

repeated the validation set approach using a different split of the data into a training set and a validation set, or if we repeated cross-validation using a different set of cross-validation folds, then the precise model with the lowest estimated test error would surely change. In this setting, we can select a model using the *one-standard-error rule*. We first calculate the standard error of the estimated test MSE for each model size, and then select the smallest model for which the estimated test error is within one standard error of the lowest point on the curve. The rationale here is that if a set of models appear to be more or less equally good, then we might as well choose the simplest model—that is, the model with the smallest number of predictors. In this case, applying the one-standard-error rule to the validation set or cross-validation approach leads to selection of the three-variable model.

one-  
standard-  
error  
rule

## 6.2 Shrinkage Methods

The subset selection methods described in Section 6.1 involve using least squares to fit a linear model that contains a subset of the predictors. As an alternative, we can fit a model containing all  $p$  predictors using a technique that *constrains* or *regularizes* the coefficient estimates, or equivalently, that *shrinks* the coefficient estimates towards zero. It may not be immediately obvious why such a constraint should improve the fit, but it turns out that shrinking the coefficient estimates can significantly reduce their variance. The two best-known techniques for shrinking the regression coefficients towards zero are *ridge regression* and the *lasso*.

### 6.2.1 Ridge Regression

Recall from Chapter 3 that the least squares fitting procedure estimates  $\beta_0, \beta_1, \dots, \beta_p$  using the values that minimize

$$\text{RSS} = \sum_{i=1}^n \left( y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2.$$

*Ridge regression* is very similar to least squares, except that the coefficients are estimated by minimizing a slightly different quantity. In particular, the ridge regression coefficient estimates  $\hat{\beta}^R$  are the values that minimize

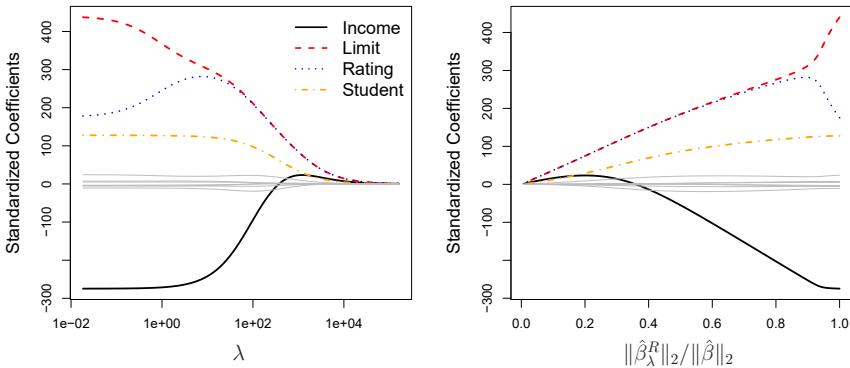
$$\sum_{i=1}^n \left( y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^p \beta_j^2 = \text{RSS} + \lambda \sum_{j=1}^p \beta_j^2, \quad (6.5)$$

where  $\lambda \geq 0$  is a *tuning parameter*, to be determined separately. Equation 6.5 trades off two different criteria. As with least squares, ridge regression seeks coefficient estimates that fit the data well, by making the RSS small. However, the second term,  $\lambda \sum_j \beta_j^2$ , called a *shrinkage penalty*, is small when  $\beta_1, \dots, \beta_p$  are close to zero, and so it has the effect of *shrinking* the estimates of  $\beta_j$  towards zero. The tuning parameter  $\lambda$  serves to control

ridge  
regression

tuning  
parameter

shrinkage  
penalty



**FIGURE 6.4.** The standardized ridge regression coefficients are displayed for the `Credit` data set, as a function of  $\lambda$  and  $\|\hat{\beta}_\lambda^R\|_2/\|\hat{\beta}\|_2$ .

the relative impact of these two terms on the regression coefficient estimates. When  $\lambda = 0$ , the penalty term has no effect, and ridge regression will produce the least squares estimates. However, as  $\lambda \rightarrow \infty$ , the impact of the shrinkage penalty grows, and the ridge regression coefficient estimates will approach zero. Unlike least squares, which generates only one set of coefficient estimates, ridge regression will produce a different set of coefficient estimates,  $\hat{\beta}_\lambda^R$ , for each value of  $\lambda$ . Selecting a good value for  $\lambda$  is critical; we defer this discussion to Section 6.2.3, where we use cross-validation.

Note that in (6.5), the shrinkage penalty is applied to  $\beta_1, \dots, \beta_p$ , but not to the intercept  $\beta_0$ . We want to shrink the estimated association of each variable with the response; however, we do not want to shrink the intercept, which is simply a measure of the mean value of the response when  $x_{i1} = x_{i2} = \dots = x_{ip} = 0$ . If we assume that the variables—that is, the columns of the data matrix  $\mathbf{X}$ —have been centered to have mean zero before ridge regression is performed, then the estimated intercept will take the form  $\hat{\beta}_0 = \bar{y} = \sum_{i=1}^n y_i/n$ .

### An Application to the Credit Data

In Figure 6.4, the ridge regression coefficient estimates for the `Credit` data set are displayed. In the left-hand panel, each curve corresponds to the ridge regression coefficient estimate for one of the ten variables, plotted as a function of  $\lambda$ . For example, the black solid line represents the ridge regression estimate for the `income` coefficient, as  $\lambda$  is varied. At the extreme left-hand side of the plot,  $\lambda$  is essentially zero, and so the corresponding ridge coefficient estimates are the same as the usual least squares estimates. But as  $\lambda$  increases, the ridge coefficient estimates shrink towards zero. When  $\lambda$  is extremely large, then all of the ridge coefficient estimates are basically zero; this corresponds to the *null model* that contains no predictors. In this plot, the `income`, `limit`, `rating`, and `student` variables are displayed in distinct colors, since these variables tend to have by far the largest coefficient estimates. While the ridge coefficient estimates tend to decrease in aggregate as  $\lambda$  increases, individual coefficients, such as `rating` and `income`, may occasionally increase as  $\lambda$  increases.

The right-hand panel of Figure 6.4 displays the same ridge coefficient estimates as the left-hand panel, but instead of displaying  $\lambda$  on the  $x$ -axis, we now display  $\|\hat{\beta}_\lambda^R\|_2/\|\hat{\beta}\|_2$ , where  $\hat{\beta}$  denotes the vector of least squares coefficient estimates. The notation  $\|\beta\|_2$  denotes the  $\ell_2$  norm (pronounced “ell 2”) of a vector, and is defined as  $\|\beta\|_2 = \sqrt{\sum_{j=1}^p \beta_j^2}$ . It measures the distance of  $\beta$  from zero. As  $\lambda$  increases, the  $\ell_2$  norm of  $\hat{\beta}_\lambda^R$  will *always* decrease, and so will  $\|\hat{\beta}_\lambda^R\|_2/\|\hat{\beta}\|_2$ . The latter quantity ranges from 1 (when  $\lambda = 0$ , in which case the ridge regression coefficient estimate is the same as the least squares estimate, and so their  $\ell_2$  norms are the same) to 0 (when  $\lambda = \infty$ , in which case the ridge regression coefficient estimate is a vector of zeros, with  $\ell_2$  norm equal to zero). Therefore, we can think of the  $x$ -axis in the right-hand panel of Figure 6.4 as the amount that the ridge regression coefficient estimates have been shrunk towards zero; a small value indicates that they have been shrunk very close to zero.

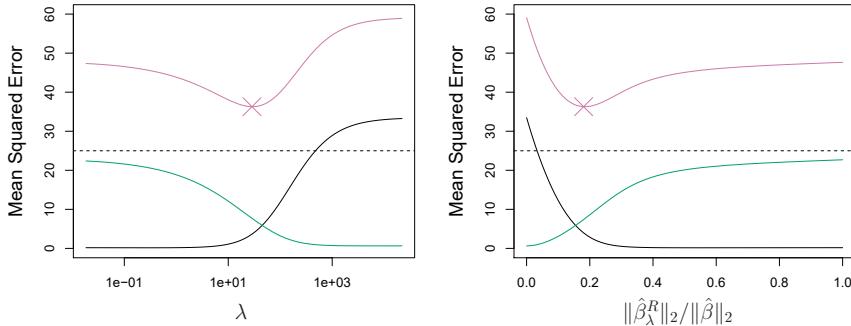
The standard least squares coefficient estimates discussed in Chapter 3 are *scale equivariant*: multiplying  $X_j$  by a constant  $c$  simply leads to a scaling of the least squares coefficient estimates by a factor of  $1/c$ . In other words, regardless of how the  $j$ th predictor is scaled,  $X_j \hat{\beta}_j$  will remain the same. In contrast, the ridge regression coefficient estimates can change *substantially* when multiplying a given predictor by a constant. For instance, consider the `income` variable, which is measured in dollars. One could reasonably have measured income in thousands of dollars, which would result in a reduction in the observed values of `income` by a factor of 1,000. Now due to the sum of squared coefficients term in the ridge regression formulation (6.5), such a change in scale will not simply cause the ridge regression coefficient estimate for `income` to change by a factor of 1,000. In other words,  $X_j \hat{\beta}_{j,\lambda}^R$  will depend not only on the value of  $\lambda$ , but also on the scaling of the  $j$ th predictor. In fact, the value of  $X_j \hat{\beta}_{j,\lambda}^R$  may even depend on the scaling of the *other* predictors! Therefore, it is best to apply ridge regression after *standardizing the predictors*, using the formula

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n} \sum_{i=1}^n (x_{ij} - \bar{x}_j)^2}}, \quad (6.6)$$

so that they are all on the same scale. In (6.6), the denominator is the estimated standard deviation of the  $j$ th predictor. Consequently, all of the standardized predictors will have a standard deviation of one. As a result the final fit will not depend on the scale on which the predictors are measured. In Figure 6.4, the  $y$ -axis displays the standardized ridge regression coefficient estimates—that is, the coefficient estimates that result from performing ridge regression using standardized predictors.

### Why Does Ridge Regression Improve Over Least Squares?

Ridge regression’s advantage over least squares is rooted in the *bias-variance trade-off*. As  $\lambda$  increases, the flexibility of the ridge regression fit decreases, leading to decreased variance but increased bias. This is illustrated in the left-hand panel of Figure 6.5, using a simulated data set containing  $p = 45$  predictors and  $n = 50$  observations. The green curve in the left-hand panel



**FIGURE 6.5.** Squared bias (black), variance (green), and test mean squared error (purple) for the ridge regression predictions on a simulated data set, as a function of  $\lambda$  and  $\|\hat{\beta}_\lambda^R\|_2 / \|\hat{\beta}\|_2$ . The horizontal dashed lines indicate the minimum possible MSE. The purple crosses indicate the ridge regression models for which the MSE is smallest.

of Figure 6.5 displays the variance of the ridge regression predictions as a function of  $\lambda$ . At the least squares coefficient estimates, which correspond to ridge regression with  $\lambda = 0$ , the variance is high but there is no bias. But as  $\lambda$  increases, the shrinkage of the ridge coefficient estimates leads to a substantial reduction in the variance of the predictions, at the expense of a slight increase in bias. Recall that the test mean squared error (MSE), plotted in purple, is closely related to the variance plus the squared bias. For values of  $\lambda$  up to about 10, the variance decreases rapidly, with very little increase in bias, plotted in black. Consequently, the MSE drops considerably as  $\lambda$  increases from 0 to 10. Beyond this point, the decrease in variance due to increasing  $\lambda$  slows, and the shrinkage on the coefficients causes them to be significantly underestimated, resulting in a large increase in the bias. The minimum MSE is achieved at approximately  $\lambda = 30$ . Interestingly, because of its high variance, the MSE associated with the least squares fit, when  $\lambda = 0$ , is almost as high as that of the null model for which all coefficient estimates are zero, when  $\lambda = \infty$ . However, for an intermediate value of  $\lambda$ , the MSE is considerably lower.

The right-hand panel of Figure 6.5 displays the same curves as the left-hand panel, this time plotted against the  $\ell_2$  norm of the ridge regression coefficient estimates divided by the  $\ell_2$  norm of the least squares estimates. Now as we move from left to right, the fits become more flexible, and so the bias decreases and the variance increases.

In general, in situations where the relationship between the response and the predictors is close to linear, the least squares estimates will have low bias but may have high variance. This means that a small change in the training data can cause a large change in the least squares coefficient estimates. In particular, when the number of variables  $p$  is almost as large as the number of observations  $n$ , as in the example in Figure 6.5, the least squares estimates will be extremely variable. And if  $p > n$ , then the least squares estimates do not even have a unique solution, whereas ridge regression can still perform well by trading off a small increase in bias for a

large decrease in variance. Hence, ridge regression works best in situations where the least squares estimates have high variance.

Ridge regression also has substantial computational advantages over best subset selection, which requires searching through  $2^p$  models. As we discussed previously, even for moderate values of  $p$ , such a search can be computationally infeasible. In contrast, for any fixed value of  $\lambda$ , ridge regression only fits a single model, and the model-fitting procedure can be performed quite quickly. In fact, one can show that the computations required to solve (6.5), *simultaneously for all values of  $\lambda$* , are almost identical to those for fitting a model using least squares.

### 6.2.2 The Lasso

Ridge regression does have one obvious disadvantage. Unlike best subset, forward stepwise, and backward stepwise selection, which will generally select models that involve just a subset of the variables, ridge regression will include all  $p$  predictors in the final model. The penalty  $\lambda \sum \beta_j^2$  in (6.5) will shrink all of the coefficients towards zero, but it will not set any of them exactly to zero (unless  $\lambda = \infty$ ). This may not be a problem for prediction accuracy, but it can create a challenge in model interpretation in settings in which the number of variables  $p$  is quite large. For example, in the `Credit` data set, it appears that the most important variables are `income`, `limit`, `rating`, and `student`. So we might wish to build a model including just these predictors. However, ridge regression will always generate a model involving all ten predictors. Increasing the value of  $\lambda$  will tend to reduce the magnitudes of the coefficients, but will not result in exclusion of any of the variables.

The *lasso* is a relatively recent alternative to ridge regression that overcomes this disadvantage. The lasso coefficients,  $\hat{\beta}_\lambda^L$ , minimize the quantity

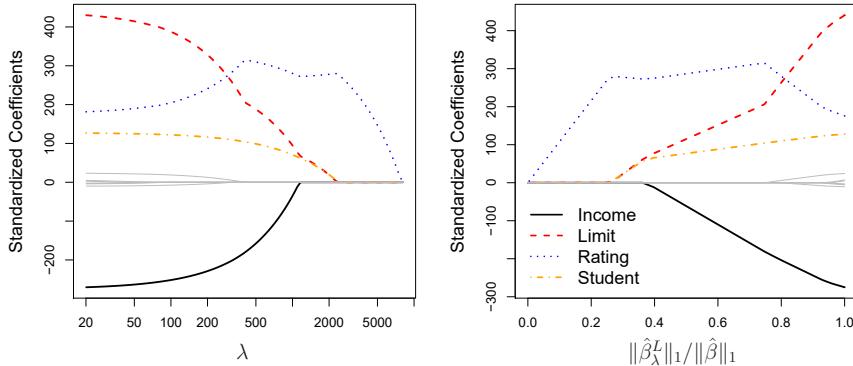
$$\sum_{i=1}^n \left( y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^p |\beta_j| = \text{RSS} + \lambda \sum_{j=1}^p |\beta_j|. \quad (6.7)$$

Comparing (6.7) to (6.5), we see that the lasso and ridge regression have similar formulations. The only difference is that the  $\beta_j^2$  term in the ridge regression penalty (6.5) has been replaced by  $|\beta_j|$  in the lasso penalty (6.7). In statistical parlance, the lasso uses an  $\ell_1$  (pronounced “ell 1”) penalty instead of an  $\ell_2$  penalty. The  $\ell_1$  norm of a coefficient vector  $\beta$  is given by  $\|\beta\|_1 = \sum |\beta_j|$ .

As with ridge regression, the lasso shrinks the coefficient estimates towards zero. However, in the case of the lasso, the  $\ell_1$  penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when the tuning parameter  $\lambda$  is sufficiently large. Hence, much like best subset selection, the lasso performs *variable selection*. As a result, models generated from the lasso are generally much easier to interpret than those produced by ridge regression. We say that the lasso yields *sparse* models—that is, models that involve only a subset of the variables. As in ridge regression, selecting a good value of  $\lambda$  for the lasso is critical; we defer this discussion to Section 6.2.3, where we use cross-validation.

lasso

sparse



**FIGURE 6.6.** The standardized lasso coefficients on the `Credit` data set are shown as a function of  $\lambda$  and  $\|\hat{\beta}_\lambda^L\|_1 / \|\hat{\beta}\|_1$ .

As an example, consider the coefficient plots in Figure 6.6, which are generated from applying the lasso to the `Credit` data set. When  $\lambda = 0$ , then the lasso simply gives the least squares fit, and when  $\lambda$  becomes sufficiently large, the lasso gives the null model in which all coefficient estimates equal zero. However, in between these two extremes, the ridge regression and lasso models are quite different from each other. Moving from left to right in the right-hand panel of Figure 6.6, we observe that at first the lasso results in a model that contains only the `rating` predictor. Then `student` and `limit` enter the model almost simultaneously, shortly followed by `income`. Eventually, the remaining variables enter the model. Hence, depending on the value of  $\lambda$ , the lasso can produce a model involving any number of variables. In contrast, ridge regression will always include all of the variables in the model, although the magnitude of the coefficient estimates will depend on  $\lambda$ .

### Another Formulation for Ridge Regression and the Lasso

One can show that the lasso and ridge regression coefficient estimates solve the problems

$$\underset{\beta}{\text{minimize}} \left\{ \sum_{i=1}^n \left( y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^p |\beta_j| \leq s \quad (6.8)$$

and

$$\underset{\beta}{\text{minimize}} \left\{ \sum_{i=1}^n \left( y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^p \beta_j^2 \leq s, \quad (6.9)$$

respectively. In other words, for every value of  $\lambda$ , there is some  $s$  such that the Equations (6.7) and (6.8) will give the same lasso coefficient estimates. Similarly, for every value of  $\lambda$  there is a corresponding  $s$  such that Equations (6.5) and (6.9) will give the same ridge regression coefficient estimates.

When  $p = 2$ , then (6.8) indicates that the lasso coefficient estimates have the smallest RSS out of all points that lie within the diamond defined by  $|\beta_1| + |\beta_2| \leq s$ . Similarly, the ridge regression estimates have the smallest RSS out of all points that lie within the circle defined by  $\beta_1^2 + \beta_2^2 \leq s$ .

We can think of (6.8) as follows. When we perform the lasso we are trying to find the set of coefficient estimates that lead to the smallest RSS, subject to the constraint that there is a *budget*  $s$  for how large  $\sum_{j=1}^p |\beta_j|$  can be. When  $s$  is extremely large, then this budget is not very restrictive, and so the coefficient estimates can be large. In fact, if  $s$  is large enough that the least squares solution falls within the budget, then (6.8) will simply yield the least squares solution. In contrast, if  $s$  is small, then  $\sum_{j=1}^p |\beta_j|$  must be small in order to avoid violating the budget. Similarly, (6.9) indicates that when we perform ridge regression, we seek a set of coefficient estimates such that the RSS is as small as possible, subject to the requirement that  $\sum_{j=1}^p \beta_j^2$  not exceed the budget  $s$ .

The formulations (6.8) and (6.9) reveal a close connection between the lasso, ridge regression, and best subset selection. Consider the problem

$$\underset{\beta}{\text{minimize}} \left\{ \sum_{i=1}^n \left( y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^p I(\beta_j \neq 0) \leq s. \quad (6.10)$$

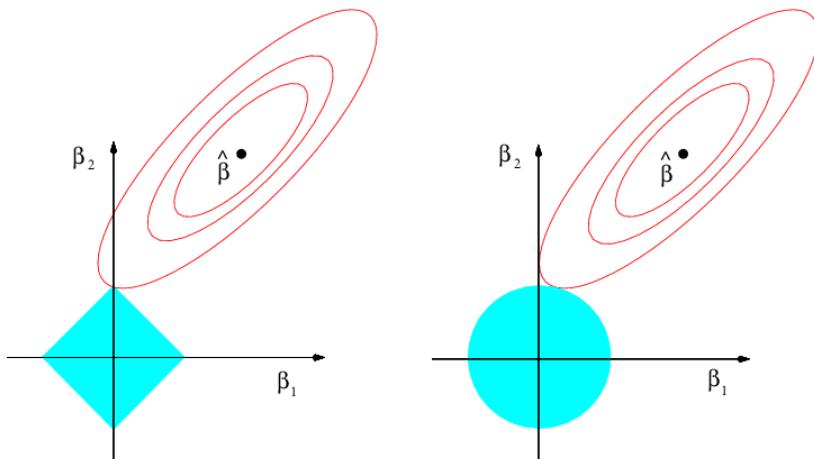
Here  $I(\beta_j \neq 0)$  is an indicator variable: it takes on a value of 1 if  $\beta_j \neq 0$ , and equals zero otherwise. Then (6.10) amounts to finding a set of coefficient estimates such that RSS is as small as possible, subject to the constraint that no more than  $s$  coefficients can be nonzero. The problem (6.10) is equivalent to best subset selection. Unfortunately, solving (6.10) is computationally infeasible when  $p$  is large, since it requires considering all  $\binom{p}{s}$  models containing  $s$  predictors. Therefore, we can interpret ridge regression and the lasso as computationally feasible alternatives to best subset selection that replace the intractable form of the budget in (6.10) with forms that are much easier to solve. Of course, the lasso is much more closely related to best subset selection, since the lasso performs feature selection for  $s$  sufficiently small in (6.8), while ridge regression does not.

### The Variable Selection Property of the Lasso

Why is it that the lasso, unlike ridge regression, results in coefficient estimates that are exactly equal to zero? The formulations (6.8) and (6.9) can be used to shed light on the issue. Figure 6.7 illustrates the situation. The least squares solution is marked as  $\hat{\beta}$ , while the blue diamond and circle represent the lasso and ridge regression constraints in (6.8) and (6.9), respectively. If  $s$  is sufficiently large, then the constraint regions will contain  $\hat{\beta}$ , and so the ridge regression and lasso estimates will be the same as the least squares estimates. (Such a large value of  $s$  corresponds to  $\lambda = 0$  in (6.5) and (6.7).) However, in Figure 6.7 the least squares estimates lie outside of the diamond and the circle, and so the least squares estimates are not the same as the lasso and ridge regression estimates.

Each of the ellipses centered around  $\hat{\beta}$  represents a *contour*: this means that all of the points on a particular ellipse have the same RSS value. As

contour



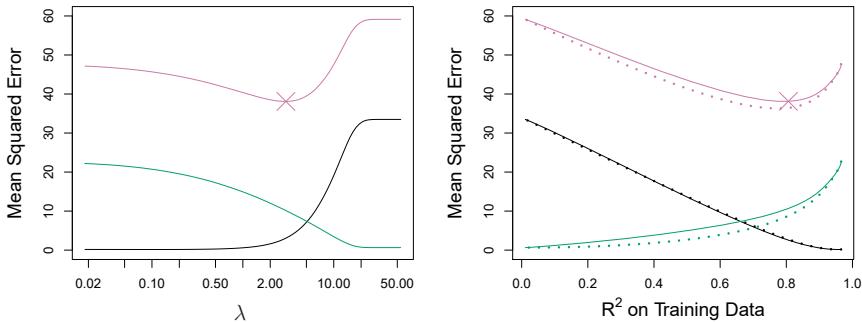
**FIGURE 6.7.** Contours of the error and constraint functions for the lasso (left) and ridge regression (right). The solid blue areas are the constraint regions,  $|\beta_1| + |\beta_2| \leq s$  and  $\beta_1^2 + \beta_2^2 \leq s$ , while the red ellipses are the contours of the RSS.

the ellipses expand away from the least squares coefficient estimates, the RSS increases. Equations (6.8) and (6.9) indicate that the lasso and ridge regression coefficient estimates are given by the first point at which an ellipse contacts the constraint region. Since ridge regression has a circular constraint with no sharp points, this intersection will not generally occur on an axis, and so the ridge regression coefficient estimates will be exclusively non-zero. However, the lasso constraint has *corners* at each of the axes, and so the ellipse will often intersect the constraint region at an axis. When this occurs, one of the coefficients will equal zero. In higher dimensions, many of the coefficient estimates may equal zero simultaneously. In Figure 6.7, the intersection occurs at  $\beta_1 = 0$ , and so the resulting model will only include  $\beta_2$ .

In Figure 6.7, we considered the simple case of  $p = 2$ . When  $p = 3$ , then the constraint region for ridge regression becomes a sphere, and the constraint region for the lasso becomes a polyhedron. When  $p > 3$ , the constraint for ridge regression becomes a hypersphere, and the constraint for the lasso becomes a polytope. However, the key ideas depicted in Figure 6.7 still hold. In particular, the lasso leads to feature selection when  $p > 2$  due to the sharp corners of the polyhedron or polytope.

### Comparing the Lasso and Ridge Regression

It is clear that the lasso has a major advantage over ridge regression, in that it produces simpler and more interpretable models that involve only a subset of the predictors. However, which method leads to better prediction accuracy? Figure 6.8 displays the variance, squared bias, and test MSE of the lasso applied to the same simulated data as in Figure 6.5. Clearly the lasso leads to qualitatively similar behavior to ridge regression, in that as  $\lambda$  increases, the variance decreases and the bias increases. In the right-hand



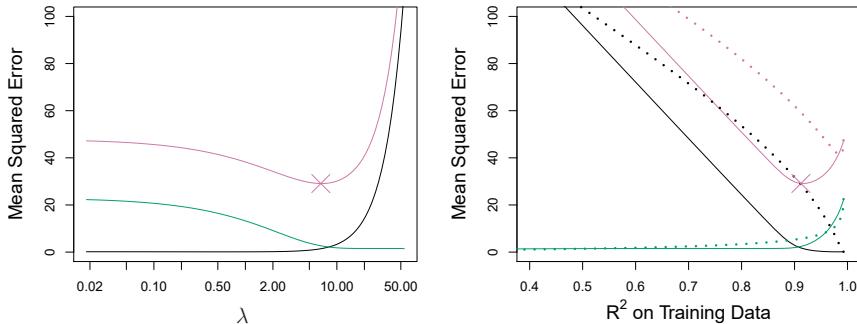
**FIGURE 6.8.** Left: Plots of squared bias (black), variance (green), and test MSE (purple) for the lasso on a simulated data set. Right: Comparison of squared bias, variance, and test MSE between lasso (solid) and ridge (dotted). Both are plotted against their  $R^2$  on the training data, as a common form of indexing. The crosses in both plots indicate the lasso model for which the MSE is smallest.

panel of Figure 6.8, the dotted lines represent the ridge regression fits. Here we plot both against their  $R^2$  on the training data. This is another useful way to index models, and can be used to compare models with different types of regularization, as is the case here. In this example, the lasso and ridge regression result in almost identical biases. However, the variance of ridge regression is slightly lower than the variance of the lasso. Consequently, the minimum MSE of ridge regression is slightly smaller than that of the lasso.

However, the data in Figure 6.8 were generated in such a way that all 45 predictors were related to the response—that is, none of the true coefficients  $\beta_1, \dots, \beta_{45}$  equaled zero. The lasso implicitly assumes that a number of the coefficients truly equal zero. Consequently, it is not surprising that ridge regression outperforms the lasso in terms of prediction error in this setting. Figure 6.9 illustrates a similar situation, except that now the response is a function of only 2 out of 45 predictors. Now the lasso tends to outperform ridge regression in terms of bias, variance, and MSE.

These two examples illustrate that neither ridge regression nor the lasso will universally dominate the other. In general, one might expect the lasso to perform better in a setting where a relatively small number of predictors have substantial coefficients, and the remaining predictors have coefficients that are very small or that equal zero. Ridge regression will perform better when the response is a function of many predictors, all with coefficients of roughly equal size. However, the number of predictors that is related to the response is never known *a priori* for real data sets. A technique such as cross-validation can be used in order to determine which approach is better on a particular data set.

As with ridge regression, when the least squares estimates have excessively high variance, the lasso solution can yield a reduction in variance at the expense of a small increase in bias, and consequently can generate more accurate predictions. Unlike ridge regression, the lasso performs variable selection, and hence results in models that are easier to interpret.



**FIGURE 6.9.** Left: Plots of squared bias (black), variance (green), and test MSE (purple) for the lasso. The simulated data is similar to that in Figure 6.8, except that now only two predictors are related to the response. Right: Comparison of squared bias, variance, and test MSE between lasso (solid) and ridge (dotted). Both are plotted against their  $R^2$  on the training data, as a common form of indexing. The crosses in both plots indicate the lasso model for which the MSE is smallest.

There are very efficient algorithms for fitting both ridge and lasso models; in both cases the entire coefficient paths can be computed with about the same amount of work as a single least squares fit. We will explore this further in the lab at the end of this chapter.

### A Simple Special Case for Ridge Regression and the Lasso

In order to obtain a better intuition about the behavior of ridge regression and the lasso, consider a simple special case with  $n = p$ , and  $\mathbf{X}$  a diagonal matrix with 1's on the diagonal and 0's in all off-diagonal elements. To simplify the problem further, assume also that we are performing regression without an intercept. With these assumptions, the usual least squares problem simplifies to finding  $\beta_1, \dots, \beta_p$  that minimize

$$\sum_{j=1}^p (y_j - \beta_j)^2. \quad (6.11)$$

In this case, the least squares solution is given by

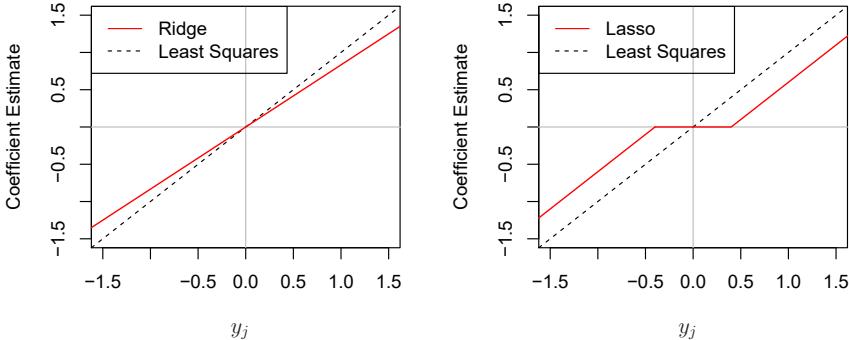
$$\hat{\beta}_j = y_j.$$

And in this setting, ridge regression amounts to finding  $\beta_1, \dots, \beta_p$  such that

$$\sum_{j=1}^p (y_j - \beta_j)^2 + \lambda \sum_{j=1}^p \beta_j^2 \quad (6.12)$$

is minimized, and the lasso amounts to finding the coefficients such that

$$\sum_{j=1}^p (y_j - \beta_j)^2 + \lambda \sum_{j=1}^p |\beta_j| \quad (6.13)$$



**FIGURE 6.10.** The ridge regression and lasso coefficient estimates for a simple setting with  $n = p$  and  $\mathbf{X}$  a diagonal matrix with 1's on the diagonal. Left: The ridge regression coefficient estimates are shrunken proportionally towards zero, relative to the least squares estimates. Right: The lasso coefficient estimates are soft-thresholded towards zero.

is minimized. One can show that in this setting, the ridge regression estimates take the form

$$\hat{\beta}_j^R = y_j / (1 + \lambda), \quad (6.14)$$

and the lasso estimates take the form

$$\hat{\beta}_j^L = \begin{cases} y_j - \lambda/2 & \text{if } y_j > \lambda/2; \\ y_j + \lambda/2 & \text{if } y_j < -\lambda/2; \\ 0 & \text{if } |y_j| \leq \lambda/2. \end{cases} \quad (6.15)$$

Figure 6.10 displays the situation. We can see that ridge regression and the lasso perform two very different types of shrinkage. In ridge regression, each least squares coefficient estimate is shrunken by the same proportion. In contrast, the lasso shrinks each least squares coefficient towards zero by a constant amount,  $\lambda/2$ ; the least squares coefficients that are less than  $\lambda/2$  in absolute value are shrunk entirely to zero. The type of shrinkage performed by the lasso in this simple setting (6.15) is known as *soft-thresholding*. The fact that some lasso coefficients are shrunk entirely to zero explains why the lasso performs feature selection.

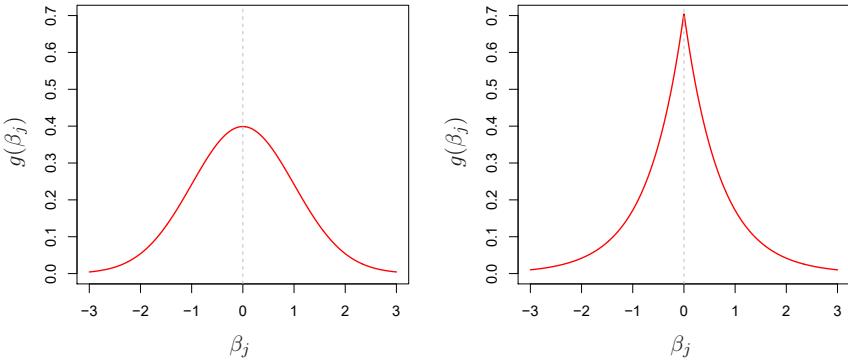
soft-thresholding

In the case of a more general data matrix  $\mathbf{X}$ , the story is a little more complicated than what is depicted in Figure 6.10, but the main ideas still hold approximately: ridge regression more or less shrinks every dimension of the data by the same proportion, whereas the lasso more or less shrinks all coefficients toward zero by a similar amount, and sufficiently small coefficients are shrunk all the way to zero.

### Bayesian Interpretation of Ridge Regression and the Lasso



We now show that one can view ridge regression and the lasso through a Bayesian lens. A Bayesian viewpoint for regression assumes that the coefficient vector  $\beta$  has some *prior* distribution, say  $p(\beta)$ , where  $\beta = (\beta_0, \beta_1, \dots, \beta_p)^T$ . The likelihood of the data can be written as  $f(Y|X, \beta)$ ,



**FIGURE 6.11.** Left: Ridge regression is the posterior mode for  $\beta$  under a Gaussian prior. Right: The lasso is the posterior mode for  $\beta$  under a double-exponential prior.

where  $X = (X_1, \dots, X_p)$ . Multiplying the prior distribution by the likelihood gives us (up to a proportionality constant) the *posterior distribution*, which takes the form

$$p(\beta|X, Y) \propto f(Y|X, \beta)p(\beta|X) = f(Y|X, \beta)p(\beta),$$

where the proportionality above follows from Bayes' theorem, and the equality above follows from the assumption that  $X$  is fixed.

We assume the usual linear model,

$$Y = \beta_0 + X_1\beta_1 + \dots + X_p\beta_p + \epsilon,$$

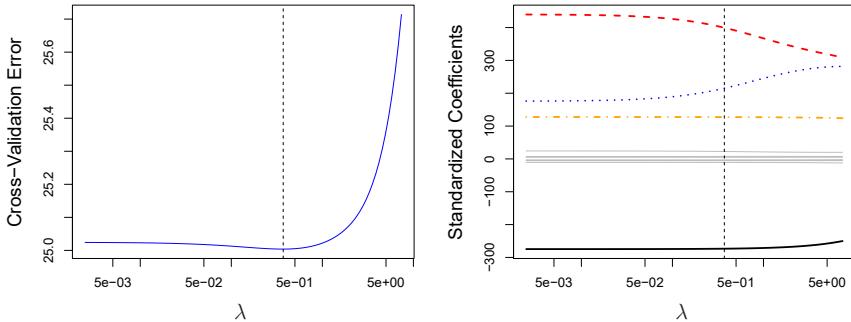
and suppose that the errors are independent and drawn from a normal distribution. Furthermore, assume that  $p(\beta) = \prod_{j=1}^p g(\beta_j)$ , for some density function  $g$ . It turns out that ridge regression and the lasso follow naturally from two special cases of  $g$ :

- If  $g$  is a Gaussian distribution with mean zero and standard deviation a function of  $\lambda$ , then it follows that the *posterior mode* for  $\beta$ —that is, the most likely value for  $\beta$ , given the data—is given by the ridge regression solution. (In fact, the ridge regression solution is also the posterior mean.)
- If  $g$  is a double-exponential (Laplace) distribution with mean zero and scale parameter a function of  $\lambda$ , then it follows that the posterior mode for  $\beta$  is the lasso solution. (However, the lasso solution is *not* the posterior mean, and in fact, the posterior mean does not yield a sparse coefficient vector.)

posterior distribution

posterior mode

The Gaussian and double-exponential priors are displayed in Figure 6.11. Therefore, from a Bayesian viewpoint, ridge regression and the lasso follow directly from assuming the usual linear model with normal errors, together with a simple prior distribution for  $\beta$ . Notice that the lasso prior is steeply peaked at zero, while the Gaussian is flatter and fatter at zero. Hence, the lasso expects a priori that many of the coefficients are (exactly) zero, while ridge assumes the coefficients are randomly distributed about zero.



**FIGURE 6.12.** Left: Cross-validation errors that result from applying ridge regression to the `Credit` data set with various values of  $\lambda$ . Right: The coefficient estimates as a function of  $\lambda$ . The vertical dashed lines indicate the value of  $\lambda$  selected by cross-validation.

### 6.2.3 Selecting the Tuning Parameter

Just as the subset selection approaches considered in Section 6.1 require a method to determine which of the models under consideration is best, implementing ridge regression and the lasso requires a method for selecting a value for the tuning parameter  $\lambda$  in (6.5) and (6.7), or equivalently, the value of the constraint  $s$  in (6.9) and (6.8). Cross-validation provides a simple way to tackle this problem. We choose a grid of  $\lambda$  values, and compute the cross-validation error for each value of  $\lambda$ , as described in Chapter 5. We then select the tuning parameter value for which the cross-validation error is smallest. Finally, the model is re-fit using all of the available observations and the selected value of the tuning parameter.

Figure 6.12 displays the choice of  $\lambda$  that results from performing leave-one-out cross-validation on the ridge regression fits from the `Credit` data set. The dashed vertical lines indicate the selected value of  $\lambda$ . In this case the value is relatively small, indicating that the optimal fit only involves a small amount of shrinkage relative to the least squares solution. In addition, the dip is not very pronounced, so there is rather a wide range of values that would give a very similar error. In a case like this we might simply use the least squares solution.

Figure 6.13 provides an illustration of ten-fold cross-validation applied to the lasso fits on the sparse simulated data from Figure 6.9. The left-hand panel of Figure 6.13 displays the cross-validation error, while the right-hand panel displays the coefficient estimates. The vertical dashed lines indicate the point at which the cross-validation error is smallest. The two colored lines in the right-hand panel of Figure 6.13 represent the two predictors that are related to the response, while the grey lines represent the unrelated predictors; these are often referred to as *signal* and *noise* variables, respectively. Not only has the lasso correctly given much larger coefficient estimates to the two signal predictors, but also the minimum cross-validation error corresponds to a set of coefficient estimates for which only the signal variables are non-zero. Hence cross-validation together with the lasso has correctly identified the two signal variables in the model, even though this is a challenging setting, with  $p = 45$  variables and only  $n = 50$

signal