Machine Learning for OR & FE

Regression II: Regularization and Shrinkage Methods

Martin Haugh

Department of Industrial Engineering and Operations Research
Columbia University
Email: martin.b.haugh@gmail.com

Some of the figures in this presentation are taken from "An Introduction to Statistical Learning, with applications in R" (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani (JWHT).

References: Sections 6.1, 6.2 and 6.4 of JWHT

Outline

Linear Regression Revisited

Subset Selection

Shrinkage Methods
- Ridge Regression
- The Lasso
- Ridge Regression Versus Lasso
- Other Shrinkage Methods

Issues in High Dimensions

Linear Regression

Recall our linear regression model:

$$Y = \beta_0 + \sum_{i=1}^{p} \beta_i X_i + \epsilon.$$

Have seen how to fit this model via least squares but often preferable to use other solutions techniques as they often result in:

1. Superior prediction accuracy, especially when p is close to N
   - in fact if $p > N$ then least squares does not yield a unique $\hat{\beta}$
   - superior prediction will result from controlling overfitting and identifying a good bias-variance trade-off.
2. Better interpretability via the exclusion of irrelevant variables.

Will consider the following methods here:

1. Subset selection where only a subset of the independent variables are retained.
2. Shrinkage methods where coefficients are shrunk towards zero
   - typically achieved via regularization.

Cross-validation often used to select the specific model.

Best-Subset Regression

Best subset regression proceeds according to Algorithm 6.1 from ISLR:

Algorithm 6.1 *Best subset selection*

1. Let $\mathcal{M}_0$ denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.
2. For $k = 1, 2, \ldots p$:
   (a) Fit all $\binom{p}{k}$ models that contain exactly k predictors.
   (b) Pick the best among these $\binom{p}{k}$ models, and call it $\mathcal{M}_k$. Here *best* is defined as having the smallest RSS, or equivalently largest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

Feasible using leaps-and-bounds algorithm for p as large as approx 40.

See Figure 6.1 in ISLR for best-subset regressions in credit example

- best RSS decreases with k so cannot use this to select k
- instead use one of the criteria listed above.

Best-subset regression infeasible for large values of p

- forward- and backward-stepwise selection are tractable alternatives.

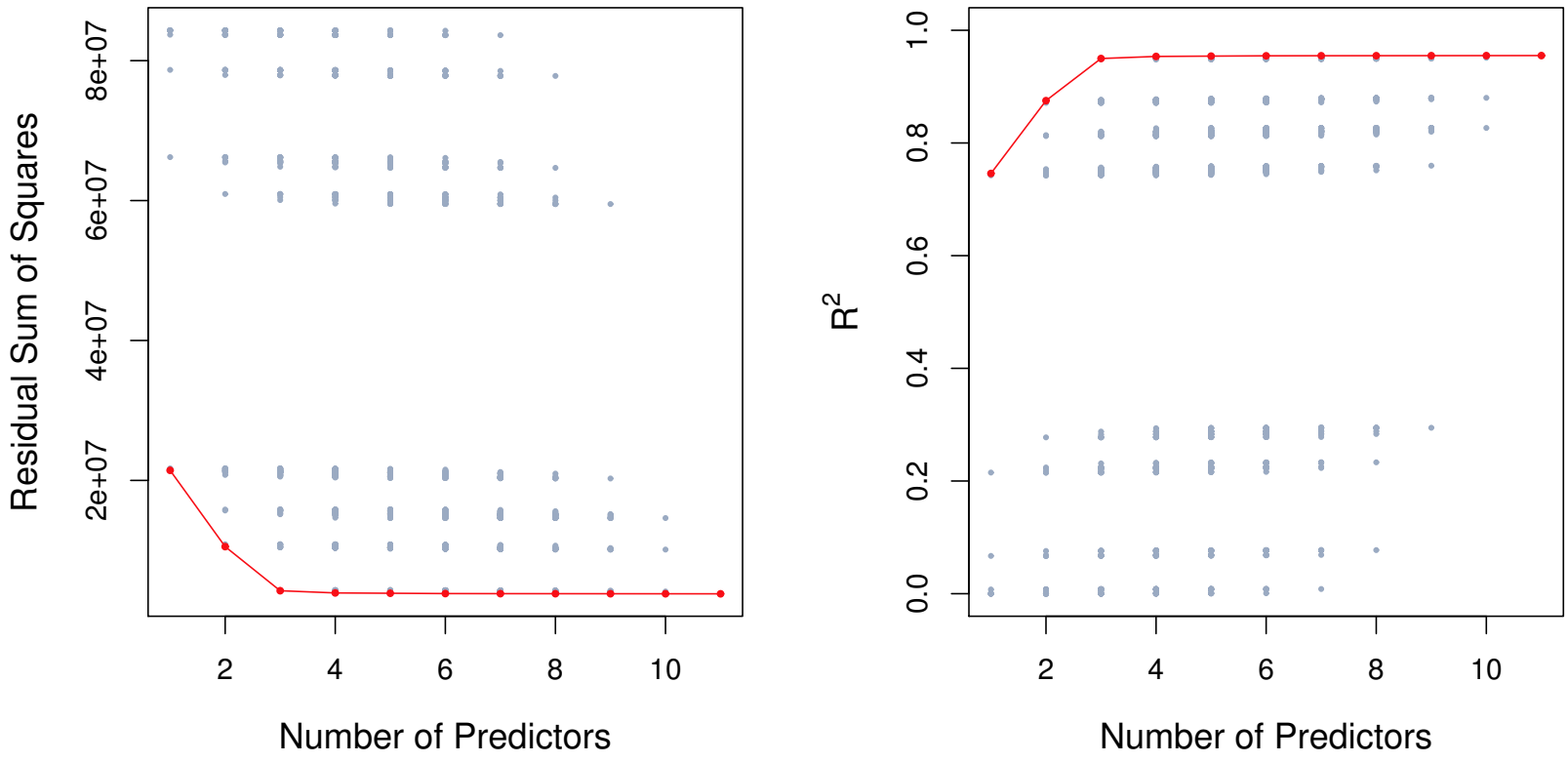

Figure 6.1 from ISLR: For each possible model containing a subset of the ten predictors in the Credit data set, the RSS and R^2 are displayed. The red frontier tracks the best model for a given number of predictors, according to RSS and R^2. Though the data set contains only ten predictors, the x-axis ranges from 1 to 11, since one of the variables is categorical and takes on three values, leading to the creation of two dummy variables.

Forward-Stepwise Selection

Forward stepwise selection is a greedy algorithm that proceeds according to Algorithm 6.2 from ISLR:

Algorithm 6.2 *Forward stepwise selection*

1. Let $\mathcal{M}_0$ denote the *null* model, which contains no predictors.
2. For $k = 0, \ldots, p-1$:
   (a) Consider all $p-k$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor.
   (b) Choose the *best* among these $p-k$ models, and call it $\mathcal{M}_{k+1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

Much faster than best subset selection. Why?

In step 3, why can we not choose the model with the largest R^2?

Backward-Stepwise Selection

Backward stepwise selection is a greedy algorithm that proceeds according to Algorithm 6.3 from ISLR:

Algorithm 6.3 *Backward stepwise selection*

1. Let $\mathcal{M}_p$ denote the *full* model, which contains all p predictors.
2. For $k = p, p-1, \ldots, 1$:
   (a) Consider all k models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.
   (b) Choose the *best* among these k models, and call it $\mathcal{M}_{k-1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

Also much faster than best subset selection. Why?

Backward stepwise selection begins with the full model and sequentially drops the least-informative predictor

- can only be used if $N > p$. Why?

Subset Selection Methods

Forward-stagewise regression is a more constrained (and slower) version of forward-stepwise regression

- see section 3.3.3 of HTF for details.

There are also hybrid approaches that consider forward and backward moves at each step

- often using the AIC, BIC or adjusted R^2 criterion to make the decision
- traditionally F-statistics were used to make these decisions but they suffer from multiple testing issues
  - an enormous problem throughout science and public policy.

Once model has been chosen it is common to print out a summary of the details of the fitted model including estimated standard errors etc.

Question: What is the problem with this?

– the bootstrap can be useful in addressing these issues.

C_p, AIC, BIC, and Adjusted R^2

Let MSE $:=$ RSS/N denote the fitted model's performance on a given data set.

Then we know (why?) the training set MSE will underestimate the test set MSE.

Would therefore like to adjust the training set MSE to get a better estimate of the test set MSE.

There are several approaches:

1. C_p applies to least-squares models and is given by

$$C_p := \frac{1}{N}(\text{RSS} + 2p\hat{\sigma}^2)$$

   Can be shown that if $\hat{\sigma}^2$ is an unbiased estimate of σ^2 (and the model is correct!) then C_p is an unbiased estimate of the test MSE.

2. The AIC (Akaike information criterion) applies to a broader range of models that are fit via maximum likelihood estimation (MLE). In the case of the linear regression model with Gaussian errors it is given by

$$AIC := \frac{1}{N\hat{\sigma}^2}(\text{RSS} + 2p\hat{\sigma}^2)$$

   For least squares models C_p and AIC are equivalent.

C_p, AIC, BIC, and Adjusted R^2

3. BIC (Bayesian Information criterion) is derived from a Bayesian viewpoint but results in a similar expression (for least squares models):

$$BIC := \frac{1}{N}(\text{RSS} + \log(N)p\hat{\sigma}^2),$$

Since $\log(N) > 2$ for $N > 7$, BIC penalizes models with many parameters more than C_p does and so its use results in the selection of smaller models.

Note: Formulas for C_p, AIC and BIC tend to vary but they should all coincide up to irrelevant constants.

4. The adjusted R^2 statistics doesn't have the theoretical justification (when $N \to \infty$) of other criteria but is quite popular as it is intuitive. It satisfies

$$\text{Adjusted } R^2 := 1 - \frac{\text{RSS}/(N-p-1))}{\text{TSS}/(N-1)}$$

Note that large values of adjusted-R^2 are "good".

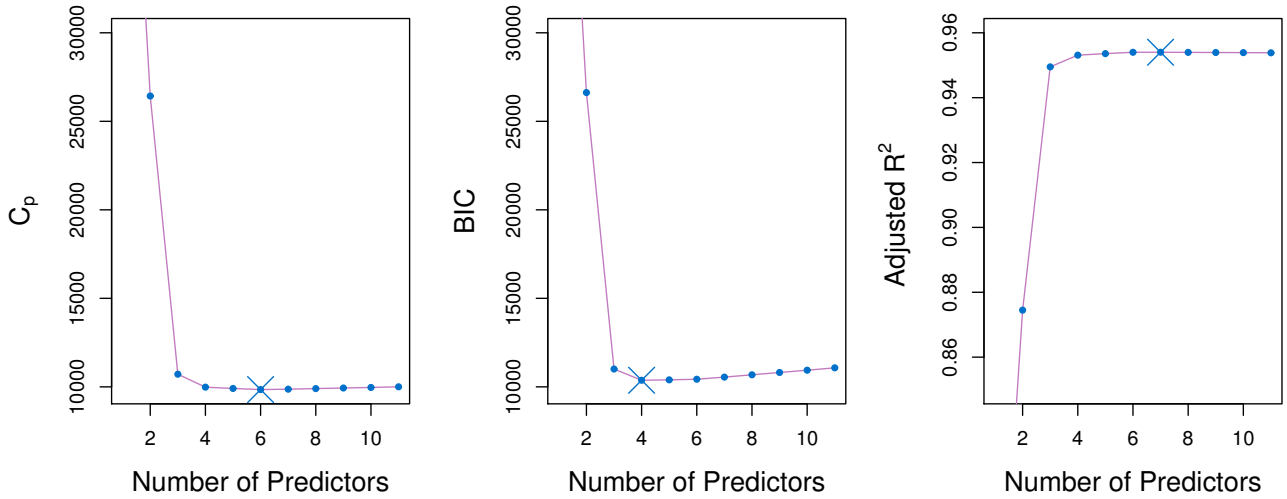

Figure 6.2 from ISLR: C_p, BIC, and adjusted R^2 are shown for the best models of each size for the Credit data set (the lower frontier in Figure 6.1). C_p and BIC are estimates of test MSE. In the middle plot we see that the BIC estimate of test error shows an increase after four variables are selected. The other two plots are rather flat after four variables are included.

While C_p, AIC, BIC, and adjusted R^2 are quite popular they can be hard to apply to more general problems.

This is not true of cross-validation which provides direct estimates of the test MSE and is easy to apply in general.

Given speed of modern computers cross-validation now appears to be the method of choice.

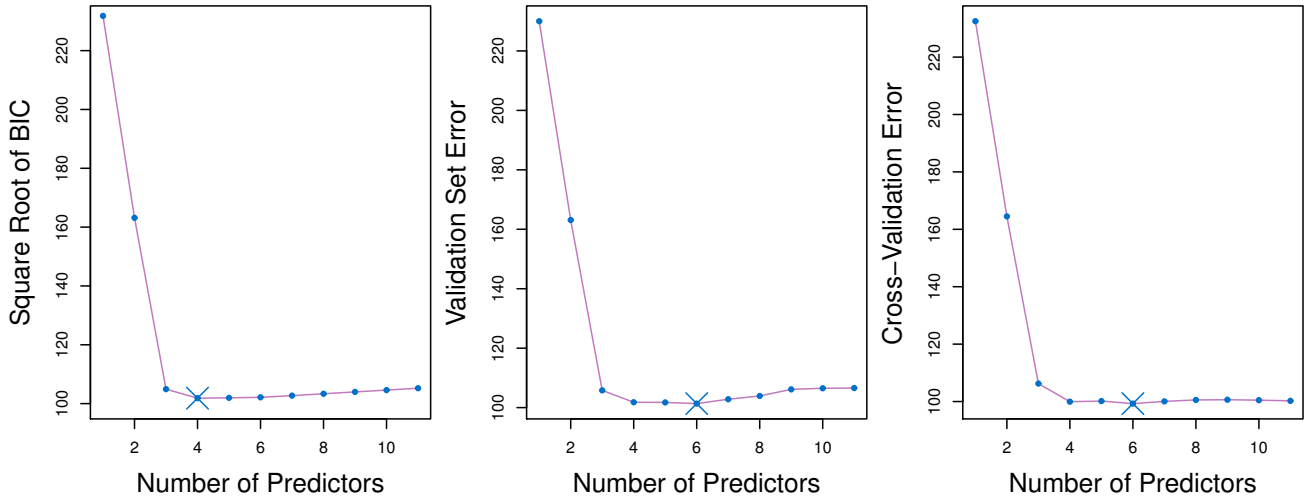

Figure 6.3 from ISLR: For the Credit data set, three quantities are displayed for the best model containing d predictors, for d ranging from 1 to 11. The overall best model, based on each of these quantities, is shown as a blue cross. Left: Square root of BIC. Center: Validation set errors. Right: Cross-validation errors.

Figure 6.3 displays the BIC, validation set errors and cross-validation error on the credit data set.

Validation errors calculated by randomly selecting 3/4 of the observations as the training set, and the remainder as the validation set.

Cross-validation errors were computed using k $=$ 10 folds.

All 3 approaches suggest using a model with just 3 predictors is sufficient. Why?

Shrinkage Methods

Will focus mainly on two shrinkage methods:

1. Ridge regression where we solve:

$$\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \left\| \mathbf{y} - \mathbf{X}\boldsymbol{\beta} \right\|^2 + \lambda \cdot \frac{1}{2} \left\| \boldsymbol{\beta} \right\|_2^2 \right\}.$$

2. The *Least Absolute Shrinkage and Selection Operator* or Lasso solves

$$\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \left\| \mathbf{y} - \mathbf{X}\boldsymbol{\beta} \right\|^2 + \lambda \left\| \boldsymbol{\beta} \right\|_1 \right\} \quad \left\| \boldsymbol{\beta} \right\|_1 = \sum_{j=1}^{n} |\beta_j|$$

   As λ increases, coefficients will abruptly drop to zero.

Question: How should we choose λ?

Note: Shrinkage methods can also be applied to classification problems!

Ridge Regression

Ridge regression solves

$$\hat{\beta}^{\mathsf{R}} = \underset{\beta}{\operatorname{argmin}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{X}\beta\|^2 + \frac{\lambda}{2} \sum_{j=1}^{p} \beta_j^2 \right\}$$

- shrinks regression coefficients towards 0 by imposing a penalty on their size
- λ is a complexity parameter that controls the amount of shrinkage.

An equivalent formulation is

$$\hat{\beta}^{\mathsf{R}} = \underset{\beta}{\operatorname{argmin}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{X}\beta\|^2 \right\} \quad (1)$$

$$\text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 \leq s$$

It is standard (why?) to scale and standardize inputs before applying ridge regression.

Ridge Regression

Note β_0 is generally not shrunk so that procedure does not depend on origin chosen for Y.

To handle this and use matrix notation we can split estimation into two steps:

1. Set $\hat{\beta}_0 = \bar{y} = \frac{\sum_{i=1}^{N} y_i}{N}$
2. Center the inputs so that $x_{ij} \to x_{ij} - \bar{x}_j$.
   Now estimate $\beta_1, \ldots, \beta_p$ using ridge regression without intercept and using the centered x_{ij}'s.

Dropping β_0 from $\boldsymbol{\beta}$, the ridge regression of step 2 therefore solves

$$\hat{\boldsymbol{\beta}}^{\mathsf{R}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \frac{\lambda}{2} \boldsymbol{\beta}^\top \boldsymbol{\beta} \right\}$$

which has solution

$$\hat{\boldsymbol{\beta}}^{\mathsf{R}} = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{y}. \tag{2}$$

Ridge Regression

Note that $\hat{\boldsymbol{\beta}}^{\mathsf{R}}$ is obtained as the solution of a least squares problem except that a positive term, i.e. λ, has been added to the diagonal of $\mathbf{X}^\top\mathbf{X}$

- this makes the problem non-singular, even if $\mathbf{X}^\top\mathbf{X}$ does not have full rank
- this was the main motivation for ridge regression when first introduced.

Ridge regression estimates can easily be obtained in a Bayesian setting

- prior distribution on each β_i is independent normal $\mathsf{N}(0, \tau^2)$
- then with $\lambda := \sigma^2/\tau^2$, obtain $\hat{\boldsymbol{\beta}}^{\mathsf{R}}$ as mean of posterior distribution.

Figure 6.4 from ISLR displays $\hat{\boldsymbol{\beta}}^{\mathsf{R}}$ for various values of λ and $||\hat{\boldsymbol{\beta}}^{\mathsf{R}}_\lambda||_2/||\hat{\boldsymbol{\beta}}||_2$

- can interpret $||\hat{\boldsymbol{\beta}}^{\mathsf{R}}_\lambda||_2/||\hat{\boldsymbol{\beta}}||_2$ as a measure of the total shrinkage achieved
- note that we recover the least squares solution as $\lambda \to 0$.

Ridge Regression on the Credit Data Set

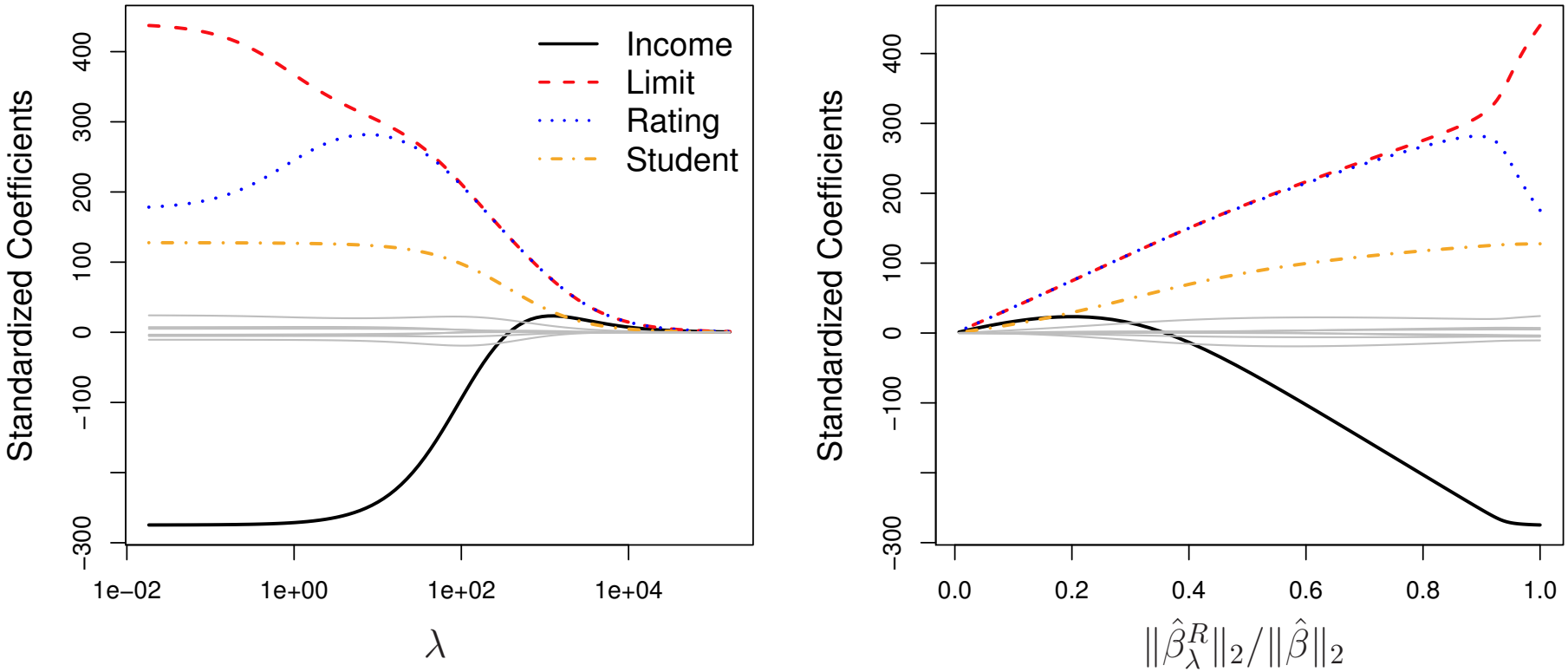

Figure 6.4 from ISLR: The standardized ridge regression coefficients are displayed for the Credit data set, as a function of λ and $||\hat{\boldsymbol{\beta}}^{\mathrm{R}}_\lambda||_2/||\hat{\boldsymbol{\beta}}_\lambda||_2$.

Note that as λ increases coefficients are shrunk towards zero.

Also note that coefficients are generally non-zero for any value of λ
- so ridge regression does not result in sparse models.

Selecting λ Via Cross-Validation

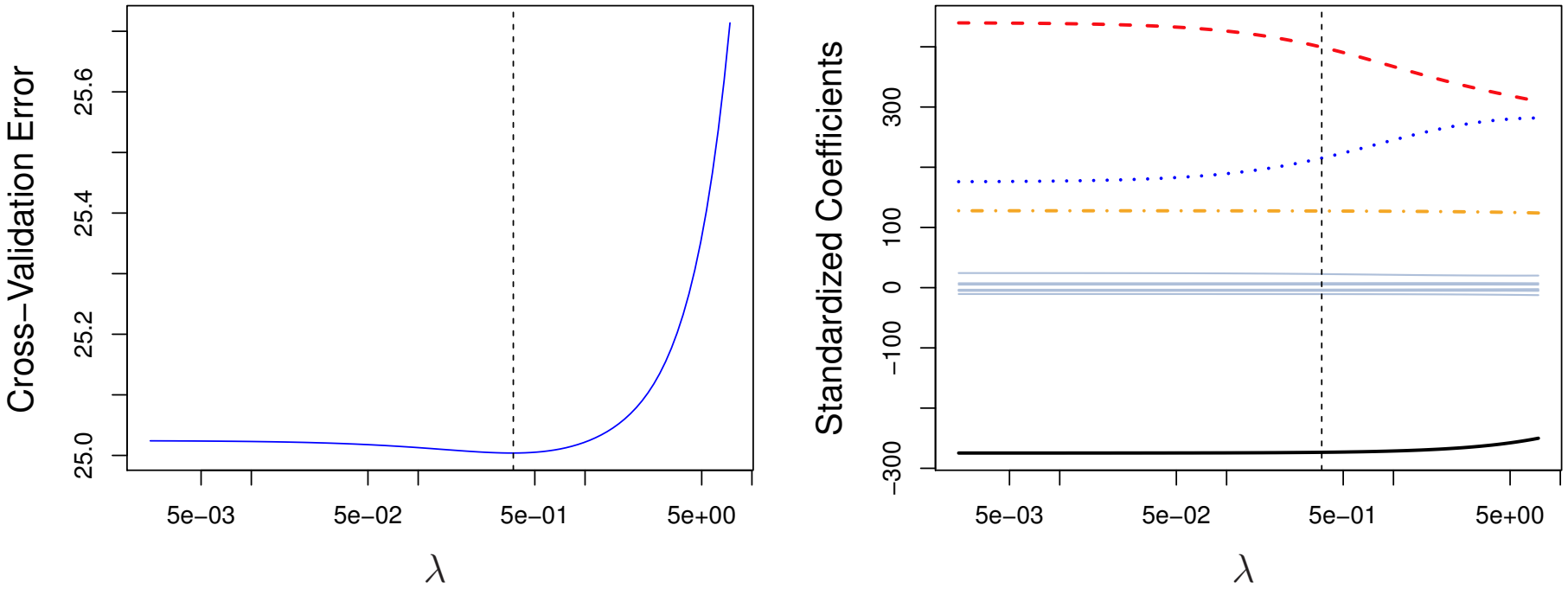

Figure 6.12 from ISLR: Cross-validation errors that result from applying ridge regression to the Credit data set with various value of λ. Right: The coefficient estimates as a function of λ. The vertical dashed lines indicate the value of λ selected by cross-validation.

Using cross-validation to select λ for the Credit data set results in only a modest amount of shrinkage.

And the cv error is relatively insensitive to choice of λ here

- so little improvement over least squares solution.

Why Does Ridge Regression Improve Over Least Squares?

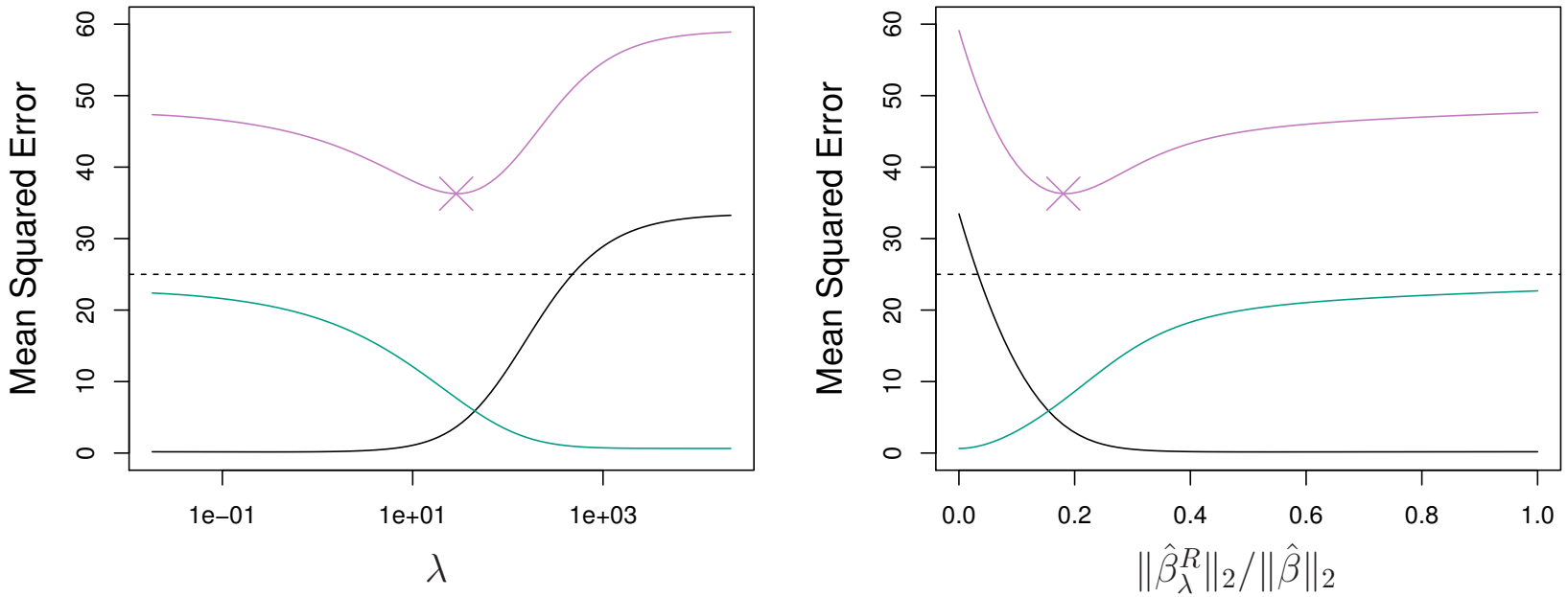

Figure 6.5 from ISLR: Squared bias (black), variance (green), and test mean squared error (purple) for the ridge regression predictions on a simulated data set, as a function of λ and $||\hat{\boldsymbol{\beta}}_\lambda^R||_2/||\hat{\boldsymbol{\beta}}_\lambda||_2$. The horizontal dashed lines indicate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

Ridge regression (and Lasso) often (significantly) outperform least-squares because it is capable (through selection of λ) of trading off a small increase in bias for a potentially much larger decrease in variance.

The Lasso

Recall that the Lasso solves

$$\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \left\| \mathbf{y} - \mathbf{X}\boldsymbol{\beta} \right\|^2 + \lambda \left\| \boldsymbol{\beta} \right\|_1 \right\}$$

where $\left\| \boldsymbol{\beta} \right\|_1 := \sum_{j=1}^{n} |\beta_j|$.

Penalizing the 1-norm ensures that coefficients will abruptly drop to zero as λ increases – results in superior interpretability.

The Lasso can also be formulated by constraining $\left\| \boldsymbol{\beta} \right\|_1$:

$$\hat{\boldsymbol{\beta}}^{\mathsf{L}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \frac{1}{2} \left\| \mathbf{y} - \mathbf{X}\boldsymbol{\beta} \right\|^2 \right\} \tag{3}$$

$$\text{subject to} \quad \sum_{j=1}^{p} |\beta_j| \leq s$$

Unlike ridge regression, a closed-form solution is not available for the Lasso

- but it can be formulated as a convex quadratic optimization problem and is therefore easy to solve numerically.

Lasso on the Credit Data Set

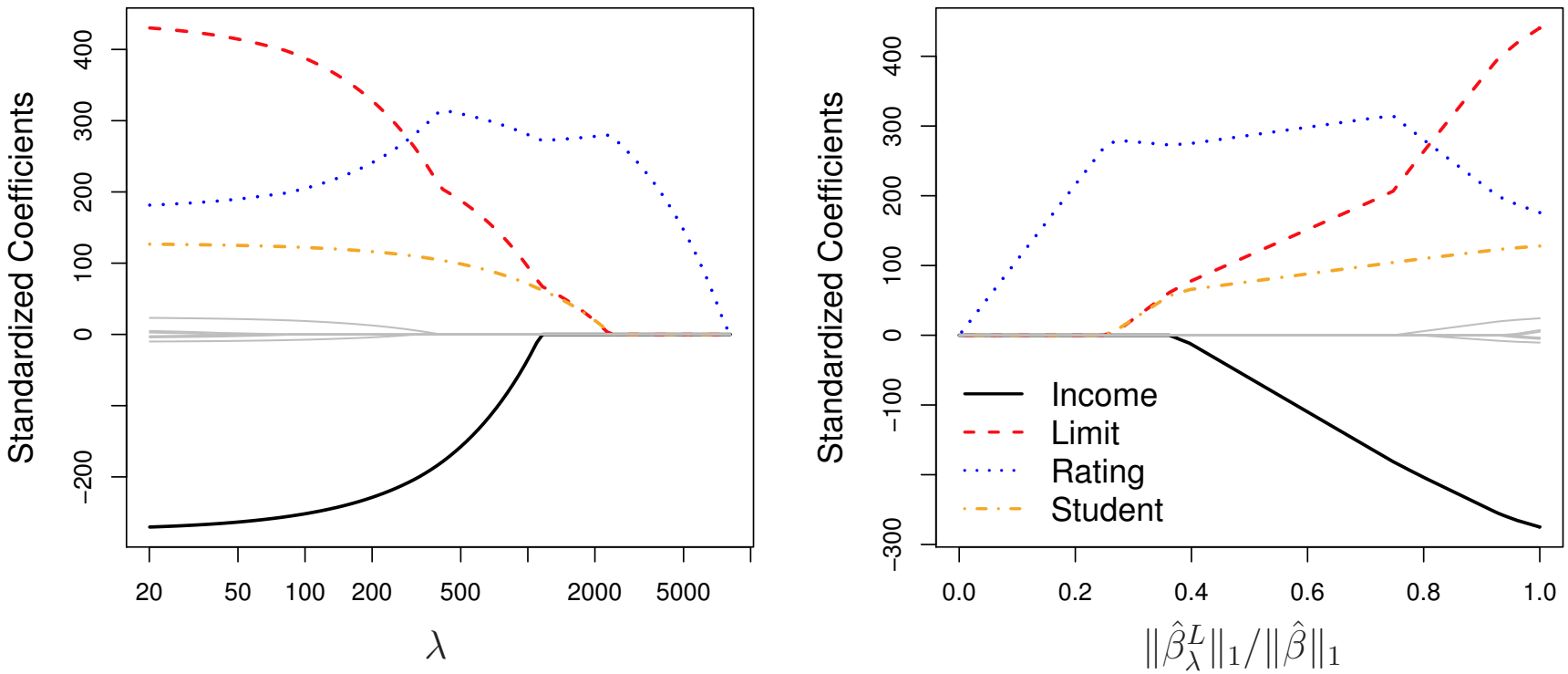

Figure 6.6 from ISLR: The standardized lasso coefficients on the Credit data set are shown as a function of λ and $||\hat{\beta}_\lambda^L||_1/||\hat{\beta}_\lambda||_1$.

Note how coefficients abruptly drop to 0 as λ increases in Figure 6.6

- contrast this with ridge regression!

Lasso results in sparse models then and can be viewed as a method for subset selection.

A Simulated Data Set

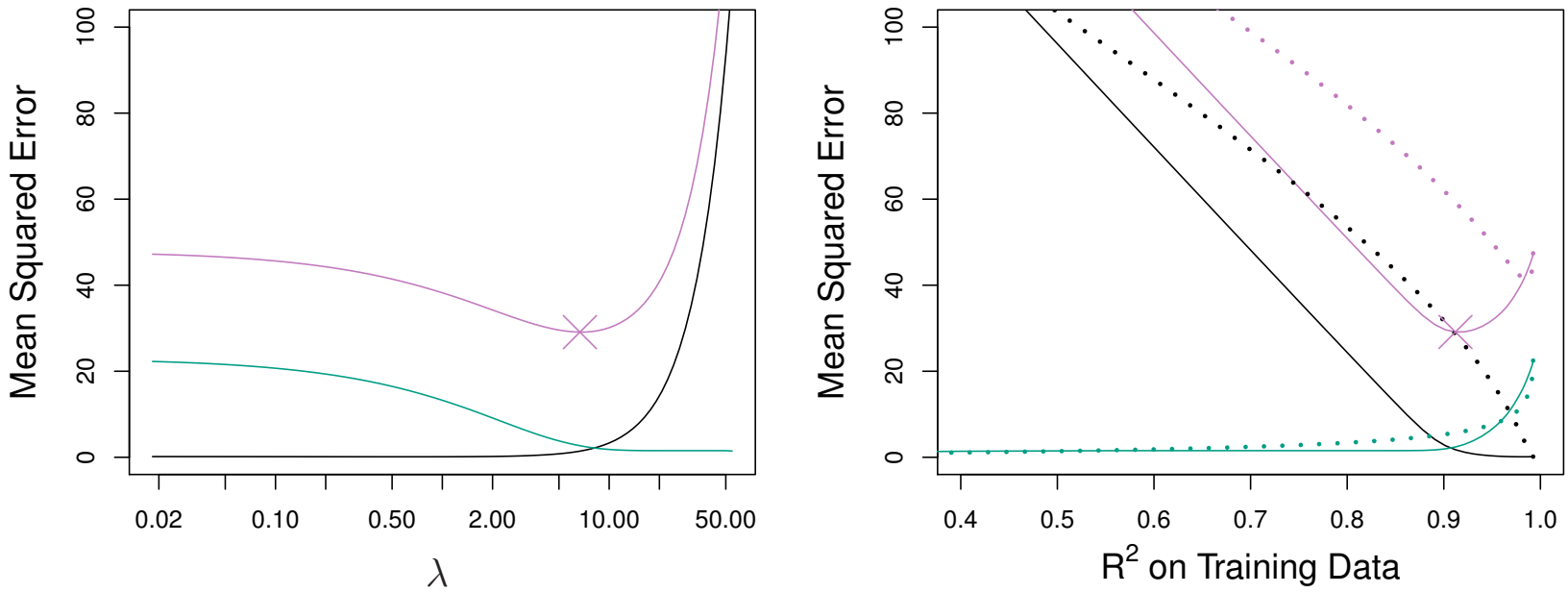

Figure 6.9 from ISLR: Left: Plots of squared bias (black), variance (green), and test MSE (purple) for the lasso. The simulated data is similar to that in Figure 6.8, except that now only two predictors are related to the response. Right: Comparison of squared bias, variance and test MSE between lasso (solid) and ridge (dashed). 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.

Figure 6.9 displays results from a simulated data set with $p = 45$ predictors – but the response Y is a function of only 2 of them!

Selecting λ Via Cross-Validation

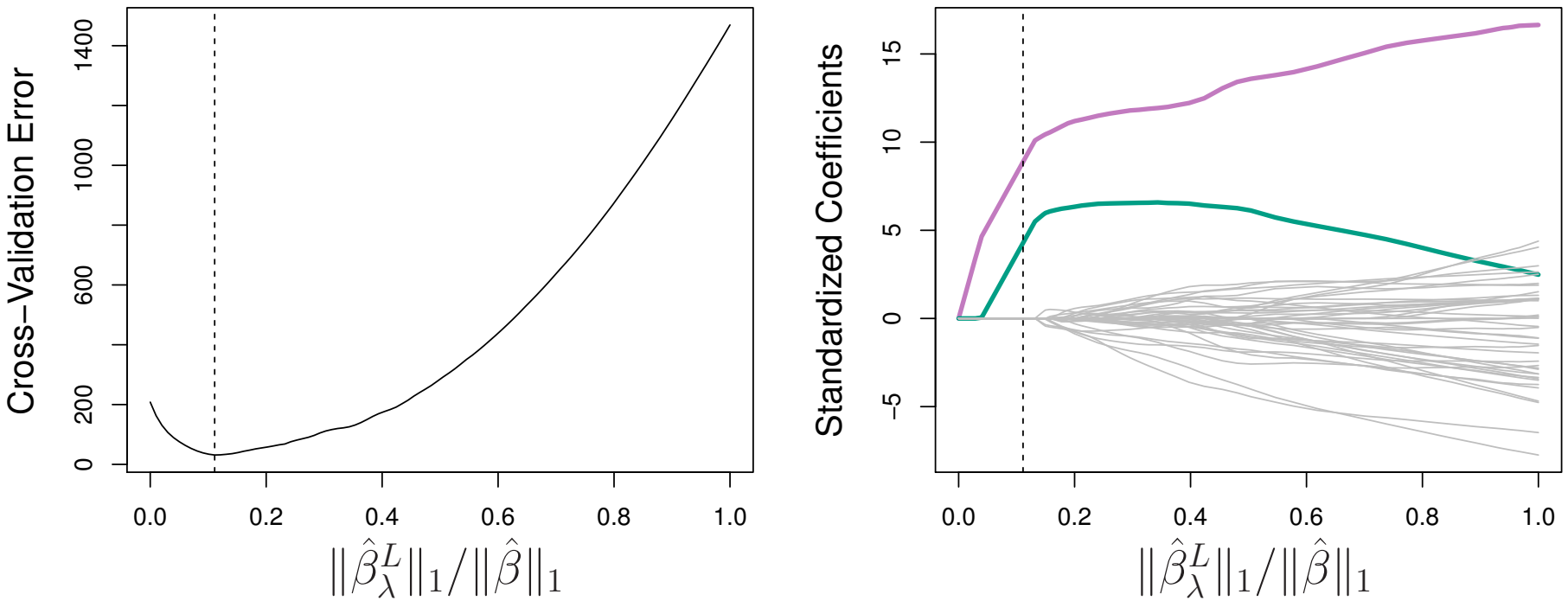

Figure 6.13 from ISLR: Left: Ten-fold cross-validation MSE for the lasso, applied to the sparse simulated data set from Figure 6.9. Right: The corresponding lasso coefficient estimates are displayed. The vertical dashed lines indicate the lasso fit for which the cross-validation error is smallest.

Note how the optimal λ (chosen via cross-validation) correctly identifies the model with the 2 predictors

- contrast with least squares solution at far right of right-hand figure!

Lasso Versus Ridge Regression

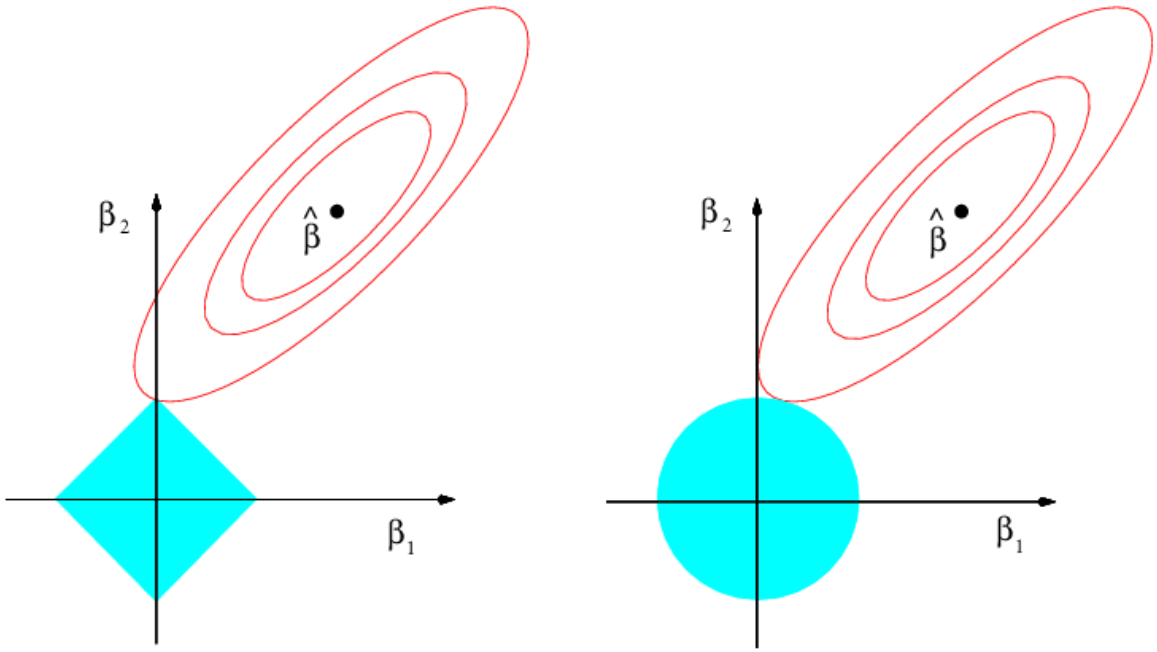

Figure 6.7 from ISLR: Contours of the error and constraint functions for the lasso (left) and ridge regression (right). The solid blue areas are the constraint regions, $|\beta_1| + |\beta_2| \leq s$ and $\beta_1^2 + \beta_2^2 \leq s$, while the red ellipses are the contours of the RSS.

Contours of the error and constraint functions of the formulations in (1) and (3) are displayed in Figure 6.7.

This perspective makes it clear why Lasso results in a sparse solution whereas ridge regression does not.

Ridge Regression Versus Lasso

The following e.g. (taken from ISLR) provides further intuition for why Lasso results in sparse solutions and ridge regression does not. We assume:

- $N = p$.
- $\mathbf{X}$ is a diagonal matrix with 1's on the diagonal.
- There is no intercept term.

Least squares then solves $\min_{\beta_1,\ldots,\beta_p} \sum_{j=1}^{N} (y_j - \beta_j)^2$

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

Ridge regression solves $\min_{\beta_1,\ldots,\beta_p} \sum_{j=1}^{N} (y_j - \beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$

Can check solution is $\hat{\beta}_j^R = y_j/(1+\lambda)$.

Lasso solves $\min_{\beta_1,\ldots,\beta_p} \sum_{j=1}^{N} (y_j - \beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$

Can check solution is

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

Other Shrinkage Methods

Group Lasso:

$$\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \left\| \mathbf{y} - \mathbf{X}\boldsymbol{\beta} \right\|^2 + \lambda \sum_{k=1}^{m} \left\| \boldsymbol{\beta}_k \right\|_2 \right\}$$

where $\boldsymbol{\beta}_k$ are non-overlapping sub-vectors of $(\beta_1, \ldots, \beta_p)^\top$

- Induces all the coefficients in the sub-vector to go to zero
- Useful when there are dummy variables in the regression.

Composite norm methods:

$$\min_{\boldsymbol{\beta}} \left\{ \left\| \mathbf{y} - \mathbf{X}\boldsymbol{\beta} \right\| + \lambda \sum_{k=1}^{m} \left\| \boldsymbol{\beta}_k \right\|_2 \right\}$$

- Useful when we want to force $\mathbf{X}\boldsymbol{\beta} = \mathbf{y}$.

Elastic nets:

$$\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} \left\| \mathbf{y} - \mathbf{X}\boldsymbol{\beta} \right\|^2 + \lambda \big((1-\alpha) \frac{1}{2} \left\| \boldsymbol{\beta} \right\|_2^2 + \alpha \left\| \boldsymbol{\beta} \right\|_1 \big) \right\}$$

High Dimensional Problems

Traditionally problems in statistics were low-dimensional with $p < N$ and often $p << N$.

But many modern setting have $p > N$. For example:

1. Classical statistics might attempt to predict blood pressure as a function of age gender and body-mass-index (BMI). Modern methods might also use measurements for approx 500k single nucleotide polymorphisms (SNPs).
2. Online advertisers may want to predict the purchasing behavior of someone using a search engine. Dummy variables for each of p search terms might be included as predictors with $p_i = 1$ if the i^{th} term was previously searched by the user and $p_i = 0$ otherwise.
3. Speech recognition problems where we have speech samples for N speakers. To represent a speech sample as a numeric vector we require very large p.

Need to be very careful in these high-dimensional settings where (unique) least squares solutions do not even exist.

Even if p is smaller than but still close to N then similar problems still arise.

Similar observations hold true for classification problems that use classical approaches such as LDA, QDA, logistic regression etc.

Issues in High Dimensions

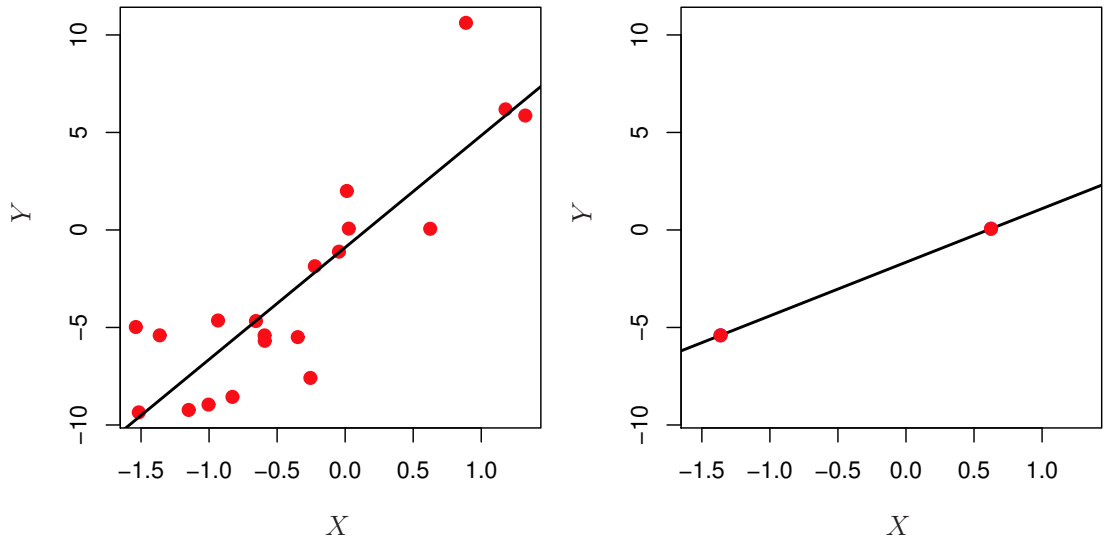

Figure 6.22 from ISLR: Left: Least squares regression in the low-dimensional setting. Right: Least squares regression with $n = 2$ observations and two parameters to be estimated (an intercept and a coefficient).

Problem in Fig. 6.22 is low dimensional but demonstrates what can go wrong when we have too little data relative to problem dimension

- this certainly occurs when $p \approx N$
- saw similar issues with the case-study in Regression I slides.

When $p \geq N$ least squares can fit the data perfectly and so R^2 will equal 1

- but likely that massive over-fitting is taking place.

Issues in High Dimensions

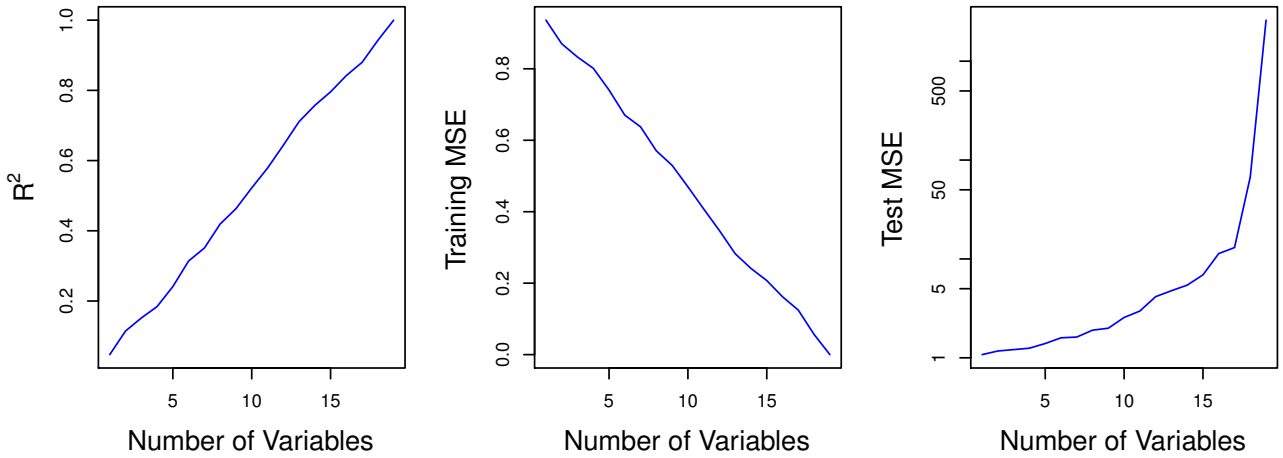

Figure 6.23 from ISLR: On a simulated example with $n = 20$ training observations, features that are completely unrelated to the outcome are added to the model. Left: The R^2 increases to 1 as more features are included. Center: The training set MSE decreases to 0 as more features are included. Right: The test set MSE increases as more features are included.

Note that in Figure 6.23 the features are completely unrelated to the response!

Estimating test error is therefore particularly vital in these settings – but C_p, AIC and BIC are not suitable due to difficulty in estimating σ^2.

The solution is to restrict the choice of models which is exactly what subset selection, ridge regression, lasso etc. do.

Issues in High Dimensions

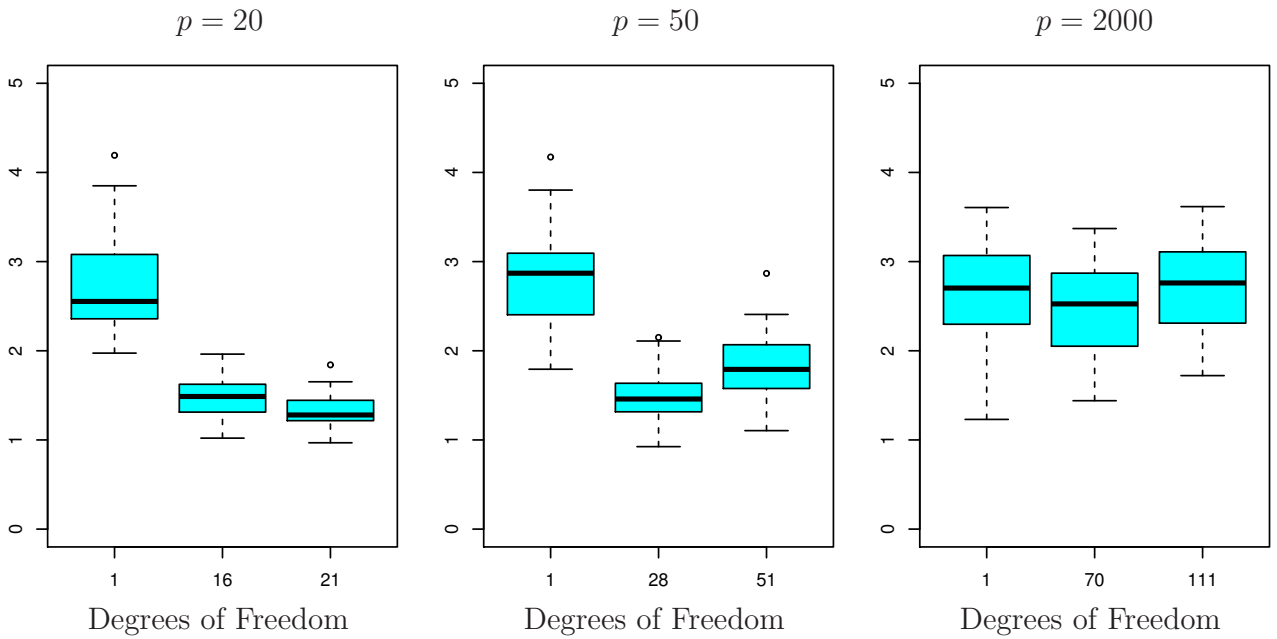

Figure 6.24 from ISLR: The lasso was performed with $n = 100$ observations and three values of p, the number of features. Of the p features, 20 were associated with the response. The boxplots show the test MSEs that result using three different values of the tuning parameter λ in (6.7). For ease of interpretation, rather than reporting λ, the degrees of freedom are reported; for the lasso this turns out to be simply the number of estimated non-zero coefficients. When $p = 20$, the lowest test MSE was obtained with the smallest amount of regularization. When $p = 50$, the lowest test MSE was achieved when there is a substantial amount of regularization. When $p = 2,000$ the lasso performed poorly regardless of the amount of regularization, due to the fact that only 20 of the 2,000 features truly are associated with the outcome.

Issues in High Dimensions

Note results in Figure 6.24 where only 20 features were relevant.

Degrees-of-freedom, $\text{df}(\lambda)$, is reported instead of λ

- $\text{df}(\lambda) =$ number of non-zero coefficient estimates in the lasso solution
- much easier to interpret!

When $p = 20$ or $p = 50$ we see the importance of choosing a good value of λ.

But we also see that lasso performed poorly when $p = 2000$

- because test error tends to increase with p unless the new features are actually informative
- note the implications of this observation – there is a cost to be paid for blindly adding new features to a model even when regularization is employed!

Multi-collinearity is clearly present in high-dimensional problems – therefore cannot hope to identify the very best predictors

- instead hope to identify good predictors.

Note that linear models – which we have been considering – are generally popular for high dimensional problems. Why?