algorithm and lasso start to differ when an active coefficient passes through zero; condition (3.58) is violated for that variable, and it is kicked out of the active set $\mathcal{B}$. Exercise 3.23 shows that these equations imply a piecewise-linear coefficient profile as λ decreases. The stationarity conditions for the non-active variables require that

$$|\mathbf{x}_k^T(\mathbf{y} - \mathbf{X}\beta)| \leq \lambda, \ \forall k \notin \mathcal{B}, \tag{3.59}$$

which again agrees with the LAR algorithm.

Figure 3.16 compares LAR and lasso to forward stepwise and stagewise regression. The setup is the same as in Figure 3.6 on page 59, except here $N = 100$ here rather than 300, so the problem is more difficult. We see that the more aggressive forward stepwise starts to overfit quite early (well before the 10 true variables can enter the model), and ultimately performs worse than the slower forward stagewise regression. The behavior of LAR and lasso is similar to that of forward stagewise regression. Incremental forward stagewise is similar to LAR and lasso, and is described in Section 3.8.1.

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 $\hat{\mathbf{y}} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_N)$ as

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

Here $\mathrm{Cov}(\hat{y}_i, y_i)$ 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 $\mathrm{df}(\hat{\mathbf{y}})$. Expression (3.60) is a useful notion of degrees of freedom, one that can be applied to any model prediction $\hat{\mathbf{y}}$. This includes models that are

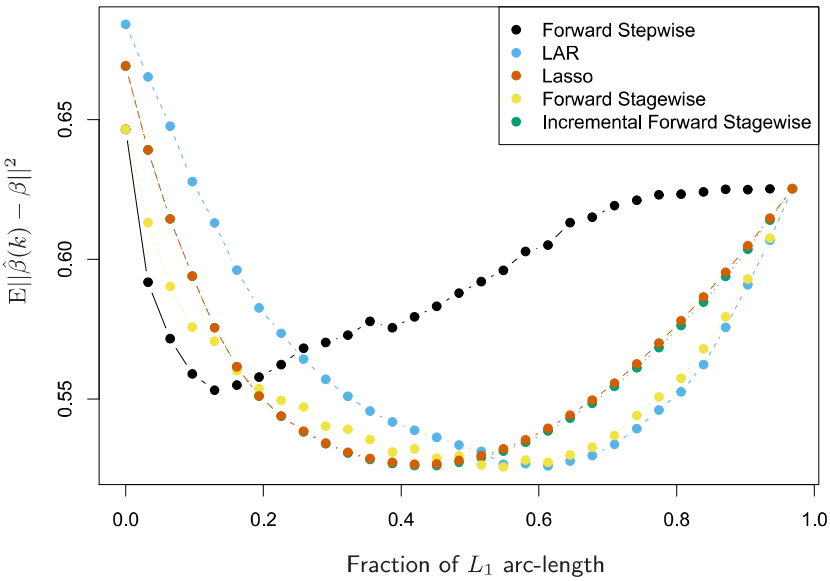

FIGURE 3.16. *Comparison of LAR and lasso with forward stepwise, forward stagewise (FS) and incremental forward stagewise (FS_0) regression. The setup is the same as in Figure 3.6, except $N = 100$ here rather than 300. Here the slower FS regression ultimately outperforms forward stepwise. LAR and lasso 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.*

adaptively fitted to the training data. This definition is motivated and discussed further in Sections 7.4–7.6.

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

For LAR and lasso, something magical happens. These techniques are adaptive in a smoother way than best subset selection, and hence estimation of degrees of freedom is more tractable. Specifically it can be shown that after the kth step of the LAR procedure, the effective degrees of freedom of the fit vector is exactly k. Now for the lasso, the (modified) LAR procedure

often takes more than p steps, since predictors can drop out. Hence the definition is a little different; for the lasso, at any stage $\mathrm{df}(\hat{\mathbf{y}})$ approximately equals the number of predictors in the model. While this approximation works reasonably well anywhere in the lasso path, for each k it works best at the *last* model in the sequence that contains k predictors. A detailed study of the degrees of freedom for the lasso may be found in Zou et al. (2007).

3.5 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, $m = 1, \ldots, M$ 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.

3.5.1 Principal Components Regression

In this approach the linear combinations Z_m used are the principal components as defined in Section 3.4.1 above.

Principal component regression forms the derived input columns $\mathbf{z}_m = \mathbf{X}v_m$, and then regresses $\mathbf{y}$ on $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_M$ for some $M \leq p$. Since the $\mathbf{z}_m$ are orthogonal, this regression is just a sum of univariate regressions:

$$\hat{\mathbf{y}}^{\mathrm{pcr}}_{(M)} = \bar{y}\mathbf{1} + \sum_{m=1}^{M} \hat{\theta}_m \mathbf{z}_m, \tag{3.61}$$

where $\hat{\theta}_m = \langle \mathbf{z}_m, \mathbf{y} \rangle / \langle \mathbf{z}_m, \mathbf{z}_m \rangle$. Since the $\mathbf{z}_m$ are each linear combinations of the original $\mathbf{x}_j$, we can express the solution (3.61) in terms of coefficients of the $\mathbf{x}_j$ (Exercise 3.13):

$$\hat{\beta}^{\mathrm{pcr}}(M) = \sum_{m=1}^{M} \hat{\theta}_m v_m. \tag{3.62}$$

As with ridge regression, principal components depend on the scaling of the inputs, so typically we first standardize them. Note that if $M = p$, we would just get back the usual least squares estimates, since the columns of $\mathbf{Z} = \mathbf{UD}$ span the column space of $\mathbf{X}$. For $M < p$ we get a reduced regression. We see that principal components regression is very similar to ridge regression: both operate via the principal components of the input matrix. Ridge regression shrinks the coefficients of the principal components (Figure 3.17), shrinking more depending on the size of the corresponding eigenvalue; principal components regression discards the $p - M$ smallest eigenvalue components. Figure 3.17 illustrates this.