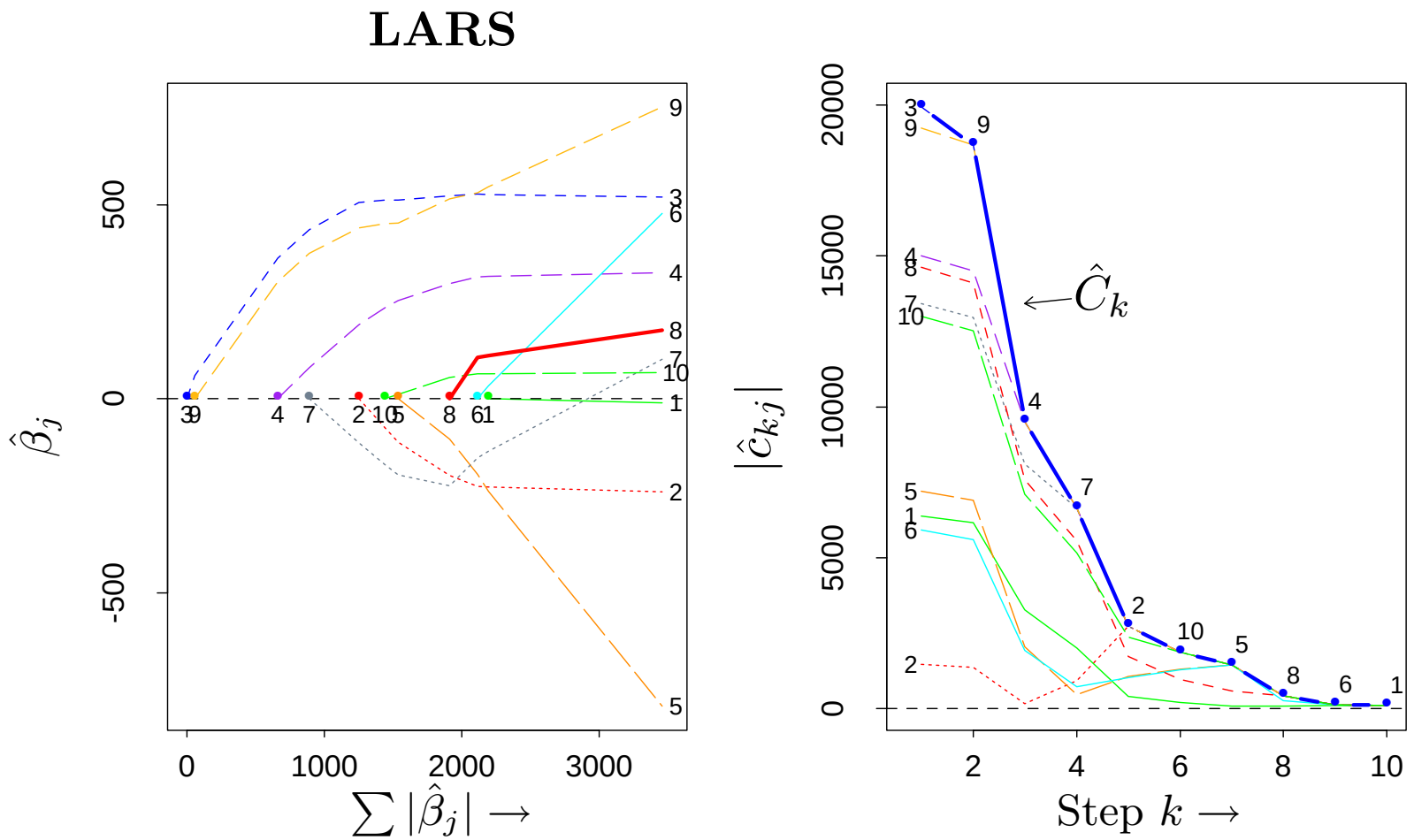
Least Angle Regression — LAR

Like a "more democratic" version of forward stepwise regression.

1. Start with $r = y$, $\hat{\beta}_1, \hat{\beta}_2, \ldots \hat{\beta}_p = 0$. Assume x_j standardized.
2. Find predictor x_j most correlated with r.
3. Increase β_j in the direction of sign(corr(r, x_j)) until some other competitor x_k has as much correlation with current residual as does x_j.
4. Move $(\hat{\beta}_j, \hat{\beta}_k)$ in the joint least squares direction for (x_j, x_k) until some other competitor x_ℓ has as much correlation with the current residual
5. Continue in this way until all predictors have been entered. Stop when corr$(r, x_j) = 0 \; \forall \; j$, i.e. OLS solution.



The LAR direction $\mathbf{u}_2$ at step 2 makes an equal angle with $\mathbf{x}_1$ and $\mathbf{x}_2$.


LARS
$\hat{\beta}_j$
500
0
-500
0
1000
2000
3000
$\sum |\hat{\beta}_j| \rightarrow$
$|\hat{C}_{kj}|$
20000
15000
10000
5000
0
$\leftarrow \hat{C}_k$
2
4
6
8
10
Step $k \rightarrow$

Relationship between the 3 algorithms

- Lasso and forward stagewise can be thought of as restricted versions of LAR
- *For Lasso*: Start with LAR. If a coefficient crosses zero, stop. Drop that predictor, recompute the best direction and continue. This gives the Lasso path

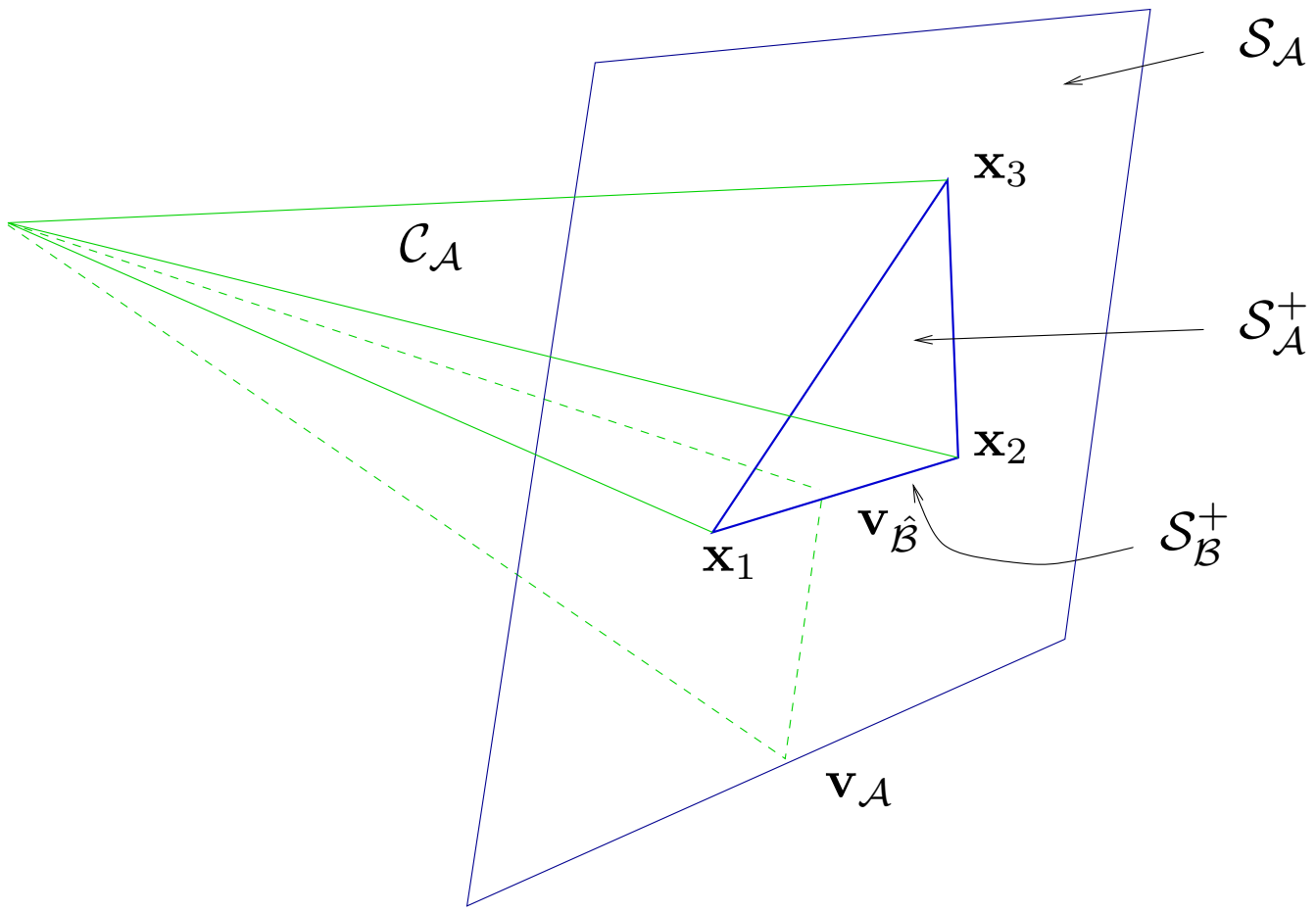
Proof (lengthy): use Karush-Kuhn-Tucker theory of convex optimization. Informally:

$$\frac{\partial}{\partial \beta_j}\left\{||\mathbf{y}-\mathbf{X}\beta||^2 \quad + \quad \lambda\sum_j |\beta_j|\ \right\} = 0$$

$$\Leftrightarrow$$

$$\langle \mathbf{x}_j, \mathbf{r}\rangle \quad = \quad \frac{\lambda}{2}\text{sign}(\hat{\beta}_j) \quad \text{if } \hat{\beta}_j \neq 0 \text{ (active)}$$

- *For forward stagewise:* Start with LAR. Compute best (equal angular) direction at each stage. If direction for any predictor j doesn't agree in sign with $\text{corr}(r, x_j)$, project direction into the "positive cone" and use the projected direction instead.
- in other words, forward stagewise always moves each predictor in the direction of $\text{corr}(r, x_j)$.
- The incremental forward stagewise procedure approximates these steps, one predictor at a time. As step size $\epsilon \to 0$, can show that it coincides with this modified version of LAR



The forward stagewise direction lies in the positive cone spanned by the (signed) predictors with equal correlation with the current residual.

Summary

- LARS—uses least squares directions in the active set of variables.
- Lasso—uses least square directions; if a variable crosses zero, it is removed from the active set.
- Forward stagewise—uses non-negative least squares directions in the active set.

Benefits

- Possible explanation of the benefit of "slow learning" in boosting: it is approximately fitting via an L_1 (lasso) penalty
- new algorithm computes entire Lasso path in same order of computation as one full least squares fit. Splus/R Software on Hastie's website:
  www-stat.stanford.edu/∼hastie/Papers#LARS
- Degrees of freedom formula for LAR:
  After k steps, degrees of freedom of fit $= k$ (with some regularity conditions)
- For Lasso, the procedure often takes $> p$ steps, since predictors can drop out. Corresponding formula (conjecture):
  Degrees of freedom for last model in sequence with k predictors is equal to k.

LARS and Boosting

Recent work with Saharon Rosset and Ji Zhu:

- extends the connections between Forward Stagewise and L_1 penalized fitting to other loss functions. In particular the Exponential loss of Adaboost, and the Binomial loss of Logitboost.
- In the separable case, L_1 regularized fitting with these losses converges to a L_1 maximizing margin (defined by β^*), as the penalty disappears. i.e. if

$$\beta(t) = \arg\min L(y, f) \quad \text{s.t. } |\beta| \leq t,$$

then

$$\lim_{t\uparrow\infty} \frac{\beta(t)}{|\beta(t)|} \rightarrow \beta^*$$

LARS and Boosting (continued)

- makes connections between SVMs and Boosting, and makes explicit the margin maximizing properties of boosting.
- experience from statistics suggests that some $\beta(t)$ along the path might perform better—a.k.a stopping early.
- Alternatively, using the "Hinge loss" of SVMs and an L_1 penalty (rather than quadratic), we get a *Lasso* version of SVMs (with at most N variables in the solution for any value of the penalty.

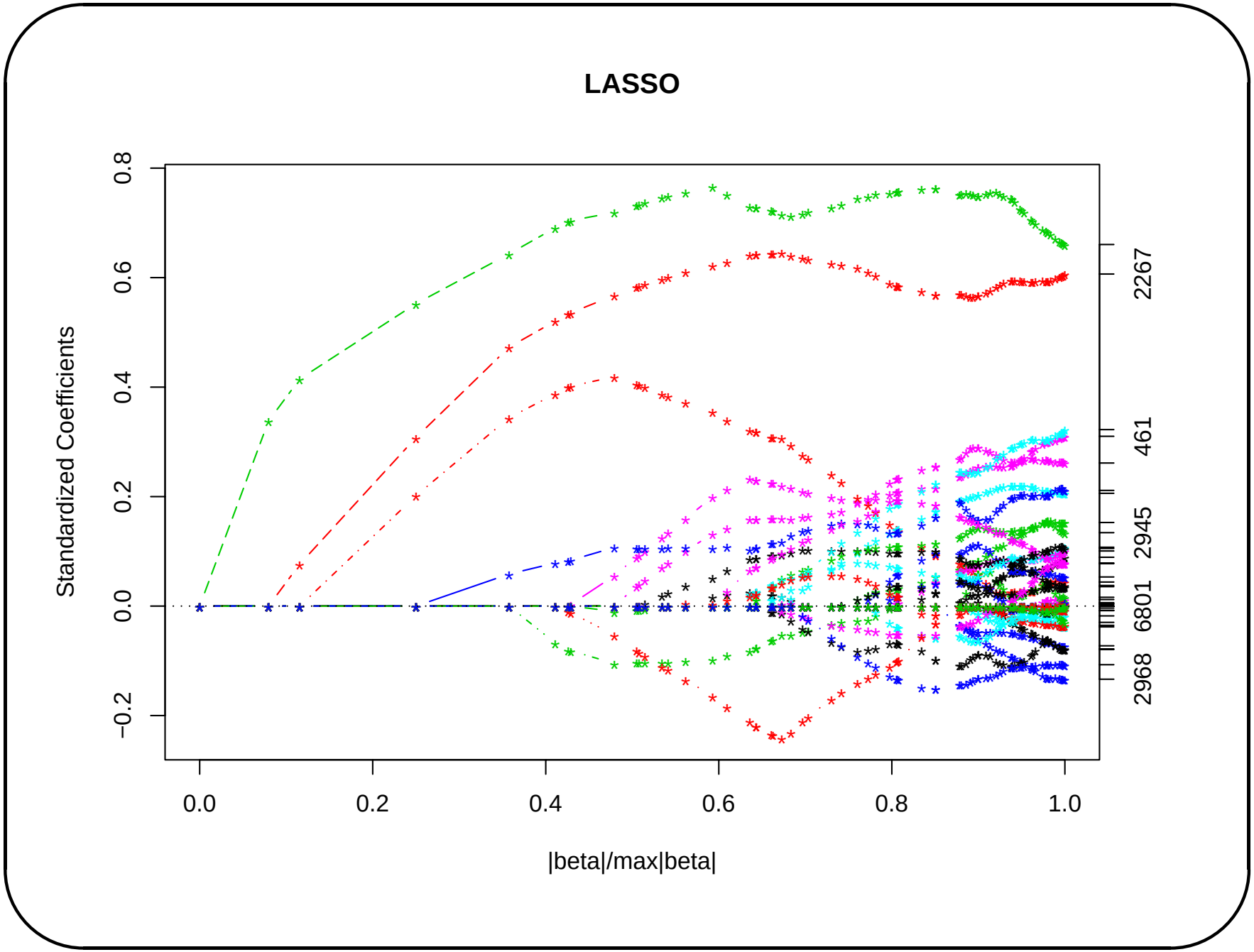
Software for R and Splus

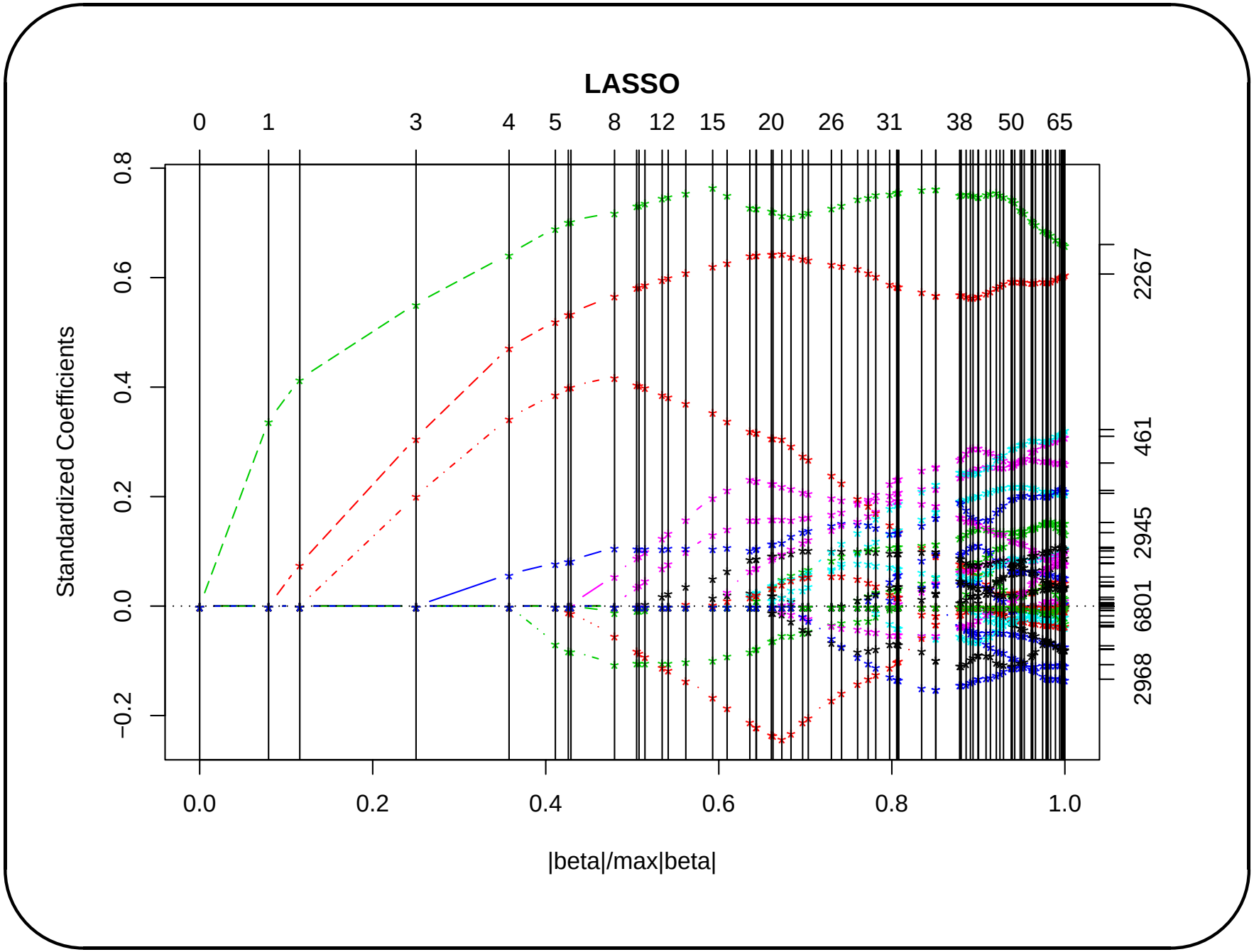
`lars()` function fits all three models: `lasso`, `lar` or `forward.stagewise`. Methods for prediction, plotting, and cross-validation. Detailed documentation provided. Visit
www-stat.stanford.edu/∼hastie/Papers/#LARS

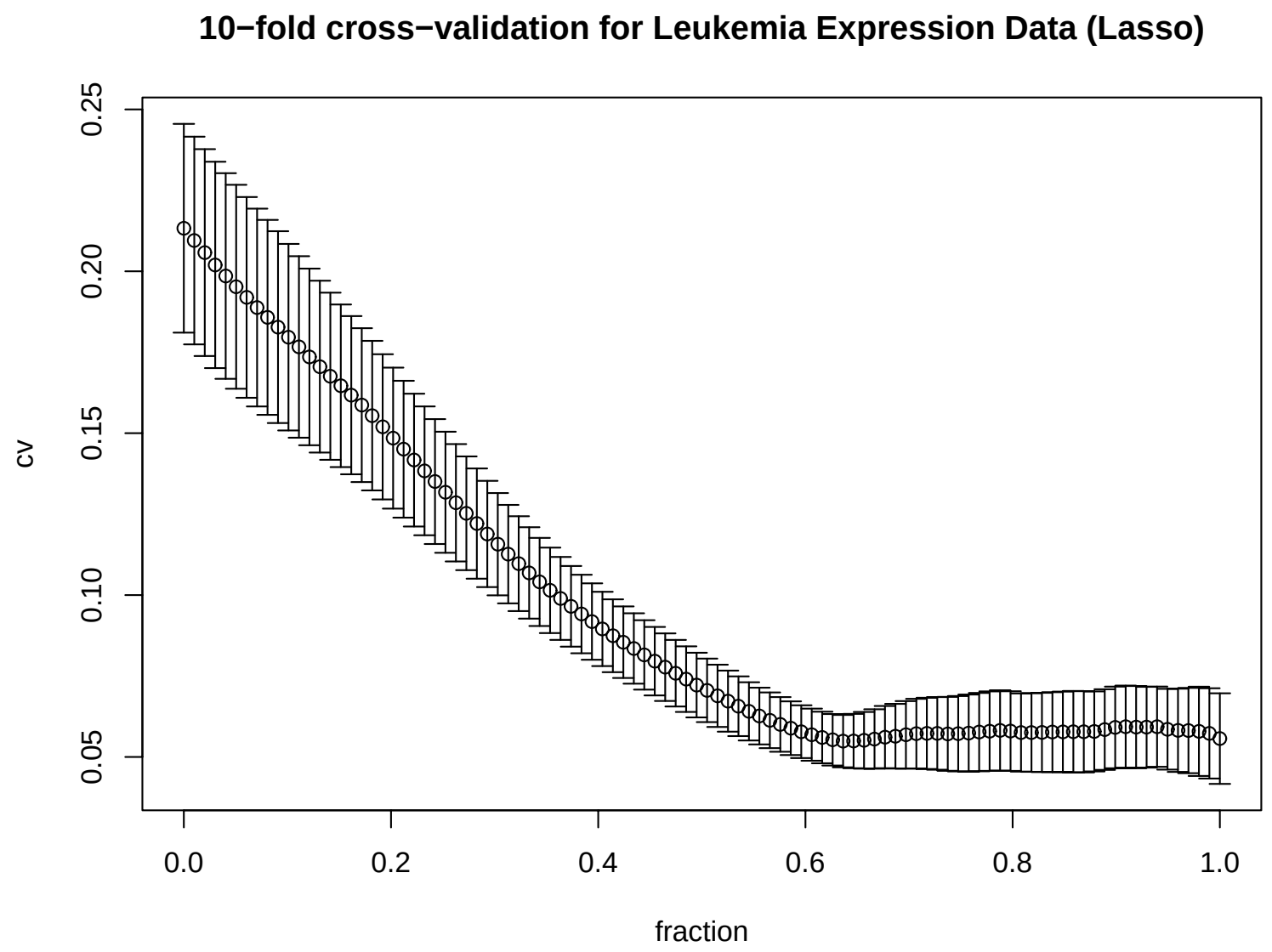
Main computations involve least squares fitting using the *active set* of variables. Computations managed by updating the Choleski R matrix (and frequent downdating for lasso and forward stagewise).

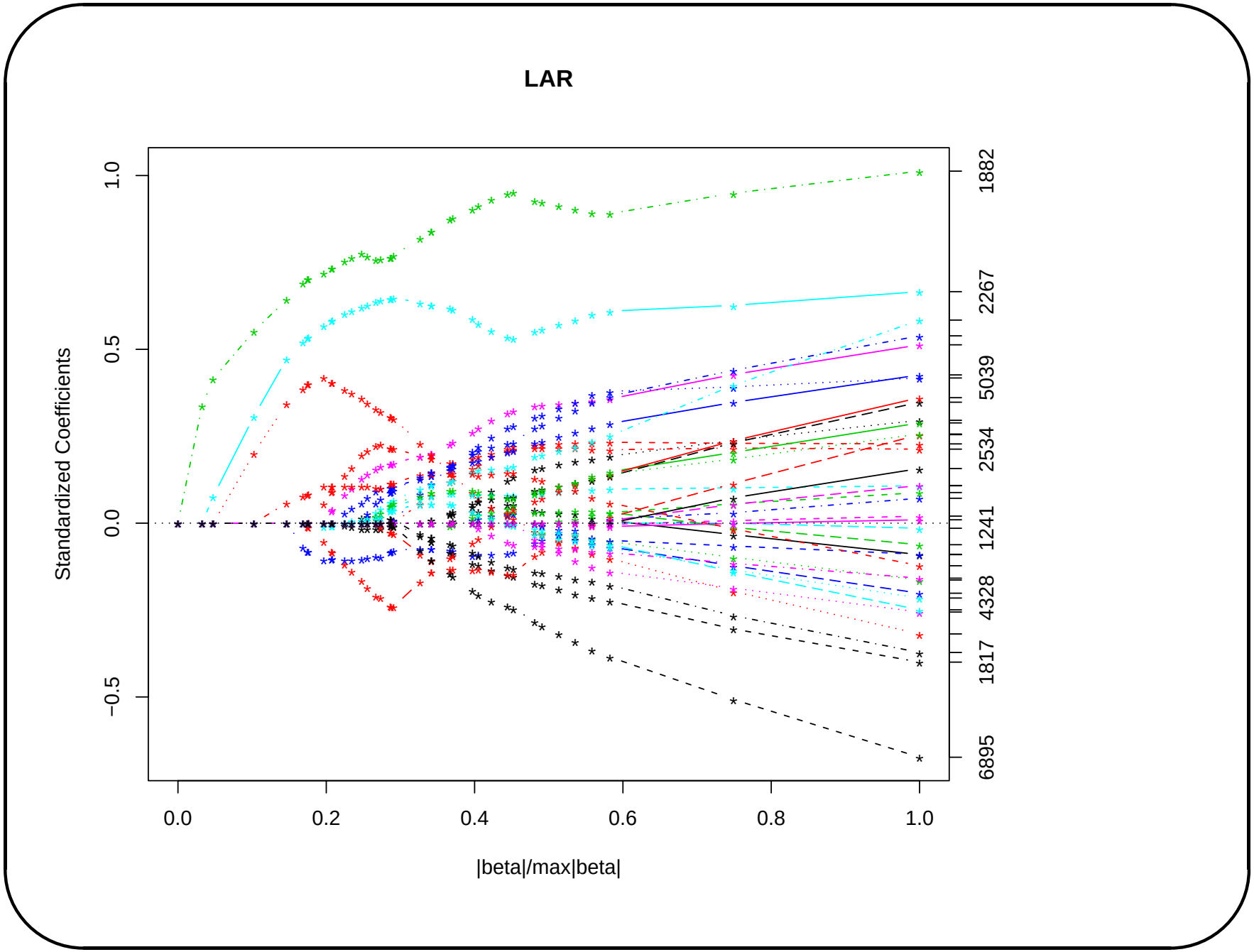
MicroArray Example

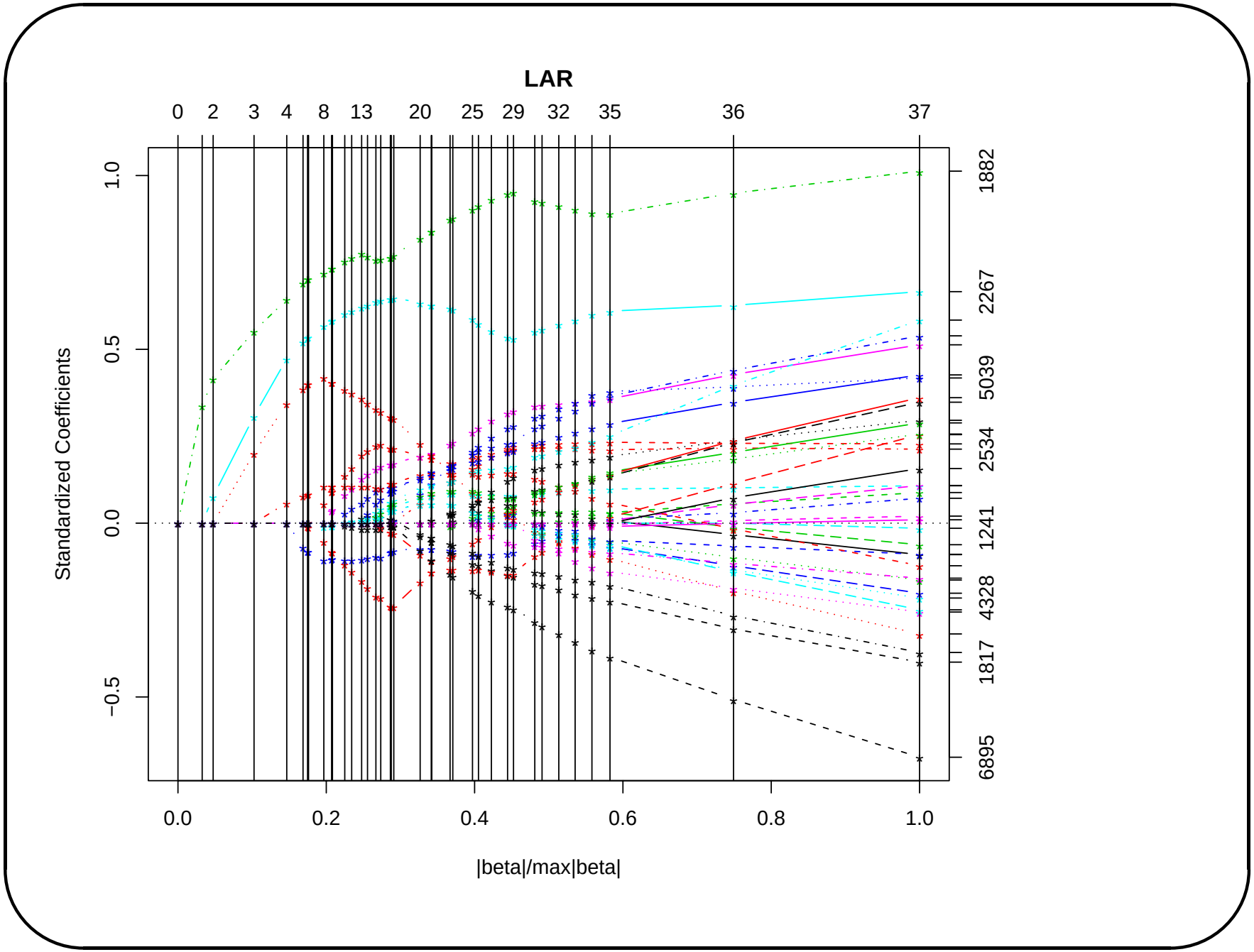
- Expression data for 38 Leukemia patients ("Golub" data).
- X matrix with 38 samples and 7129 variables (genes)
- Response Y is dichotomous ALL (27) vs AML (11)
- LARS (lasso) took 4 seconds in R version 1.7 on a 1.8Ghz Dell workstation running Linux.
- In 70 steps, 52 variables ever non zero, at most 37 at a time.

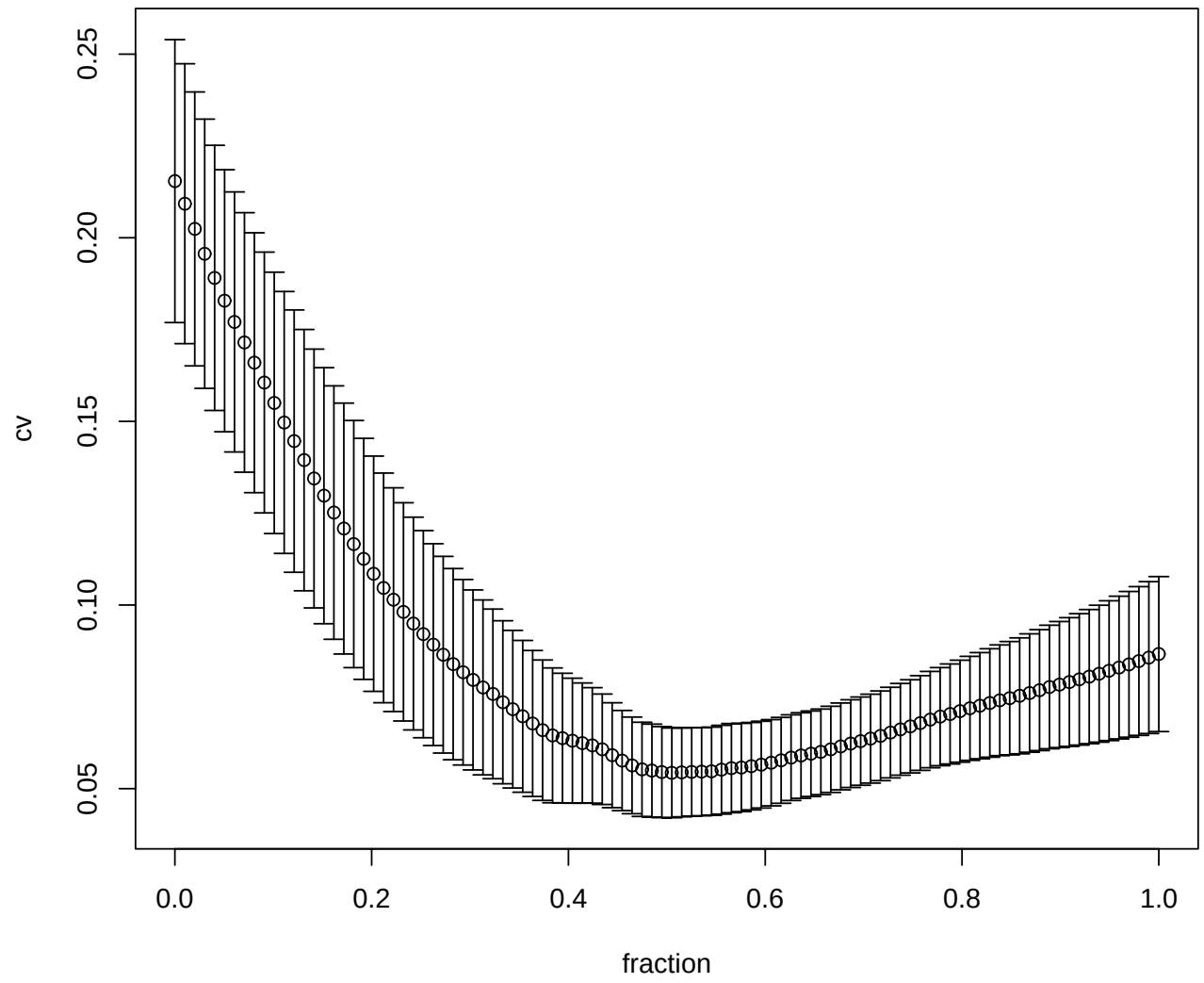

LASSO
Standardized Coefficients
0.8
0.6
0.4
0.2
0.0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
|beta|/max|beta|
2267
461
2945
6801
2968

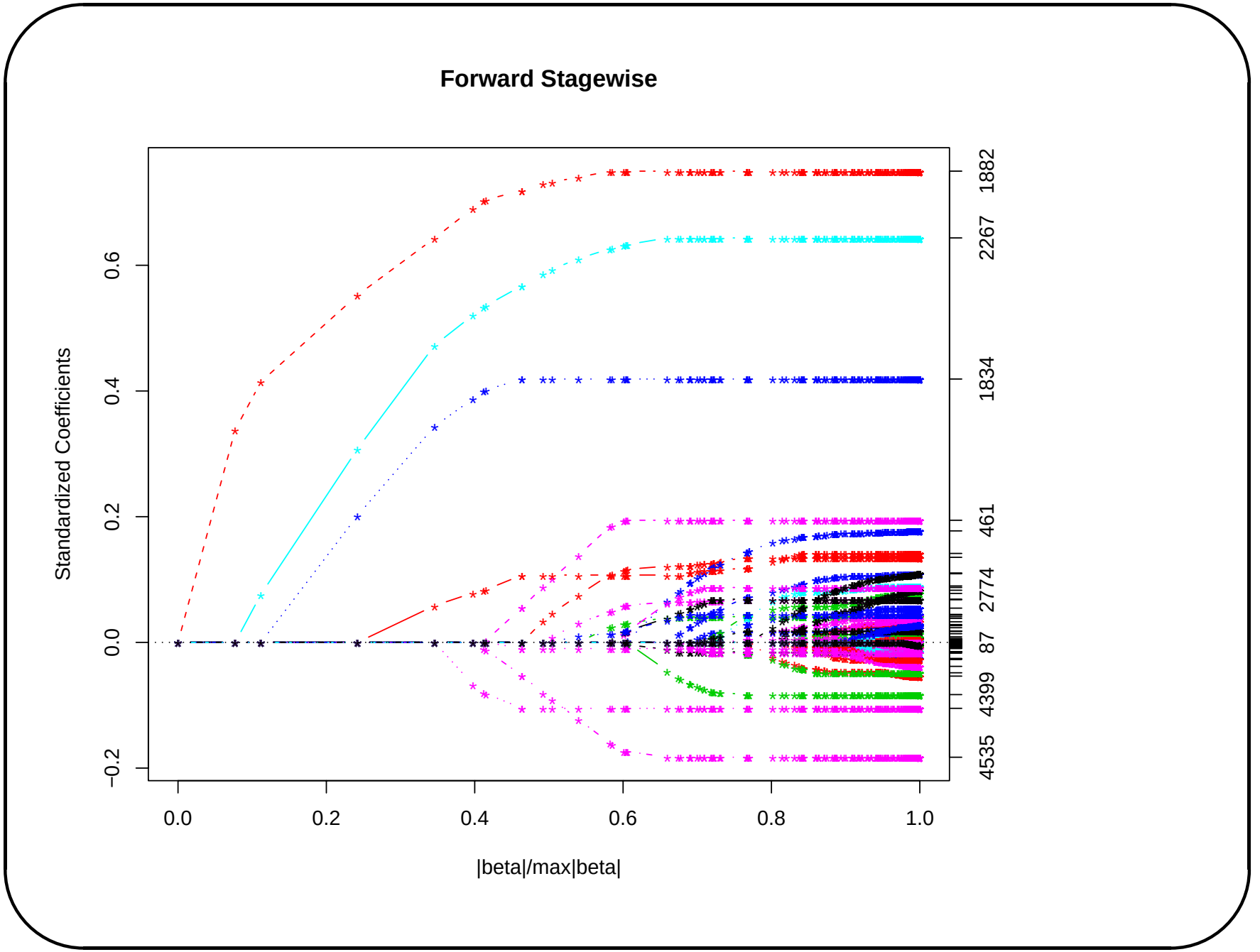

LASSO
0 1 3 4 5 8 12 15 20 26 31 38 50 65
Standardized Coefficients
0.8
0.6
0.4
0.2
0.0
−0.2
2267
461
2945
6801
2968
0.0 0.2 0.4 0.6 0.8 1.0
|beta|/max|beta|

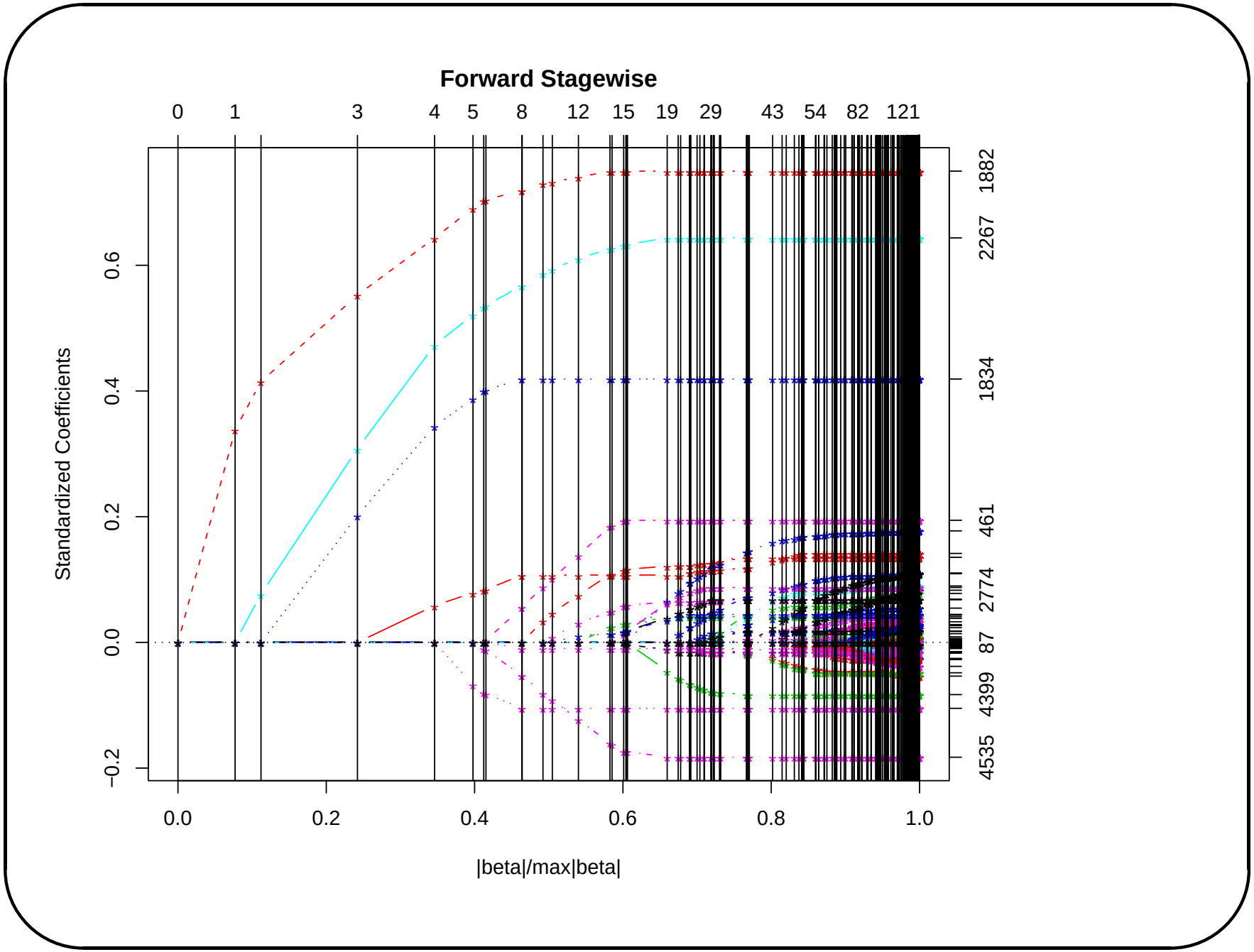

10−fold cross−validation for Leukemia Expression Data (Lasso)
cv
0.05
0.10
0.15
0.20
0.25
0.0
0.2
0.4
0.6
0.8
1.0
fraction

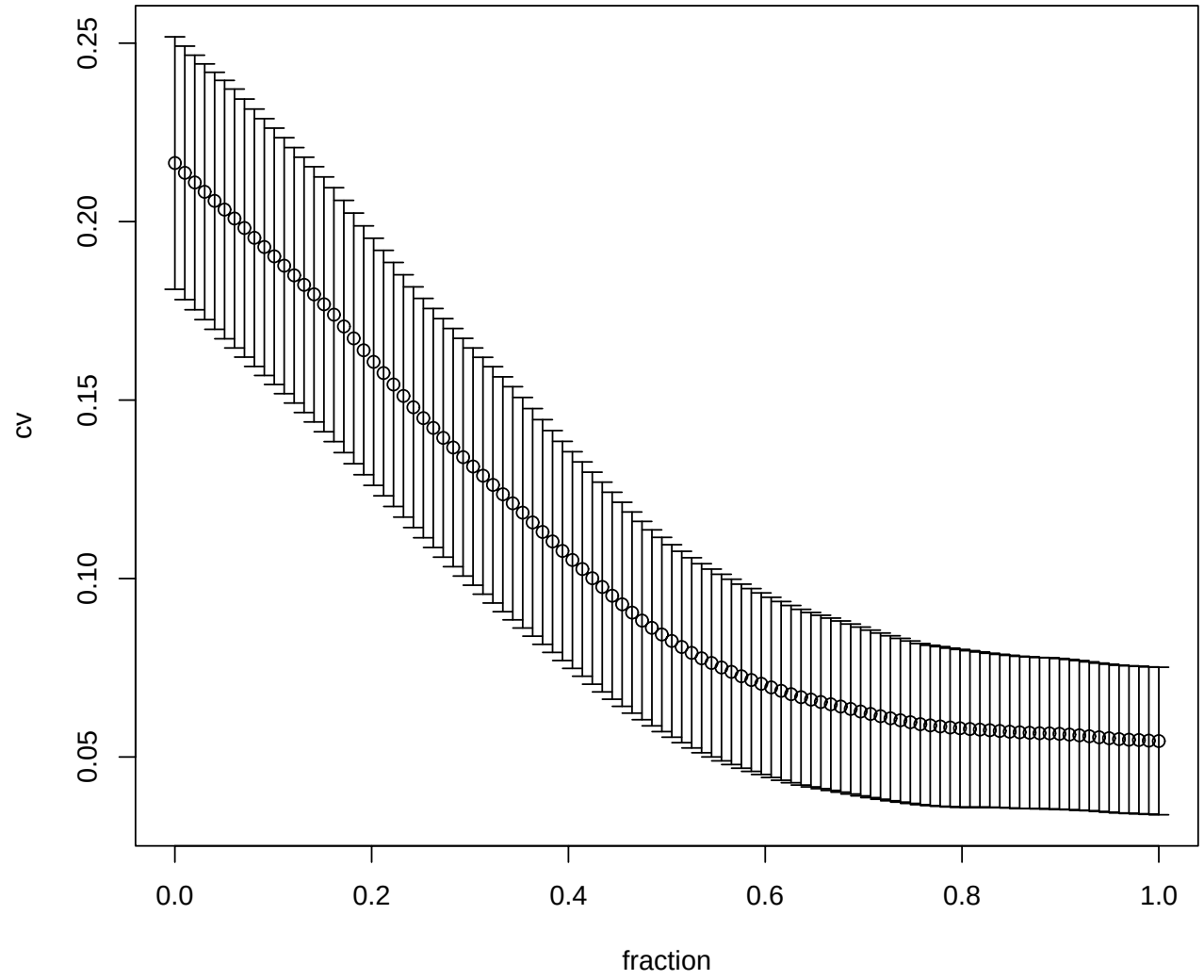

LAR
Standardized Coefficients
1.0
0.5
0.0
−0.5
0.0
0.2
0.4
0.6
0.8
1.0
|beta|/max|beta|
1882
2267
5039
2534
1241
4328
1817
6895


LAR
0 2 3 4 8 13 20 25 29 32 35 36 37
Standardized Coefficients
1.0
0.5
0.0
−0.5
1882
2267
5039
2534
1241
4328
1817
6895
0.0 0.2 0.4 0.6 0.8 1.0
|beta|/max|beta|

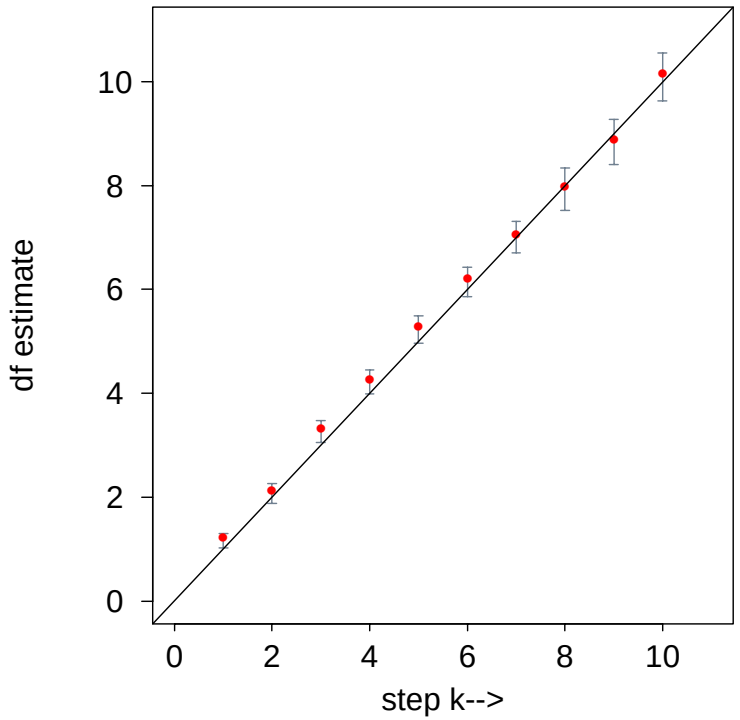

10−fold cross−validation for Leukemia Expression Data (LAR)
cv
0.05
0.10
0.15
0.20
0.25
0.0
0.2
0.4
0.6
0.8
1.0
fraction

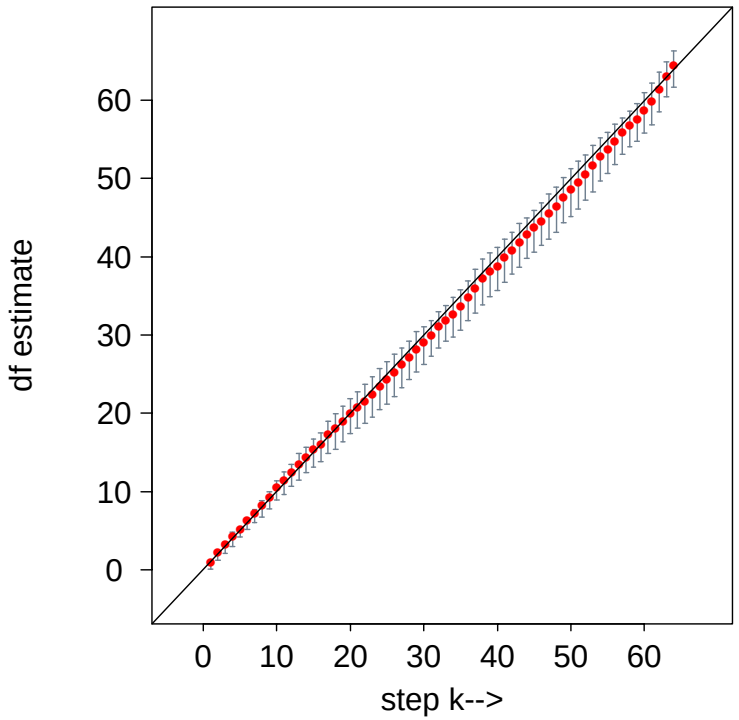

Forward Stagewise
Standardized Coefficients
−0.2
0.0
0.2
0.4
0.6
0.0
0.2
0.4
0.6
0.8
1.0
|beta|/max|beta|
1882
2267
1834
461
2774
87
4399
4535


Forward Stagewise
0 1 3 4 5 8 12 15 19 29 43 54 82 121
Standardized Coefficients
−0.2 0.0 0.2 0.4 0.6
0.0 0.2 0.4 0.6 0.8 1.0
|beta|/max|beta|
1882 2267 1834 461 2774 87 4399 4535


10−fold cross−validation for Leukemia Expression Data (Stagewise)
cv
0.05
0.10
0.15
0.20
0.25
0.0
0.2
0.4
0.6
0.8
1.0
fraction

Degrees of freedom





Degree of Freedom result

Proof is based on is an application of Stein's unbiased risk estimate (SURE). Suppose that $g : \mathbb{R}^n \to \mathbb{R}^n$ is almost differentiable and set $\nabla \cdot g = \sum_{i=1}^n \partial g_i/\partial x_i$. If $\mathbf{y} \sim N_n(\mu, \sigma^2\mathbf{I})$, then Stein's formula states that

$$\sum_{i=1}^n \text{cov}(g_i, y_i)/\sigma^2 = E[\nabla \cdot g(\mathbf{y})].$$

LHS is degrees of freedom. Set $g(\cdot)$ equal to the LAR estimate. In orthogonal case, $\partial g_i/\partial x_i$ is 1 if predictor is in model, 0 otherwise. Hence RHS equals number of predictors in model ($= k$).

Non-orthogonal case is much harder.

Future directions

- Lasso has applications in genetics, e.g. microarray data, mass spectroscopy for measuring proteins. LARS algorithm will allow application to large problems.
- generalization to other models, e.g. logistic regression
- other stepwise algorithms, for other loss functions.
- use ideas to make better versions of boosting (Bogdan Popescu, Ji Zhu, Saharon Rosset)