

***Linear Methods for  
Regression  
- Hastie – Chap – III***

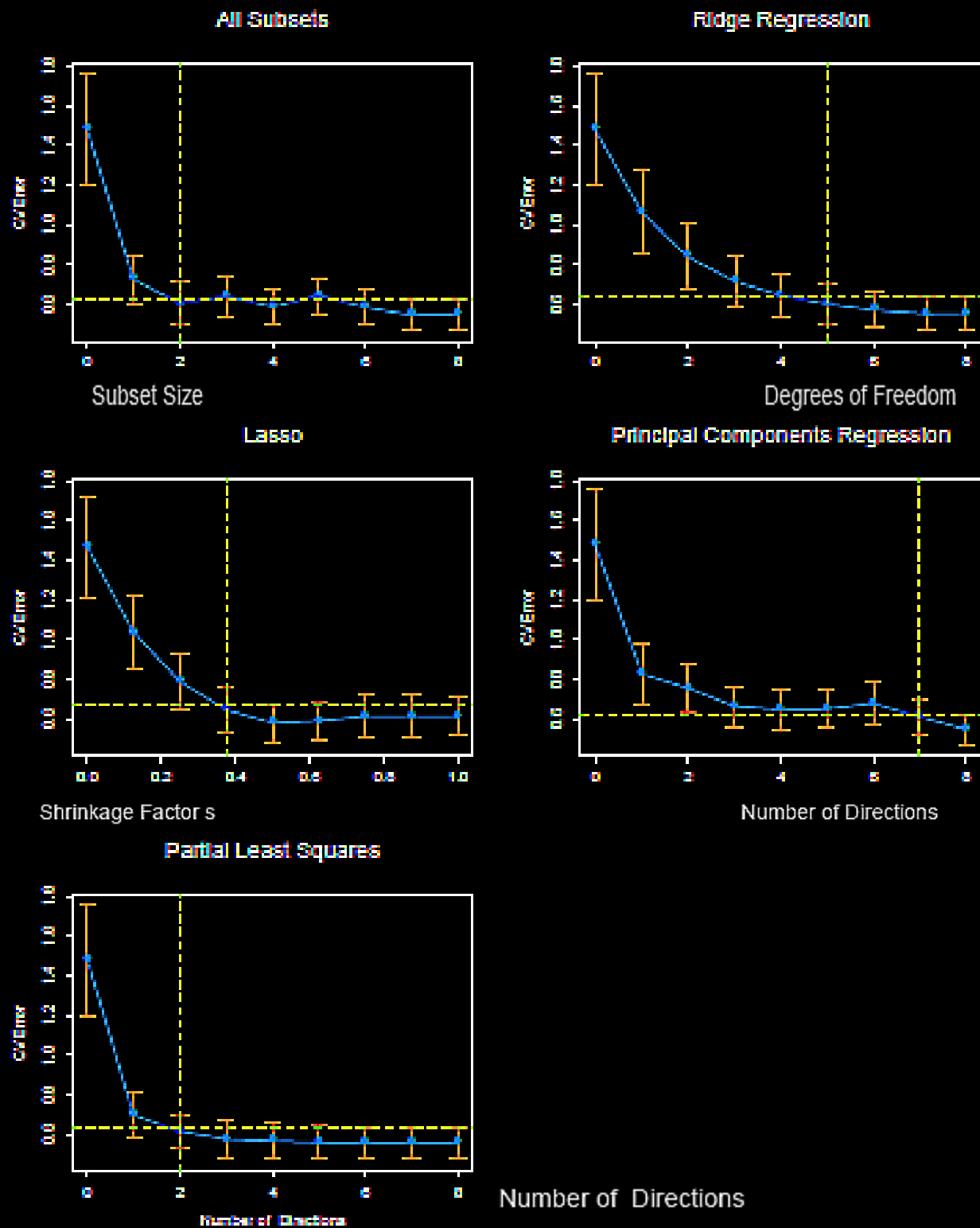
***(part – B)***

# ***Shrinkage Methods***

- By retaining a subset of the predictors and discarding the rest, subset selection produces a model that is interpretable and has possibly lower prediction error than the full model. However, because it is a discrete process—variables are either retained or discarded—it often exhibits high variance, and so doesn't reduce the prediction error of the full model. Shrinkage methods are more continuous, and don't suffer as much from high variability.

## ***Ridge Regression***

- Ridge regression shrinks the regression coefficients by imposing a penalty on their size.



**FIGURE 3.7.** Estimated prediction error curves and their standard errors for the various selection and shrinkage methods. Each curve is plotted as a function of the corresponding complexity parameter for that method. The horizontal axis has been chosen so that the model complexity increases as we move from left to right. The estimates of prediction error and their standard errors were obtained by tenfold cross-validation; full details are given in Section 7.10. The least complex model within one standard error of the best is chosen, indicated by the purple/Yellow vertical broken lines.

- The ridge coefficients minimize a penalized residual sum of squares,

$$\hat{\beta}^{ridge} = \operatorname{argmin}_{\beta} \left\{ \sum_{i=1}^N (y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j)^2 + \lambda \sum_{j=1}^p \beta_j^2 \right\}. \quad (3.41)$$

Here,  $\lambda \geq 0$  is a complexity parameter that controls the amount of shrinkage: the larger the value of  $\lambda$ , the greater the amount of shrinkage. The coefficients are shrunk toward zero (and each other).

- An equivalent way to write the ridge problem is

$$\begin{aligned} \hat{\beta}^{ridge} = \operatorname{argmin}_{\beta} \sum_{i=1}^N (y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j)^2, \\ \text{subject to } \sum_{j=1}^p \beta_j^2 \leq t, \end{aligned} \quad (3.42)$$

- Which makes explicit the size constraint on the parameters. There is a one to-one correspondence between the parameters  $\lambda$  in (3.41) and  $t$  in (3.42). When there are many correlated variables in a linear regression model, their coefficients can become poorly determined and exhibit high variance. *A wildly large positive coefficient on one variable can be canceled by a similarly large negative coefficient on its correlated cousin.* By imposing a size constraint on the coefficients, as in (3.42), this problem is alleviated.
- The ridge solutions are not equivariant under scaling of the inputs, and so one normally standardizes the inputs before solving (3.41). The solution to (3.41) can be separated into two parts, after reparametrization using centered inputs: each  $x_{ij}$  gets replaced by  $x_{ij} - \bar{x}_j$ .
- We estimate  $\beta_0$  by  $\bar{y} = \frac{1}{N} \sum_1^N y_i$ .

- The remaining coefficients get estimated by a ridge regression without intercept, using the centered  $x_{ij}$ . Henceforth we assume that this centering has been done, so that the input matrix  $\mathbf{X}$  has  $p$  (rather than  $p + 1$ ) columns.
- Writing the criterion in (3.41) in matrix form,

$$RSS(\lambda) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \boldsymbol{\beta}^T \boldsymbol{\beta}, \quad (3.43)$$

- The ridge regression solutions are easily seen to be

$$\hat{\boldsymbol{\beta}}^{ridge} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}, \quad (3.44)$$

where,  $\mathbf{I}$  is the  $p \times p$  identity matrix. Notice that with the choice of quadratic penalty  $\boldsymbol{\beta}^T \boldsymbol{\beta}$ , the ridge regression solution is again a linear function of  $y$ . The solution adds a positive constant to the diagonal of  $\mathbf{X}^T \mathbf{X}$  before inversion.

- This makes the problem nonsingular, even if  $\mathbf{X}^T \mathbf{X}$  is not of full rank, and was the main motivation for ridge regression when it was first introduced in statistics (Hoerl and Kennard, 1970).
- Traditional descriptions of ridge regression start with definition (3.44). We choose to motivate it via (3.41) and (3.42), as these provide insight into how it works.

- Ridge regression can also be derived as the mean or mode of a posterior distribution, with a suitably chosen prior distribution.
- In detail, suppose  $y_i \sim N(\beta_0 + x_i^T \beta, \sigma^2)$ , and the parameters  $\beta_j$  are each distributed as  $N(0, \tau^2)$ , independently of one another. Then the (negative) log-posterior density of  $\beta$ , with  $\tau^2$  and  $\sigma^2$  assumed known, is equal to the expression in curly braces in (3.41), with  $\lambda = \sigma^2/\tau^2$ . Thus the ridge estimate is the mode of the posterior distribution; since the distribution is Gaussian, it is also the posterior mean.
- The *singular value decomposition (SVD)* of the centered input matrix  $\mathbf{X}$  gives us some additional insight into the nature of ridge regression. The *SVD* of the  $N \times p$  matrix  $\mathbf{X}$  has the form

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T \quad (3.44)$$



- Using the singular value decomposition we can write the least squares fitted vector as

$$\begin{aligned} \mathbf{X}\hat{\beta}^{ls} &= \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}, \\ &= \mathbf{U}\mathbf{U}^T \mathbf{y}, \end{aligned} \tag{3.46}$$

- After some simplification. Note that  $\mathbf{U}^T \mathbf{y}$  are the coordinates of  $\mathbf{y}$  with respect to the orthonormal basis  $\mathbf{U}$ . Note also the similarity with (3.33);  $\mathbf{Q}$  and  $\mathbf{U}$  are generally different orthogonal bases for the column space of  $\mathbf{X}$  (*Exercise 3.8*).

Now the ridge solutions are

$$\hat{\mathbf{y}} = \mathbf{Q}\mathbf{Q}^T \mathbf{y}. \tag{3.33}$$

$$\begin{aligned} \mathbf{X}\hat{\beta}^{ridge} &= \mathbf{X}(\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y} \\ &= \mathbf{U} \mathbf{D}(\mathbf{D}^2 + \lambda \mathbf{I})^{-1} \mathbf{D} \mathbf{U}^T \mathbf{y} \\ &= \sum_{j=1}^p \mathbf{u}_j \frac{d_j^2}{d_j^2 + \lambda} \mathbf{u}_j^T \mathbf{y}, \end{aligned} \tag{3.47}$$

- Where the  $\mathbf{u}_j$  are the columns of  $\mathbf{U}$ . Note that since  $\lambda \geq 0$ , we have  $d_j^2 / (d_j^2 + \lambda) \leq 1$ . Like linear regression, ridge regression computes the coordinates of  $\mathbf{y}$  with respect to the orthonormal basis  $\mathbf{U}$ . It then shrinks these coordinates by the factors  $d_j^2 / (d_j^2 + \lambda)$ . This means that a greater amount of shrinkage is applied to the coordinates of basis vectors with smaller  $d_j^2$ .
- What does a small value of  $d_j^2$  mean? The *SVD* of the centered  $\mathbf{X}$  is another way of expressing the *principal components* of the variables in  $\mathbf{X}$ . The sample covariance matrix is given by  $\mathbf{S} = \mathbf{X}^T \mathbf{X} / N$ , and from (3.45) we have

$$\mathbf{X}^T \mathbf{X} = \mathbf{V} \mathbf{D}^2 \mathbf{V}^T, \quad (3.48)$$

- Which is the *eigen decomposition* of  $\mathbf{X}^T \mathbf{X}$  (and of  $\mathbf{S}$ , up to a factor  $N$ ). The eigenvectors  $v_j$  (columns of  $\mathbf{V}$ ) are also called the *principal components* (or Karhunen–Loeve) directions of  $\mathbf{X}$ . Sample variance is easily seen to be and in fact

$$\mathbf{z}_1 = \mathbf{X}v_1 = \mathbf{u}_1 d_1$$

$$\text{Var}(\mathbf{z}_1) = \text{Var}(\mathbf{X}v_1) = \frac{d_1^2}{N}, \quad (3.49)$$

- Subsequent principal components  $\mathbf{z}_j$  have maximum variance  $d_j^2/N$ , subject to being orthogonal to the earlier ones. Conversely the last principal component has minimum variance. Hence the small singular values  $d_j$  correspond to directions in the column space of  $\mathbf{X}$  having small variance, and ridge regression shrinks these directions the most.

- In Figure 3.7 we have plotted the estimated prediction error versus the quantity

$$\begin{aligned}df(\lambda) &= \text{tr}[\mathbf{X}(\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T], \\ &= \text{tr}(\mathbf{H}_\lambda) \\ &= \sum_{j=1}^p \frac{d_j^2}{d_j^2 + \lambda} .\end{aligned}\tag{3.50}$$

- This monotone decreasing function of  $\lambda$  is the *effective degrees of freedom* of the ridge regression fit. Usually in a linear-regression fit with  $p$  variables, the degrees-of-freedom of the fit is  $p$ , the number of free parameters. The idea is that although all  $p$  coefficients in a ridge fit will be non-zero, they are fit in a restricted fashion controlled by  $\lambda$ . Note that  $df(\lambda) = p$  when  $\lambda = 0$  (no regularization) and  $df(\lambda) \rightarrow 0$  as  $\lambda \rightarrow \infty$ .

# The Lasso

- The lasso is a shrinkage method like ridge, with subtle but important differences. The lasso estimate is defined by

$$\hat{\beta}^{lasso} = \operatorname{argmin}_{\beta} \sum_{i=1}^N (y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j)^2$$

*subject to*  $\sum_{j=1}^p |\beta_j| \leq t$  (3.51)

- Write the lasso problem in the equivalent Lagrangian form

$$\hat{\beta}^{lasso} = \operatorname{argmin}_{\beta} \left\{ \frac{1}{2} \sum_{i=1}^N (y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j)^2 + \lambda \sum_{j=1}^p |\beta_j| \right\}.$$

(3.52)

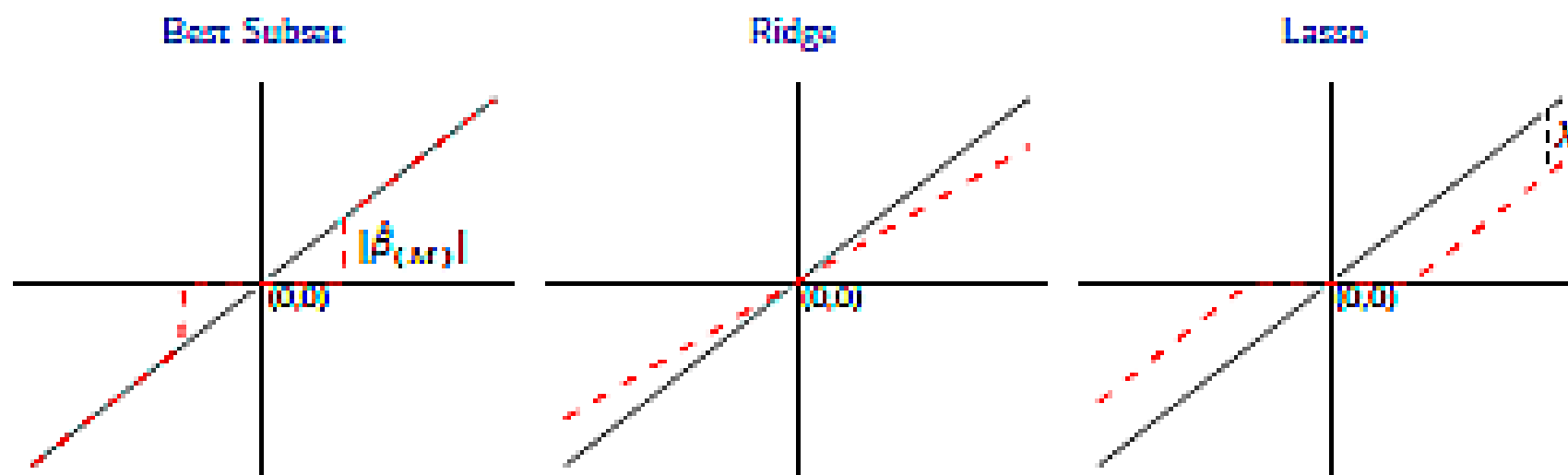
- Notice the similarity to the ridge regression problem (3.42) or (3.41): the  $L_2$  ridge penalty  $\sum_1^p \beta_j^2$  is replaced by the  $L_1$  lasso penalty  $\sum_1^p |\beta_j|$ . Thus the lasso does a kind of continuous subset selection. If  $t$  is chosen larger than  $t_0 = \sum_1^p |\hat{\beta}_j|$  (where  $\hat{\beta}_j = \hat{\beta}_j^{ls}$ , the least squares estimates), then the lasso estimates are the  $\hat{\beta}_j$ 's. On the other hand, for  $t = t_0/2$  say, then the least squares coefficients are shrunk by about 50% on average.

# ***Discussion: Subset Selection, Ridge Regression and the Lasso***

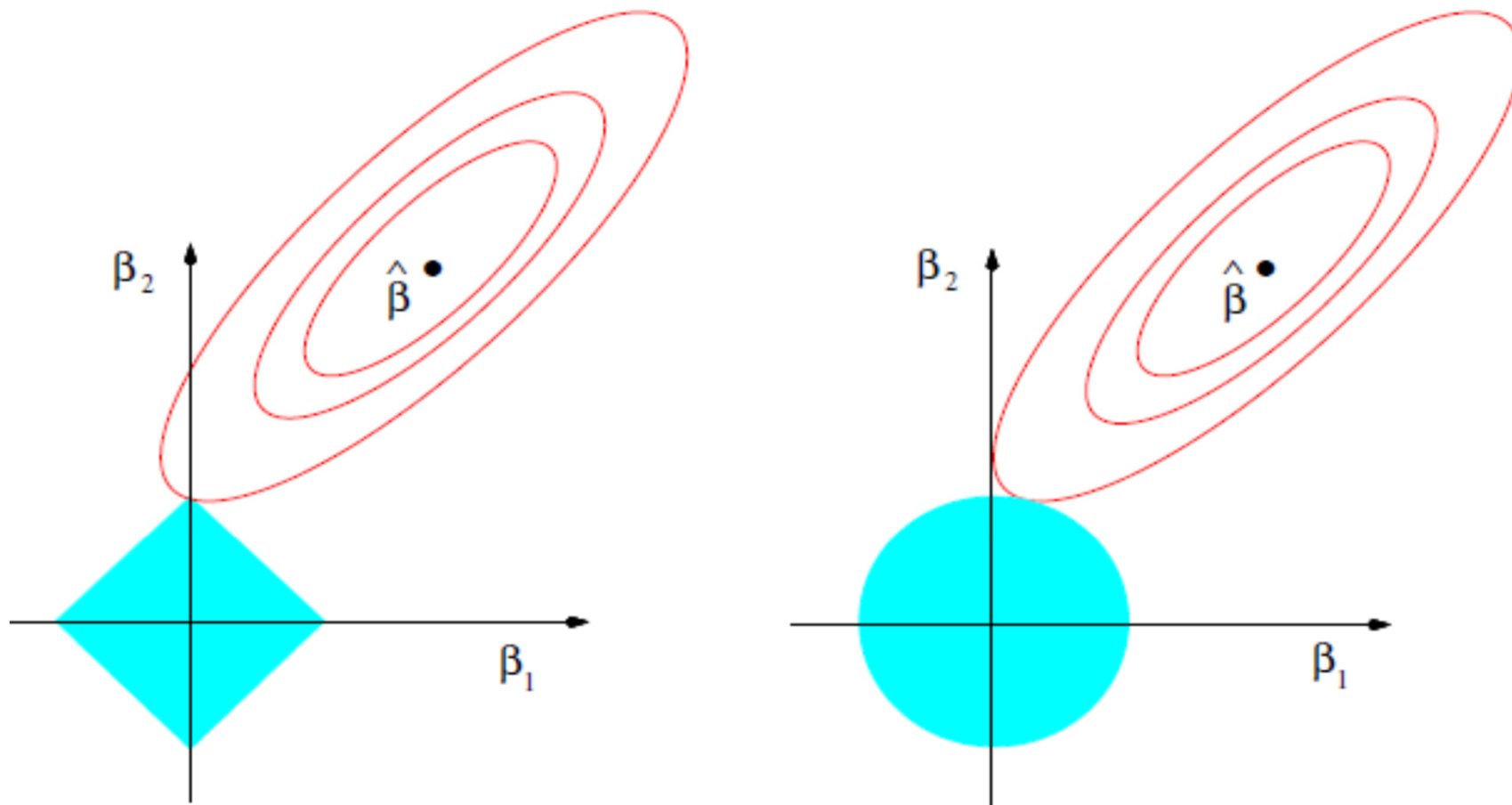
- In the case of an orthonormal input matrix  $\mathbf{X}$  the three procedures have explicit solutions. Each method applies a simple transformation to the least squares estimate  $\hat{\beta}_j$ , as detailed in Table 3.4.
- Ridge regression does a proportional shrinkage. Lasso translates each coefficient by a constant factor  $\lambda$ , truncating at zero. This is called “soft thresholding,”. Best-subset selection drops all variables with coefficients smaller than the  $M^{\text{th}}$  largest; this is a form of “hard-thresholding.”
- Back to the no orthogonal case; some pictures help understand their relationship. Figure 3.11 depicts the lasso (*left*) and ridge regression (*right*) when there are only two parameters. The residual sum of squares has elliptical contours, centered at the full least squares estimate.

TABLE 3.4. Estimators of  $\beta_j$  in the case of orthonormal columns of  $X$ .  $M$  and  $\lambda$  are constants chosen by the corresponding techniques;  $\text{sign}$  denotes the sign of its argument ( $\pm 1$ ), and  $x_+$  denotes “positive part” of  $x$ . Below the table, estimators are shown by broken red lines. The  $45^\circ$  line in gray shows the unrestricted estimate for reference.

Estimator	Formula
Best subset (size $M$ )	$\hat{\beta}_j \cdot I( \hat{\beta}_j  \geq  \hat{\beta}_{(M)} )$
Ridge	$\hat{\beta}_j / (1 + \lambda)$
Lasso	$\text{sign}(\hat{\beta}_j) ( \hat{\beta}_j  - \lambda)_+$







**FIGURE 3.11.** Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions  $|\beta_1| + |\beta_2| \leq t$  and  $\beta_1^2 + \beta_2^2 \leq t^2$ , respectively, while the red ellipses are the contours of the least squares error function.

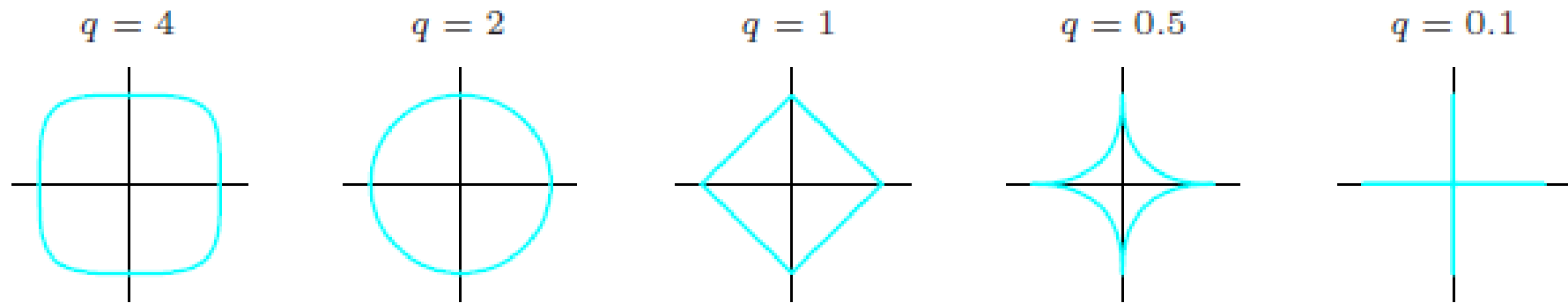
- Region for ridge regression is the disk  $\beta_1^2 + \beta_2^2 \leq t$ , while that for lasso is the diamond  $|\beta_1| + |\beta_2| \leq t$ . Both methods find the first point where the elliptical contours hit the constraint region. Unlike the disk, the diamond has corners; if the solution occurs at a corner, then it has one parameter  $\beta_j$  equal to zero. When  $p > 2$ , the diamond becomes a rhomboid, and has many corners, flat edges and faces; there are many more opportunities for the estimated parameters to be zero.
- Consider the criterion

$$\tilde{\beta} = \operatorname{argmin}_{\beta} \left\{ \sum_{i=1}^N (y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j)^2 + \lambda \sum_{j=1}^p |\beta_j|^q \right\}. \quad (3.53)$$

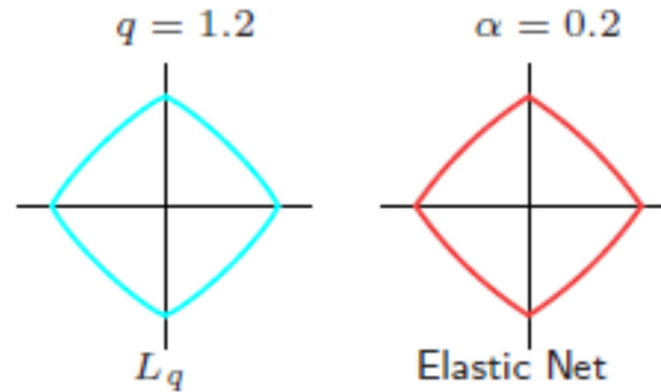
for  $q \geq 0$ .

The contours of constant value of  $\sum_j |\beta_j|^q$  are shown in Figure 3.12, for the case of two inputs.

- Thinking of  $|\beta_j|^q$  as the log-prior density. The case  $q = 1$  (*lasso*) is the smallest  $q$  such that the constraint region is convex; non-convex constraint regions make the optimization problem more difficult. In this view, the lasso, ridge regression and best subset selection are Bayes estimates with different priors. They are derived as posterior modes, that is, maximizers of the posterior.



**FIGURE 3.12.** Contours of constant value of  $\sum_j |\beta_j|^q$  for given values of  $q$ .



- FIGURE 3.13.** Contours of constant value of  $\sum_j |\beta_j|^q$  for  $q = 1.2$  (left plot), and the elastic-net penalty  $\sum_j (\alpha \beta_j^2 + (1 - \alpha)|\beta_j|)$  for  $\alpha = 0.2$  (right plot). Although visually very similar, the elastic
- Zou and Hastie (2005) introduced the elastic net penalty

$$\lambda \sum_{j=1}^p (\alpha \beta_j^2 + (1 - \alpha)|\beta_j|), \quad (3.54)$$

# ***Least Angle Regression***

- Least angle regression (*LAR*) is a relative newcomer (*Efron et al., 2004*), and can be viewed as a kind of “*democratic*” version of forward stepwise regression
- Forward stepwise regression builds a model sequentially, adding one variable at a time. At each step, it identifies the best variable to include in the active set, and then updates the least squares fit to include all the active variables.
- Least angle regression uses a similar strategy, but only enters “as much” of a predictor as it deserves. At the first step it identifies the variable most correlated with the response. Rather than fit this variable completely, *LAR* moves the coefficient of this variable continuously toward its leastsquares value (causing its correlation with the evolving residual to decrease in absolute value).

- As soon as another variable “catches up” in terms of correlation with the residual, the process is paused. The second variable then joins the active set, and their coefficients are moved together in a way that keeps their correlations tied and decreasing. This process is continued until all the variables are in the model, and ends at the full least-squares fit. Algorithm 3.2 provides the details.

---

## Algorithm 3.2 Least Angle Regression.

---

1. Standardize the predictors to have mean zero and unit norm. Start with the residual  $\mathbf{r} = \mathbf{y} - \bar{y}, \beta_1, \beta_2, \dots, \beta_p = 0$ .
2. Find the predictor  $\mathbf{x}_j$  most correlated with  $\mathbf{r}$ .
3. Move  $\beta_j$  from 0 towards its least-squares coefficient  $\langle \mathbf{x}_j, \mathbf{r} \rangle$ , until some other competitor  $\mathbf{x}_k$  has as much correlation with the current residual as does  $\mathbf{x}_j$ .
4. Move  $\beta_j$  and  $\beta_k$  in the direction defined by their joint least squares coefficient of the current residual on  $(\mathbf{x}_j, \mathbf{x}_k)$ , until some other competitor  $\mathbf{x}_l$  has as much correlation with the current residual.
5. Continue in this way until all  $p$  predictors have been entered. After  $\min(N - 1, p)$  steps, we arrive at the full least-squares solution.

- Suppose  $A_k$  is the active set of variables at the beginning of the  $k^{\text{th}}$  step, and let  $\beta_{A_k}$  be the coefficient vector for these variables at this step; there will be  $k - 1$  nonzero values, and the one just entered will be zero. If  $\mathbf{r}_k = \mathbf{y} - \mathbf{X}_{A_k}\beta_{A_k}$  is the current residual, then the direction for this step is

$$\delta_k = (\mathbf{X}_{A_k}^T \mathbf{X})^{-1} \mathbf{X}_{A_k}^T \mathbf{r}_k. \quad (3.55)$$

- The coefficient profile then evolves as  $\beta_{A_k}(\alpha) = \beta_{A_k} + \alpha \cdot \delta_k$ .
- By construction the coefficients in *LAR* change in a piecewise linear fashion.
- Read Eqns 3.56-3.59 for similarity between LAR vs Lasso (self study).



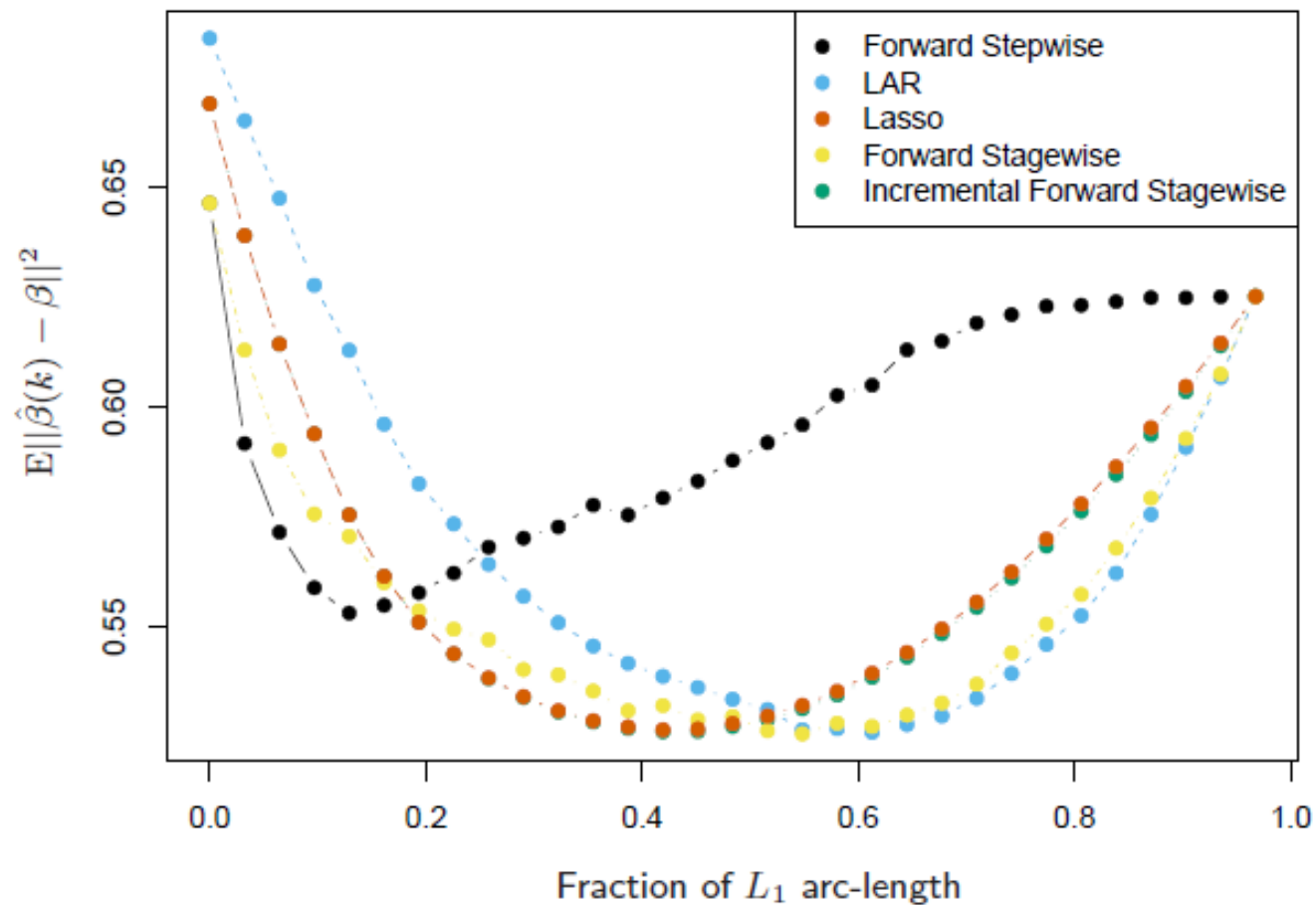
---

**Algorithm 3.2a** Least Angle Regression: Lasso Modification.

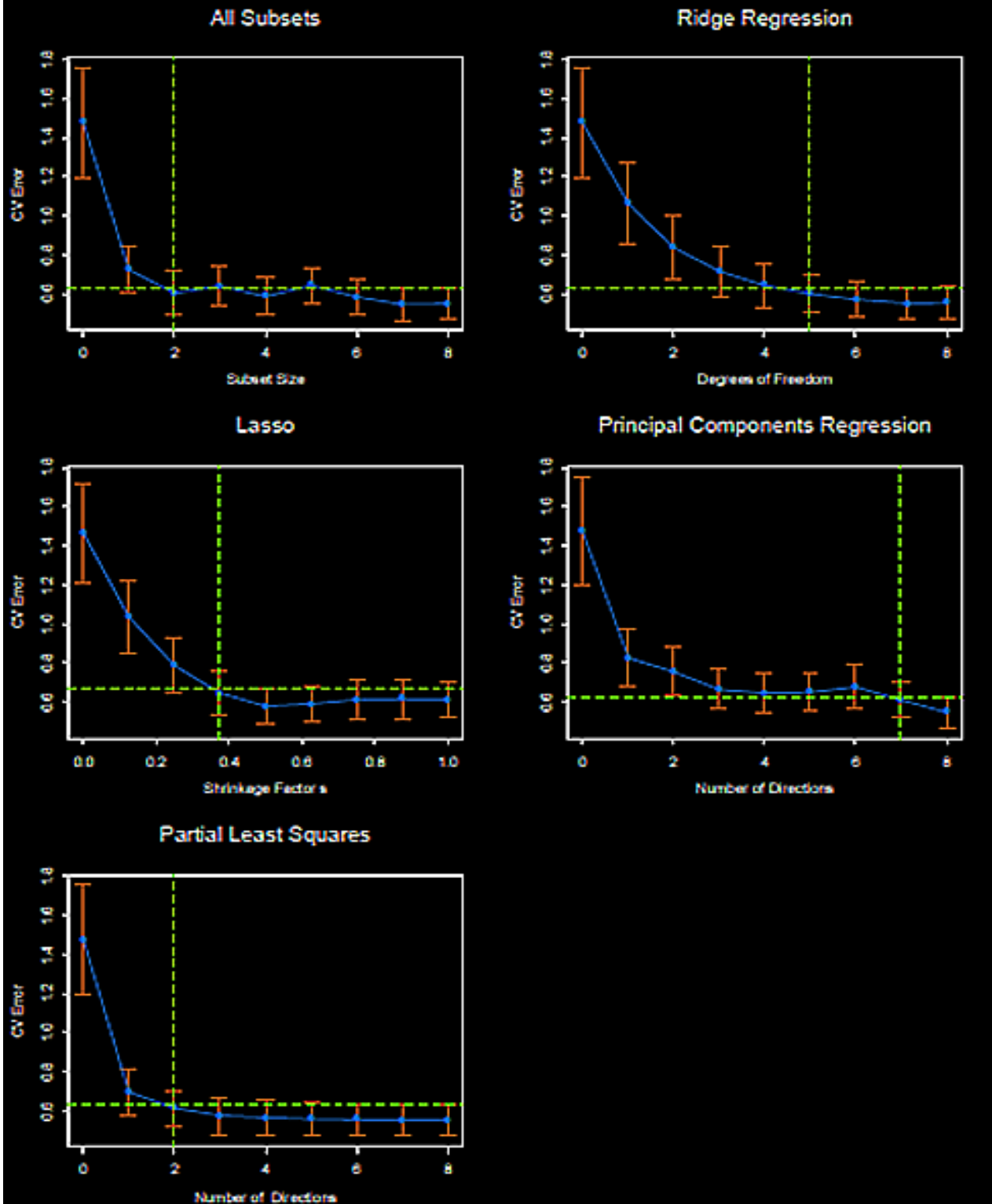
---

4a. If a non-zero coefficient hits zero, drop its variable from the active set of variables and recompute the current joint least squares direction.

- The *LAR*(lasso) algorithm is extremely efficient, requiring the same order of computation as that of a single least squares fit using the  $p$  predictors. Least angle regression always takes  $p$  steps to get to the full least squares estimates. The lasso path can have more than  $p$  steps, although the two are often quite similar. Algorithm 3.2 with the lasso modification 3.2a is an efficient way of computing the solution to any lasso problem, especially when  $p \gg N$



**FIGURE 3.16.** Comparison of LAR and lasso with forward stepwise, forward stagewise (FS) and incremental forward stagewise ( $FS_0$ ) regression. The setup is the same as in Figure 3.6, except  $N = 100$  here rather than 300. Here the slower FS regression ultimately outperforms forward stepwise. LAR and lasso show similar behavior to FS and  $FS_0$ . Since the procedures take different numbers of steps (across simulation replicates and methods), we plot the MSE as a function of the fraction of total  $L_1$  arc-length toward the least-squares fit.



**FIGURE 3.7.** Estimated prediction error curves and their standard errors for the various selection and shrinkage methods. Each curve is plotted as a function of the corresponding complexity parameter for that method. The horizontal axis has been chosen so that the model complexity increases as we move from left to right. The estimates of prediction error and their standard errors were obtained by tenfold cross-validation; full details are given in Section 7.10. The least complex model within one standard error of the best is chosen, indicated by the green vertical broken lines.

## Degrees-of-Freedom Formula for LAR and Lasso

Suppose that we fit a linear model via the least angle regression procedure, stopping at some number of steps  $k < p$ , or equivalently using a lasso bound  $t$  that produces a constrained version of the full least squares fit. How many parameters, or “degrees of freedom” have we used?

Consider first a linear regression using a subset of  $k$  features. If this subset is prespecified in advance without reference to the training data, then the degrees of freedom used in the fitted model is defined to be  $k$ . Indeed, in classical statistics, the number of linearly independent parameters is what is meant by “degrees of freedom.” Alternatively, suppose that we carry out a best subset selection to determine the “optimal” set of  $k$  predictors. Then the resulting model has  $k$  parameters, but in some sense we have used up more than  $k$  degrees of freedom.

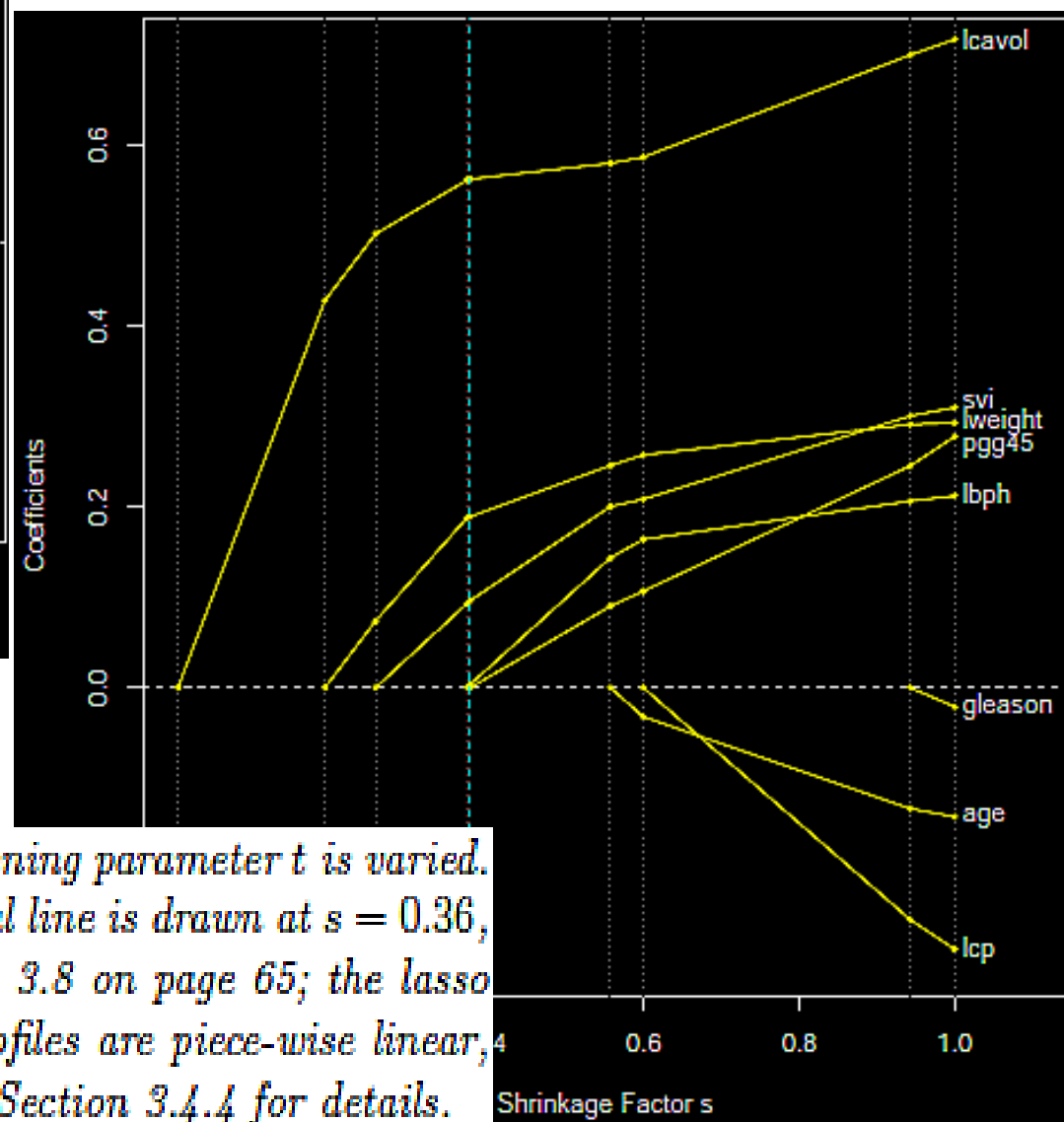
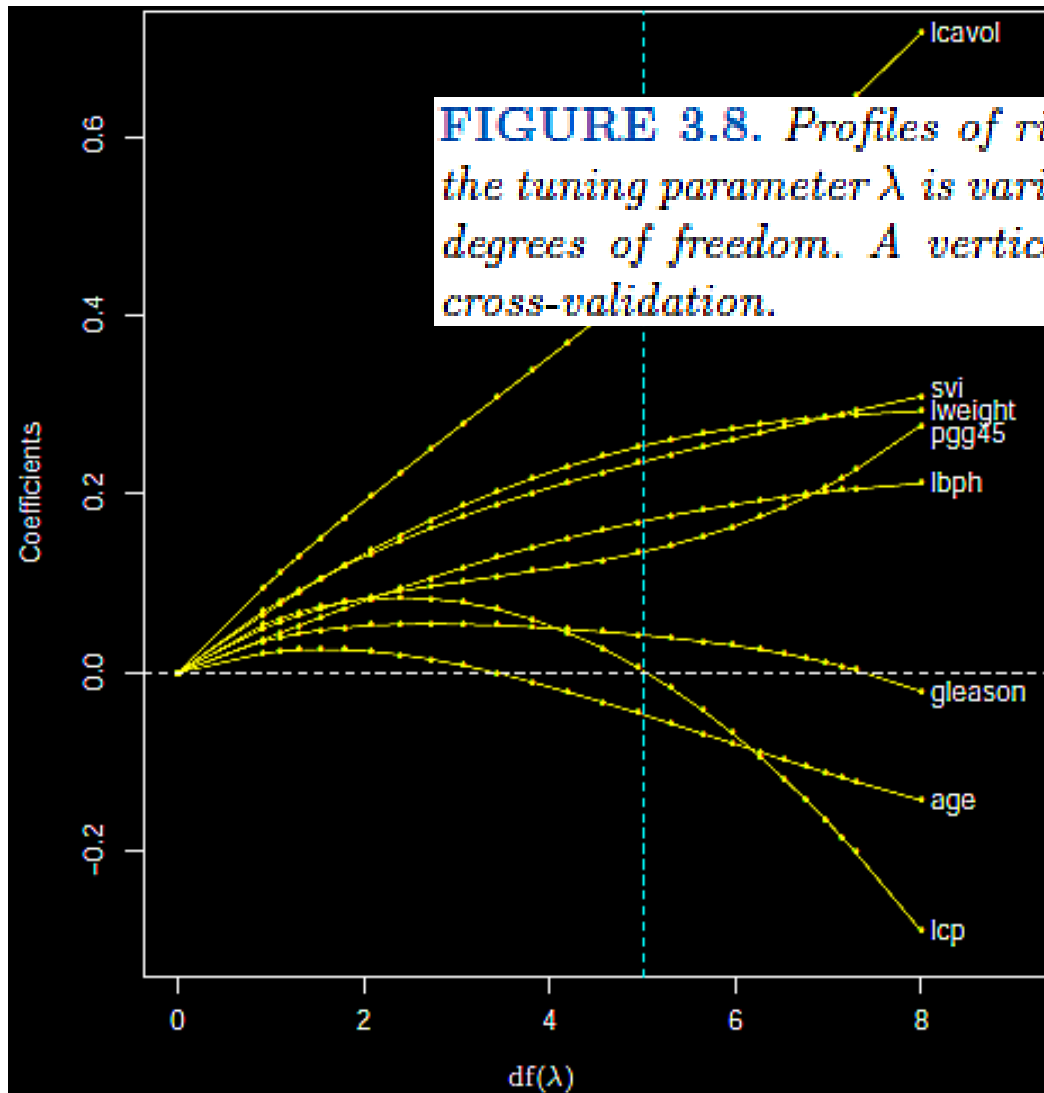
We need a more general definition for the effective degrees of freedom of an adaptively fitted model. We define the degrees of freedom of the fitted vector  $\hat{\mathbf{y}} = (\hat{y}_1, \hat{y}_2, \dots, \hat{y}_N)$  as

$$\text{df}(\hat{\mathbf{y}}) = \frac{1}{\sigma^2} \sum_{i=1}^N \text{Cov}(\hat{y}_i, y_i). \quad (3.60)$$

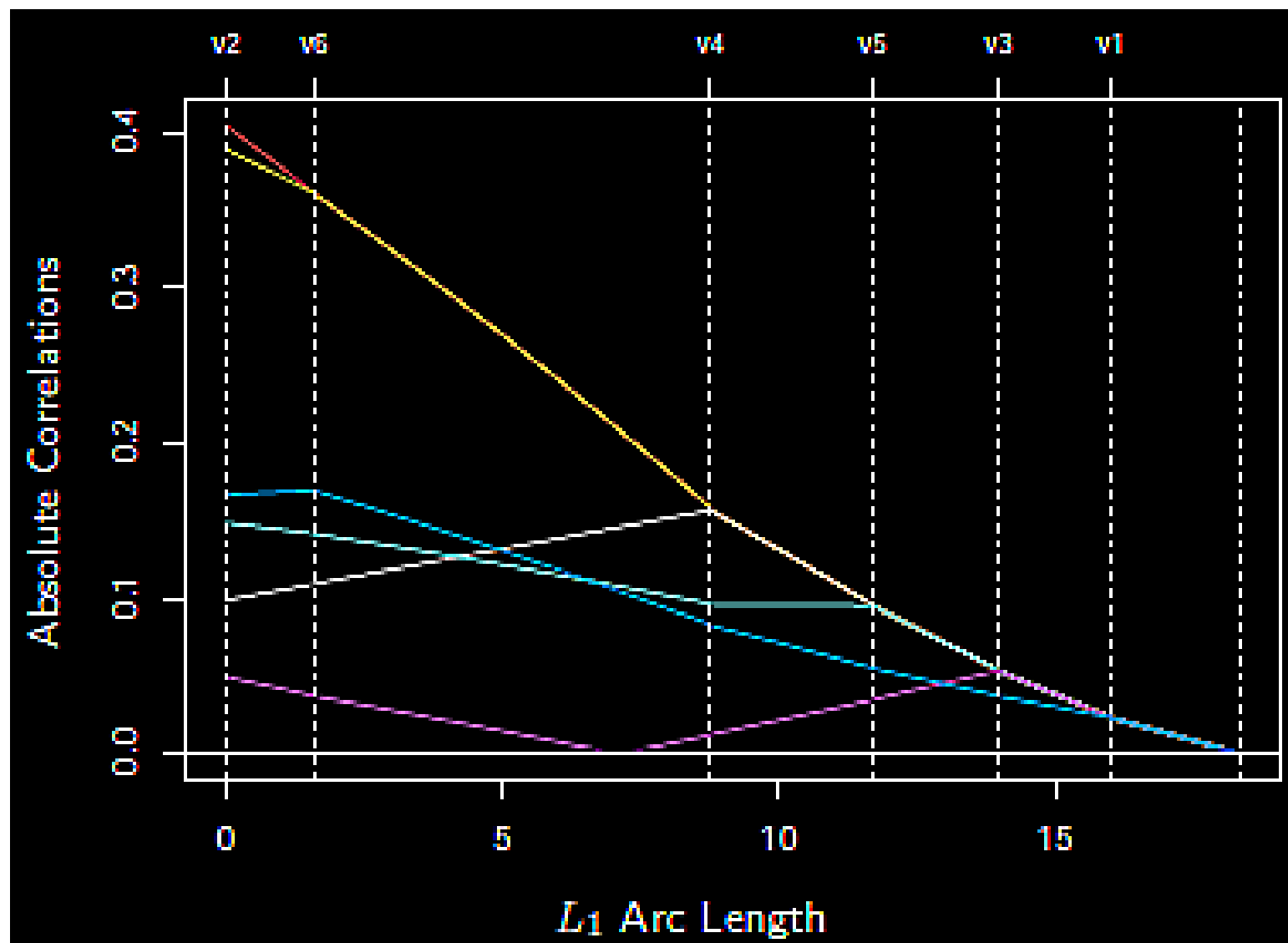
Here  $\text{Cov}(\hat{y}_i, y_i)$  refers to the sampling covariance between the predicted value  $\hat{y}_i$  and its corresponding outcome value  $y_i$ . This makes intuitive sense: the harder that we fit to the data, the larger this covariance and hence  $\text{df}(\hat{\mathbf{y}})$ . Expression (3.60) is a useful notion of degrees of freedom, one that

Now for a linear regression with  $k$  fixed predictors, it is easy to show that  $\text{df}(\hat{y}) = k$ . Likewise for ridge regression, this definition leads to the closed-form expression (3.50) on page 68:  $\text{df}(\hat{y}) = \text{tr}(\mathbf{S}_\lambda)$ . In both these cases, (3.60) is simple to evaluate because the fit  $\hat{y} = \mathbf{H}_\lambda \mathbf{y}$  is linear in  $\mathbf{y}$ . If we think about definition (3.60) in the context of a best subset selection of size  $k$ , it seems clear that  $\text{df}(\hat{y})$  will be larger than  $k$ , and this can be verified by estimating  $\text{Cov}(\hat{y}_i, y_i) / \sigma^2$  directly by simulation. However there is no closed form method for estimating  $\text{df}(\hat{y})$  for best subset selection.

**FIGURE 3.8.** Profiles of ridge coefficients for the prostate cancer example, as the tuning parameter  $\lambda$  is varied. Coefficients are plotted versus  $df(\lambda)$ , the effective degrees of freedom. A vertical line is drawn at  $df = 5.0$ , the value chosen by cross-validation.

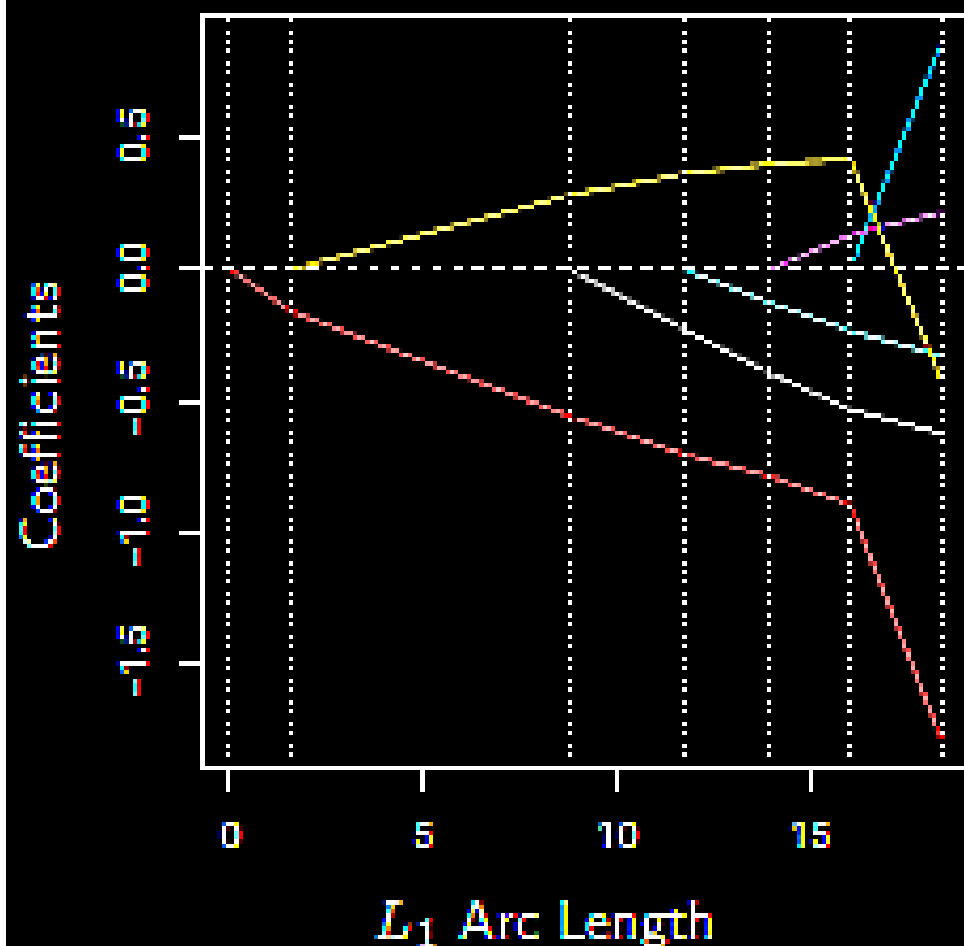


**FIGURE 3.10.** Profiles of lasso coefficients, as the tuning parameter  $t$  is varied. Coefficients are plotted versus  $s = t / \sum_1^p |\hat{\beta}_j|$ . A vertical line is drawn at  $s = 0.36$ , the value chosen by cross-validation. Compare Figure 3.8 on page 65; the lasso profiles hit zero, while those for ridge do not. The profiles are piece-wise linear, and so are computed only at the points displayed; see Section 3.4.4 for details.

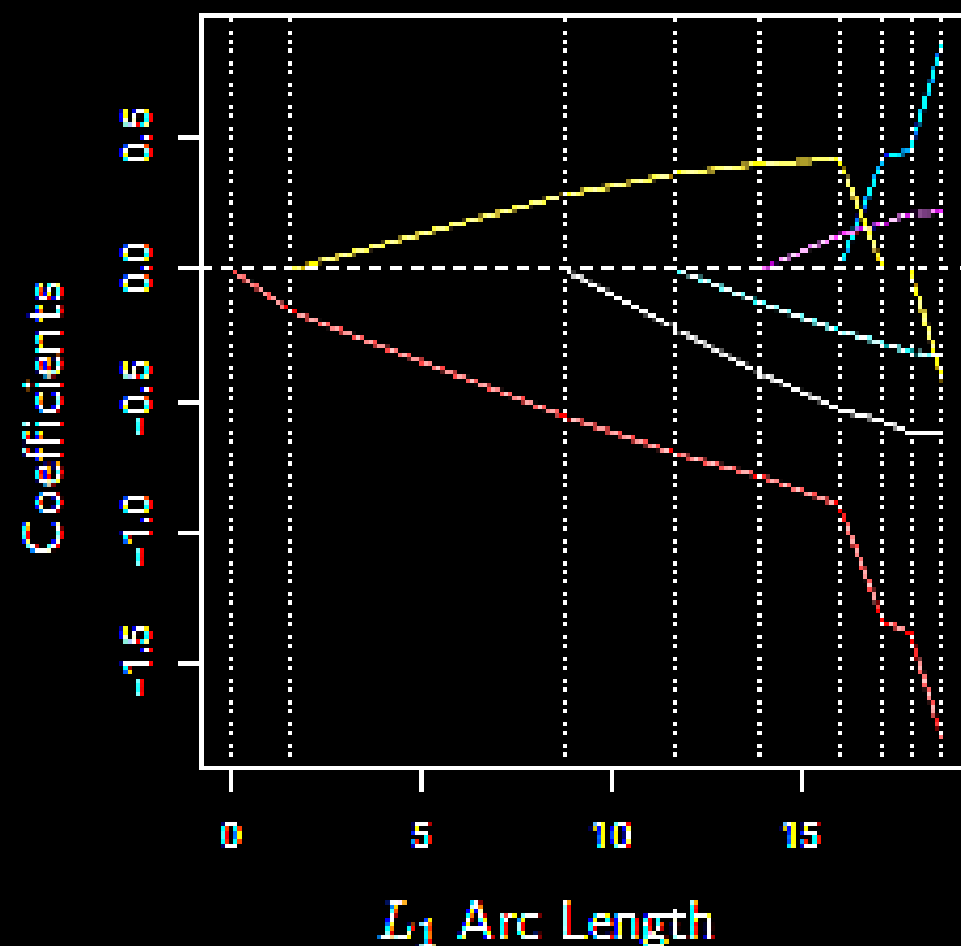


**FIGURE 3.14.** Progression of the absolute correlations during each step of the LAR procedure, using a simulated data set with six predictors. The labels at the top of the plot indicate which variables enter the active set at each step. The step length are measured in units of  $L_1$  arc length.

Least Angle Regression

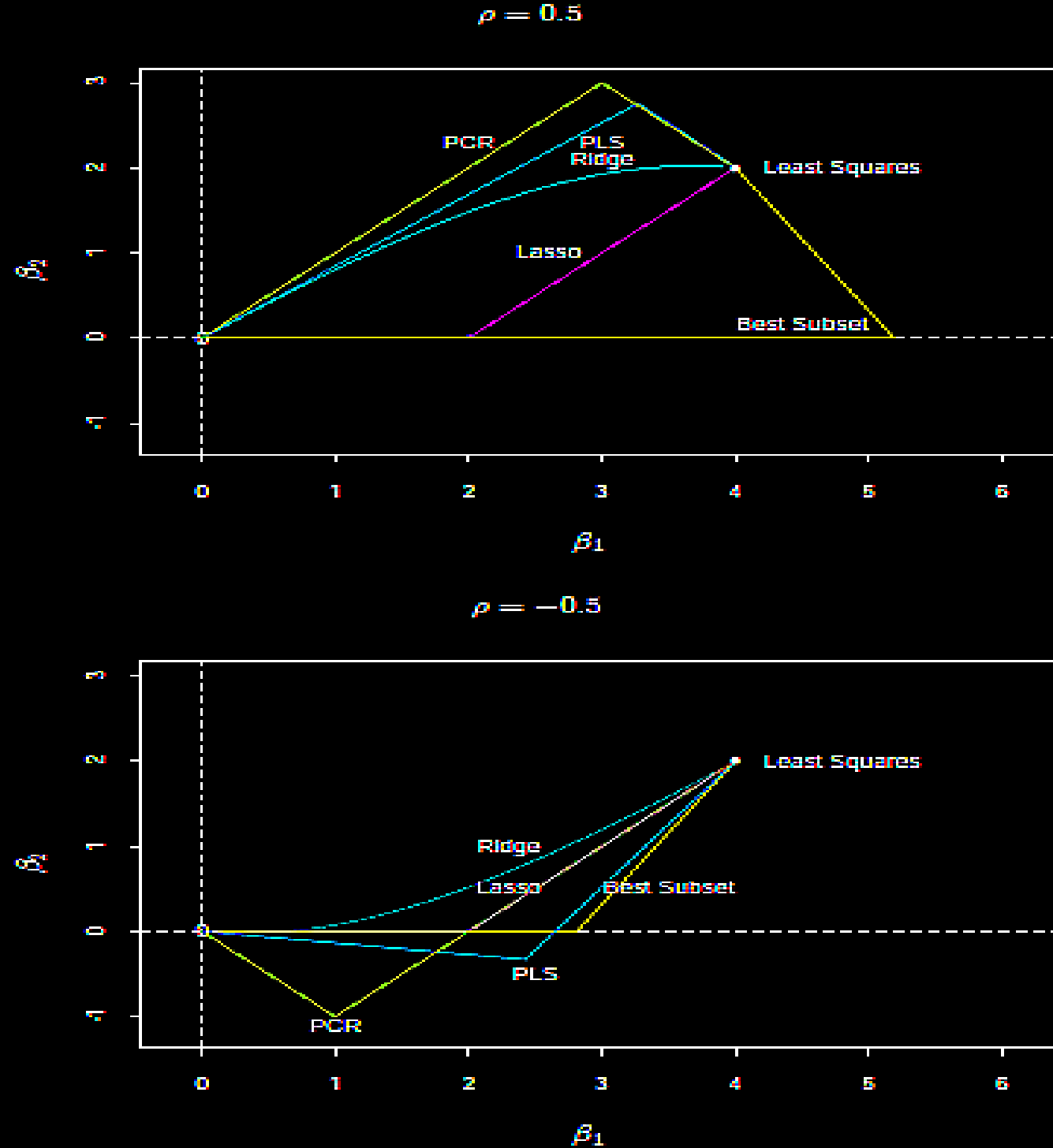


Lasso



**FIGURE 3.15.** Left panel shows the LAR coefficient profiles on the simulated data, as a function of the  $L_1$  arc length. The right panel shows the Lasso profile. They are identical until the dark-blue coefficient crosses zero at an arc length of about 18.





**FIGURE 3.18.** Coefficient profiles from different methods for a simple problem: two inputs with correlation  $\pm 0.5$ , and the true regression coefficients  $\beta = (4, 2)$ .

# ***Methods Using Derived Input Directions***

- In many situations we have a large number of inputs, often very correlated. The methods in this section produce a small number of linear combinations  $Z_m, m = 1, \dots, M$  of the original inputs  $X_j$ , and the  $Z_m$  are then used in place of the  $X_j$  as inputs in the regression. The methods differ in how the linear combinations are constructed.

## ***Principal Components Regression***

- - *<\* Left for self-study \*>*

# ***Partial Least Squares***

- This technique also constructs a set of linear combinations of the inputs for regression, but unlike principal components regression it uses  $y$  (in addition to  $\mathbf{X}$ ) for this construction. Like principal component regression, partial least squares (*PLS*) is not scale invariant, so we assume that each  $\mathbf{x}_j$  is standardized to have mean 0 and variance 1. *PLS* begins by computing  $\hat{\phi}_{1j} = \langle \mathbf{x}_j, \mathbf{y} \rangle$  for each  $j$ . From this we construct the derived input  $\mathbf{z}_1 = \sum_j \hat{\phi}_{1j} \mathbf{x}_j$ , which is the first partial least squares direction. The outcome  $y$  is regressed on  $\mathbf{z}_1$  giving coefficient  $\hat{\theta}_1$ , and then we orthogonalize  $\mathbf{x}_1, \dots, \mathbf{x}_p$  with respect to  $\mathbf{z}_1$ . We continue this process, until  $M \leq p$  directions have been obtained. In this manner, partial least squares produces a sequence of derived, orthogonal inputs or directions  $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_M$ . As with principal-component regression, if we were to construct all  $M = p$  directions, we would get back a solution equivalent to the usual least squares estimates; using  $M < p$  directions produces a reduced regression.

---

**Algorithm 3.3** *Partial Least Squares.*

---

1. Standardize each  $\mathbf{x}_j$  to have mean zero and variance one. Set  $\hat{\mathbf{y}}^{(0)} = \bar{y}\mathbf{1}$ , and  $\mathbf{x}_j^{(0)} = \mathbf{x}_j$ ,  $j = 1, \dots, p$ .
  2. For  $m = 1, 2, \dots, p$ 
    - (a)  $\mathbf{z}_m = \sum_{j=1}^p \hat{\varphi}_{mj} \mathbf{x}_j^{(m-1)}$ , where  $\hat{\varphi}_{mj} = \langle \mathbf{x}_j^{(m-1)}, \mathbf{y} \rangle$ .
    - (b)  $\hat{\theta}_m = \langle \mathbf{z}_m, \mathbf{y} \rangle / \langle \mathbf{z}_m, \mathbf{z}_m \rangle$ .
    - (c)  $\hat{\mathbf{y}}^{(m)} = \hat{\mathbf{y}}^{(m-1)} + \hat{\theta}_m \mathbf{z}_m$ .
    - (d) Orthogonalize each  $\mathbf{x}_j^{(m-1)}$  with respect to  $\mathbf{z}_m$ :  $\mathbf{x}_j^{(m)} = \mathbf{x}_j^{(m-1)} - [\langle \mathbf{z}_m, \mathbf{x}_j^{(m-1)} \rangle / \langle \mathbf{z}_m, \mathbf{z}_m \rangle] \mathbf{z}_m$ ,  $j = 1, 2, \dots, p$ .
  3. Output the sequence of fitted vectors  $\{\hat{\mathbf{y}}^{(m)}\}_1^p$ . Since the  $\{\mathbf{z}_\ell\}_1^m$  are linear in the original  $\mathbf{x}_j$ , so is  $\hat{\mathbf{y}}^{(m)} = \mathbf{X} \hat{\beta}^{\text{pls}}(m)$ . These linear coefficients can be recovered from the sequence of PLS transformations.
-

- What optimization problem is partial least squares solving? Since it uses the response  $y$  to construct its directions, its solution path is a nonlinear function of  $y$ . It can be shown (*Exercise 3.15*) that partial least squares seeks directions that have high variance and have high correlation with the response, in contrast to principal components regression which keys only on high variance the  $m^{\text{th}}$  principal component direction  $v_m$  solves:

$$\max_{\alpha} \text{Var}(\mathbf{X}_{\alpha}) \tag{3.63}$$

$$\text{subject to } \|\alpha\| = 1, \alpha^T \mathbf{S} v_{\ell} = 0, \ell = 1, \dots, m - 1,$$

- here  $S$  is the sample covariance matrix of the  $\mathbf{x}_j$  .

- The  $m$ th PLS direction  $\hat{\varphi}_m$  solves:

$$\max_{\alpha} \text{Corr}^2(\mathbf{y}, \mathbf{X}\alpha) \text{Var}(\mathbf{X}\alpha) \quad (3.64)$$

subject to  $\|\alpha\| = 1, \alpha^T \mathbf{S} \hat{\varphi}_\ell = 0, \ell = 1, \dots, m - 1,$

- If the input matrix  $\mathbf{X}$  is orthogonal, then partial least squares finds the least squares estimates after  $m = 1$  steps.

# ***Discussion: A Comparison of the Selection and Shrinkage Methods***

- To summarize, *PLS*, *PCR* and ridge regression tend to behave similarly.
- Ridge regression may be preferred because it shrinks smoothly, rather than in discrete steps.
- Lasso falls somewhere between ridge regression and best subset regression, and enjoys some of the properties of each.

---

## Algorithm 3.2 Least Angle Regression (revisit)

---

1. Standardize the predictors to have mean zero and unit norm. Start with the residual  $\mathbf{r} = \mathbf{y} - \bar{y}, \beta_1, \beta_2, \dots, \beta_p = 0$ .
2. Find the predictor  $\mathbf{x}_j$  most correlated with  $\mathbf{r}$ .
3. Move  $\beta_j$  from 0 towards its least-squares coefficient  $\langle \mathbf{x}_j, \mathbf{r} \rangle$ , until some other competitor  $\mathbf{x}_k$  has as much correlation with the current residual as does  $\mathbf{x}_j$ .
4. Move  $\beta_j$  and  $\beta_k$  in the direction defined by their joint least squares coefficient of the current residual on  $(\mathbf{x}_j, \mathbf{x}_k)$ , until some other competitor  $\mathbf{x}_l$  has as much correlation with the current residual.
5. Continue in this way until all  $p$  predictors have been entered. After  $\min(N - 1, p)$  steps, we arrive at the full least-squares solution.



- Suppose  $A_k$  is the active set of variables at the beginning of the  $k^{\text{th}}$  step, and let  $\beta_{A_k}$  be the coefficient vector for these variables at this step; there will be  $k - 1$  nonzero values, and the one just entered will be zero. If  $\mathbf{r}_k = \mathbf{y} - \mathbf{X}_{A_k}\beta_{A_k}$  is the current residual, then the direction for this step is

$$\delta_k = (\mathbf{X}_{A_k}^T \mathbf{X})^{-1} \mathbf{X}_{A_k}^T \mathbf{r}_k. \quad (3.55)$$

- The coefficient profile then evolves as  $\beta_{A_k}(\alpha) = \beta_{A_k} + \alpha \cdot \delta_k$ .
- By construction the coefficients in *LAR* change in a piecewise linear fashion.
- Read Eqns 3.56-3.59 for similarity between LAR vs Lasso (self study).

# ***Incremental Forward Stagewise Regression***

**Algorithm 3.4** *Incremental Forward Stagewise Regression— $FS_\epsilon$ .*

1. Start with the residual  $\mathbf{r}$  equal to  $\mathbf{y}$  and  $\beta_1, \beta_2, \dots, \beta_p = 0$ . All the predictors are standardized to have mean zero and unit norm.
2. Find the predictor  $\mathbf{x}_j$  most correlated with  $\mathbf{r}$
3. Update  $\beta_j \leftarrow \beta_j + \delta_j$ , where  $\delta_j = \epsilon \cdot \text{sign}[\langle \mathbf{x}_j, \mathbf{r} \rangle]$  and  $\epsilon > 0$  is a small step size, and set  $\mathbf{r} \leftarrow \mathbf{r} - \delta_j \mathbf{x}_j$ .
4. Repeat steps 2 and 3 many times, until the residuals are uncorrelated with all the predictors.

## Limiting procedure of $\varepsilon \rightarrow 0$ , is “**Infinitesimal forward stagewise regression**” or **FS<sub>0</sub>**

---

**Algorithm 3.2b** *Least Angle Regression: FS<sub>0</sub> Modification.*

---

4. Find the new direction by solving the constrained least squares problem

$$\min_b \|\mathbf{r} - \mathbf{X}_{\mathcal{A}}b\|_2^2 \text{ subject to } b_j s_j \geq 0, j \in \mathcal{A},$$

where  $s_j$  is the sign of  $\langle \mathbf{x}_j, \mathbf{r} \rangle$ .

---

Step 4, in Algm. 3.2:

Move  $\beta_j$  and  $\beta_k$  in the direction defined by their joint least squares coefficient of the current residual on  $(\mathbf{x}_j, \mathbf{x}_k)$ , until some other competitor  $\mathbf{x}_l$  has as much correlation with the current residual.

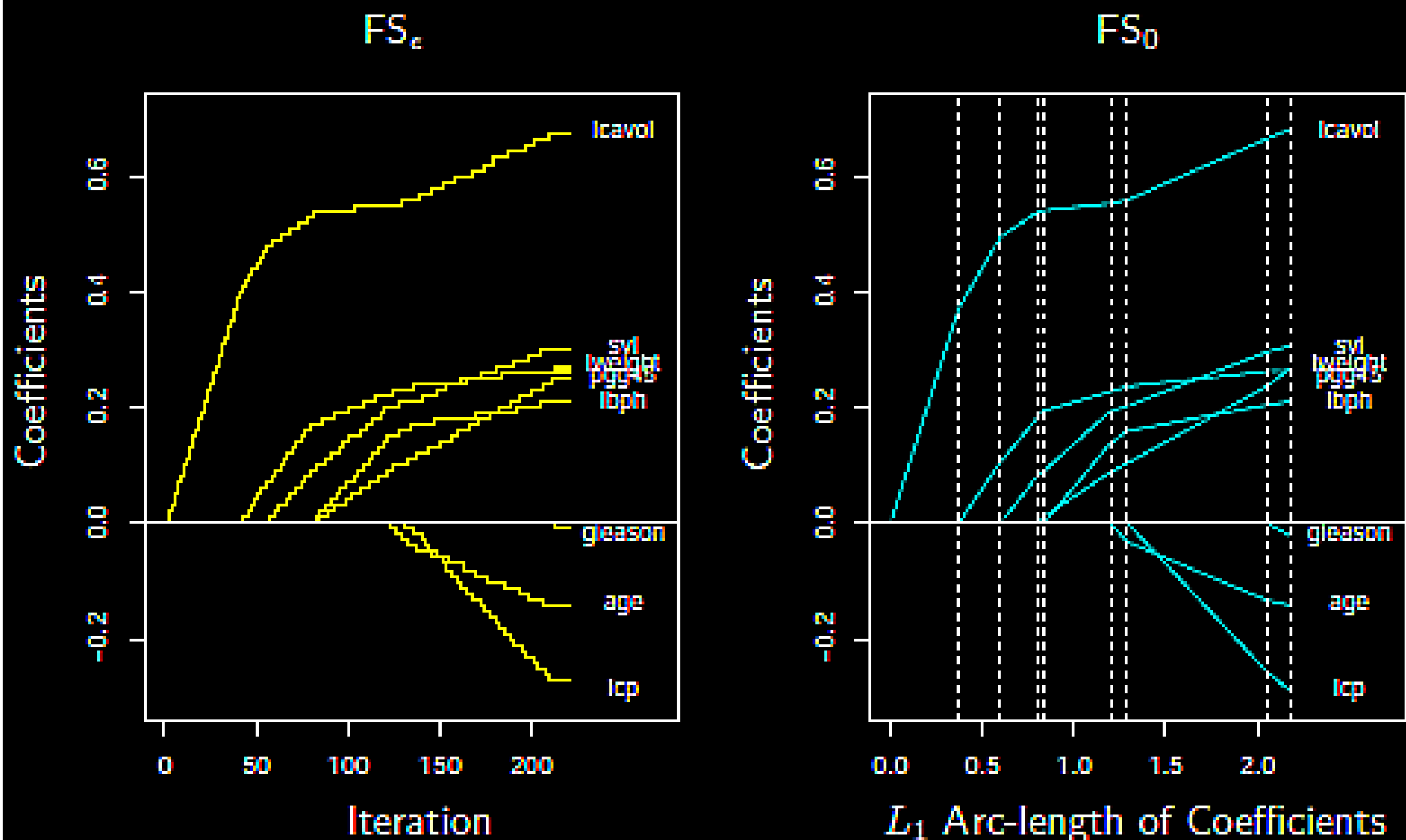


FIGURE 3.19. Coefficient profiles for the prostate data. The left panel shows incremental forward stagewise regression with step size  $\epsilon = 0.01$ . The right panel shows the infinitesimal version  $FS_0$  obtained letting  $\epsilon \rightarrow 0$ . This profile was fit by the modification 3.2b to the LAR Algorithm 3.2. In this example the  $FS_0$  profiles are monotone, and hence identical to those of lasso and LAR.

# *The Grouped Lasso*

- In some problems, the predictors belong to pre-defined groups; In this situation it may be desirable to shrink and select the members of a group together. The *grouped lasso* is one way to achieve this. Suppose that the  $p$  predictors are divided into  $L$  groups, with  $p_\ell$  the number in group  $\ell$ . For ease of notation, we use a matrix  $X_\ell$  to represent the predictors corresponding to the  $\ell$ th group, with corresponding coefficient vector  $\beta_\ell$ . The grouped-lasso minimizes the convex criterion

- $$\min_{\beta \in R^p} \left( \|y - \beta_0 \mathbf{1} - \sum_{\ell=1}^L X_\ell \beta_\ell\|_2^2 + \lambda \sum_{\ell=1}^L \sqrt{p_\ell} \|\beta_\ell\|_2 \right) \quad (3.80)$$

- where the  $\sqrt{p_\ell}$  terms accounts for the varying group sizes, and  $\|\cdot\|_2$  is the Euclidean norm (not squared).

- Since the Euclidean norm of a vector  $\beta_\ell$  is zero only if all of its components are zero, this procedure encourages sparsity at both the group and individual levels. That is, for some values of  $\lambda$ , an entire group of predictors may drop out of the model. This procedure was proposed by Bakin (1999) and Lin and Zhang (2006), and studied and generalized by Yuan and Lin (2007).

## ***Further Properties of the Lasso***

- A number of authors have studied the ability of the lasso and related procedures to recover the correct model, as  $N$  and  $p$  grow. Examples of this work include Knight and Fu (2000), Greenshtein and Ritov (2004), Tropp (2004), Donoho (2006b), Meinshausen (2007), Meinshausen and Bühlmann (2006), Tropp (2006), Zhao and Yu (2006), Wainwright (2006), and Bunea et al. (2007).

# ***Computational Considerations***

- Least squares fitting is usually done via the Cholesky decomposition of the matrix  $X^T X$  or a ***QR*** decomposition of  $X$ . With  $N$  observations and  $p$  features, the Cholesky decomposition requires  $p^3 + Np^2/2$  operations, while the ***QR*** decomposition requires  $Np^2$  operations. Depending on the relative size of  $N$  and  $p$ , the Cholesky can sometimes be faster; on the other hand, it can be less numerically stable (Lawson and Hansen, 1974). Computation of the lasso via the ***LAR*** algorithm has the same order of computation as a least squares fit.

