Performance Estimation and Regularization

Kasthuri Kannan, PhD.

Machine Learning, Spring 2018

Bias-Variance Tradeoff

- Fundamental to machine learning approaches

$$E\left(y_0 - \hat{f}(x_0)\right)^2 = \mathrm{Var}(\hat{f}(x_0)) + [\mathrm{Bias}(\hat{f}(x_0))]^2 + \mathrm{Var}(\epsilon).$$

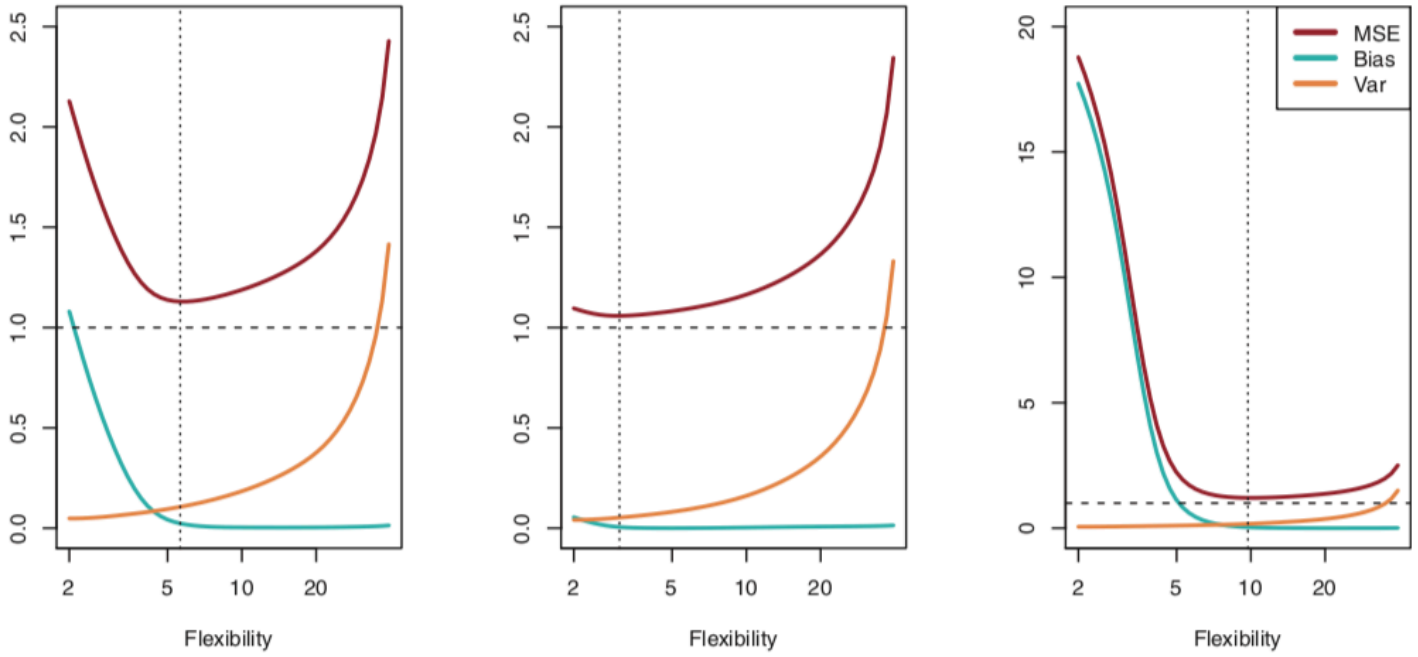
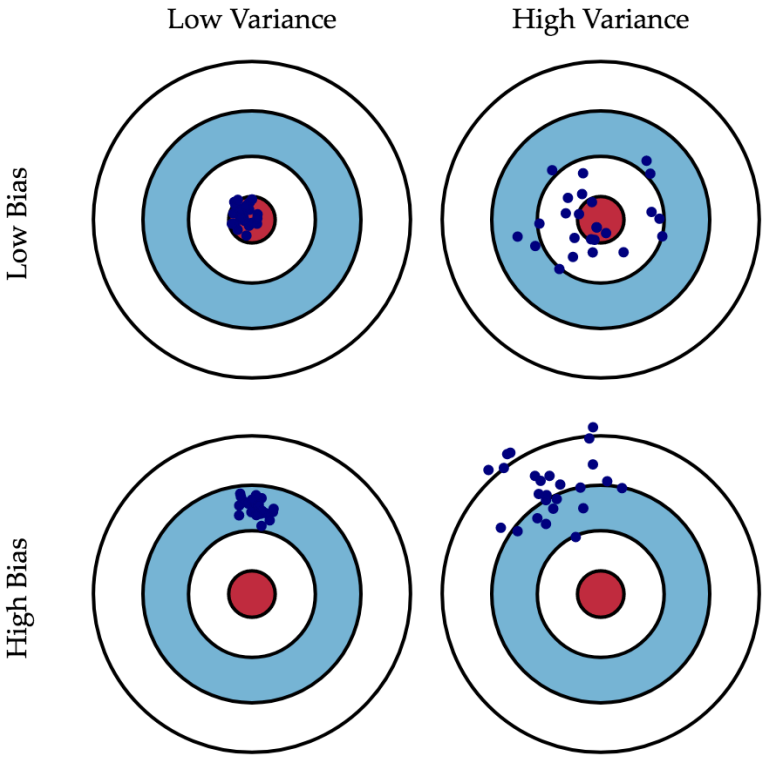


FIGURE 2.12. *Squared bias (blue curve), variance (orange curve), Var(ϵ) (dashed line), and test MSE (red curve) for the three data sets in Figures 2.9–2.11. The vertical dotted line indicates the flexibility level corresponding to the smallest test MSE.*

Bias-Variance Tradeoff

- *Error due to Bias*: The error due to bias is taken as the difference between the expected (or average) prediction of our model and the correct value which we are trying to predict
- *Error due to Variance*: The error due to variance is taken as the variability of a model prediction for a given data point



Performance Estimation

- Model selection and model assessment are two important aspects of machine learning
- Performance estimation is a part of model assessment
- *Resampling methods* are indispensible tools for performance estimation
- Basic Idea
  - Repeatedly draw different samples from the training data, fit a model to each new sample,
  - examine the extent to which the resulting fits differ

Performance Estimation Methods

- Two popular approaches
  - Cross-validation
  - Bootstrapping

- Cross-validation can be used to estimate the test error associated with a given statistical learning method

- Or to select the appropriate level of flexibility

- The bootstrap is commonly used to provide a measure of accuracy of a parameter estimate or of a given statistical learning method

Training and Testing errors

- $\{(x_1,y_1),\ldots,(x_n,y_n)\}$, where $y_1,\ldots,y_n$ are qualitative variables

- Common approach for quantifying the accuracy is the *training error rate* - the proportion of mistakes that are made if we apply our estimate to the training observations:

$$\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i).$$

- The *test error rate* associated with a set of test observations of the form (x_0, y_0) is given by:

$$\text{Ave}\left(I(y_0 \neq \hat{y}_0)\right),$$

where $\hat{y}_0$ is the predicted class label that results from applying the classifier to the test observation with predictor x_0

- A good classifier is one for which the above test error is smallest

Training and Testing Errors - Difference

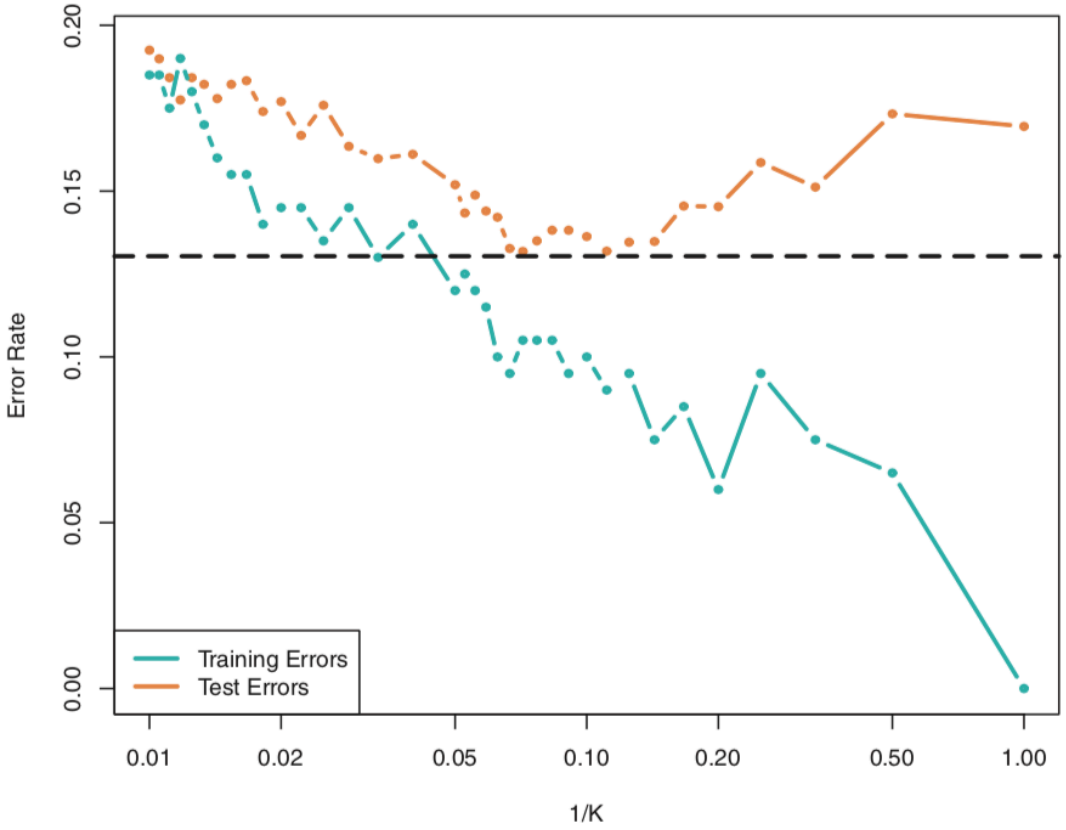


FIGURE 2.17. *The KNN training error rate (blue, 200 observations) and test error rate (orange, 5,000 observations) on the data from Figure 2.13, as the level of flexibility (assessed using $1/K$) increases, or equivalently as the number of neighbors K decreases. The black dashed line indicates the Bayes error rate. The jumpiness of the curves is due to the small size of the training data set.*

Cross-Validation

- Estimate the test error rate by holding out a subset of the training observations from the fitting process, and then applying the statistical learning method to those held out observations

- A very simple strategy

- It involves randomly dividing the available set of observations into two parts, a *training set* and a *validation set* or *hold-out set*

The Validation Set Approach

FIGURE 5.1. *A schematic display of the validation set approach. A set of n observations are randomly split into a training set (shown in blue, containing observations 7, 22, and 13, among others) and a validation set (shown in beige, and containing observation 91, among others). The statistical learning method is fit on the training set, and its performance is evaluated on the validation set.*

Auto Data Set

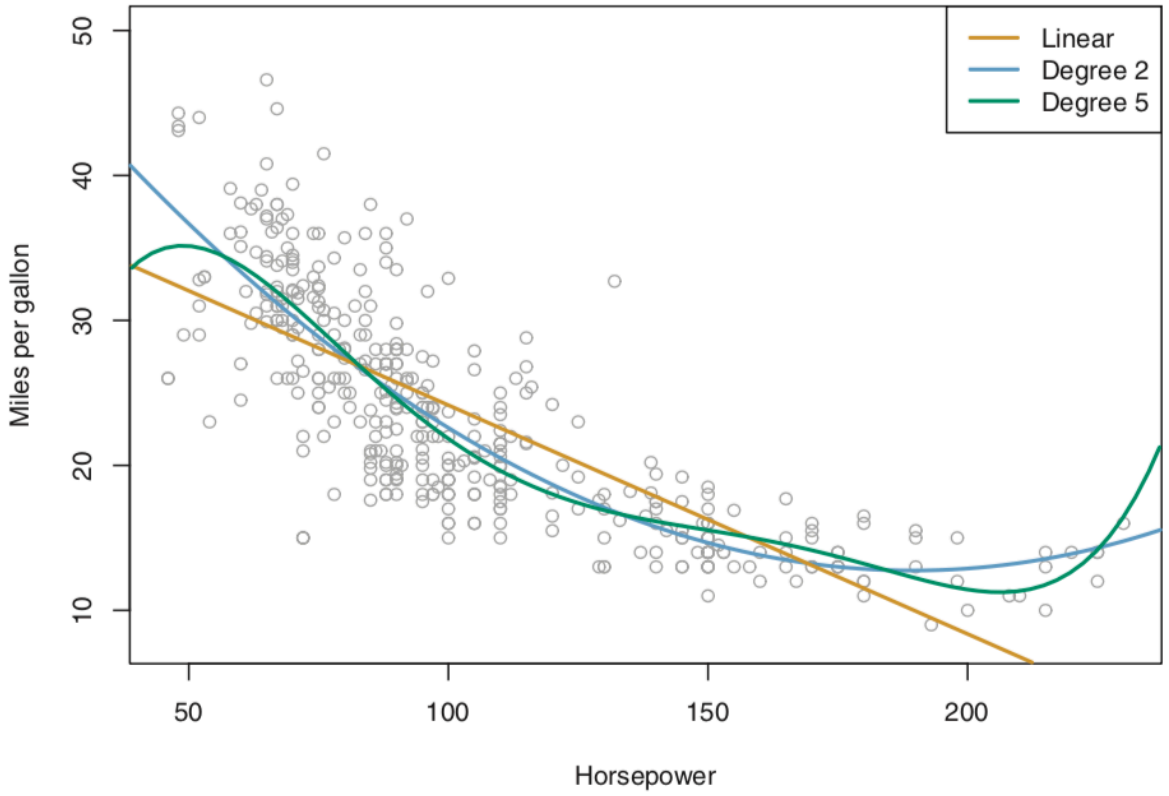


FIGURE 3.8. *The* `Auto` *data set. For a number of cars,* `mpg` *and* `horsepower` *are shown. The linear regression fit is shown in orange. The linear regression fit for a model that includes* $\texttt{horsepower}^2$ *is shown as a blue curve. The linear regression fit for a model that includes all polynomials of* `horsepower` *up to fifth-degree is shown in green.*

$$\texttt{mpg} = \beta_0 + \beta_1 \times \texttt{horsepower} + \beta_2 \times \texttt{horsepower}^2 + \epsilon$$

Auto Data Set – Fit Statistics

	Coefficient	Std. error	t-statistic	p-value
Intercept	56.9001	1.8004	31.6	< 0.0001
horsepower	-0.4662	0.0311	-15.0	< 0.0001
horsepower^2	0.0012	0.0001	10.1	< 0.0001

TABLE 3.10. *For the* Auto *data set, least squares coefficient estimates associated with the regression of* mpg *onto* horsepower *and* horsepower^2.

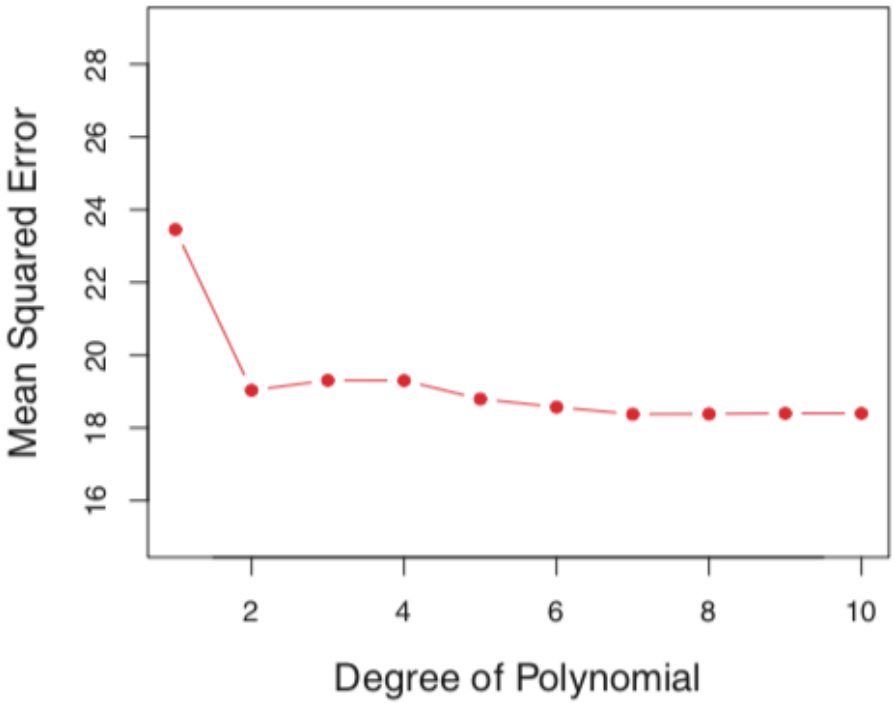
The R^2 of the quadratic fit is 0.688, compared to 0.606 for the linear fit

It is natural to wonder whether a cubic or higher-order fit might provide even better results

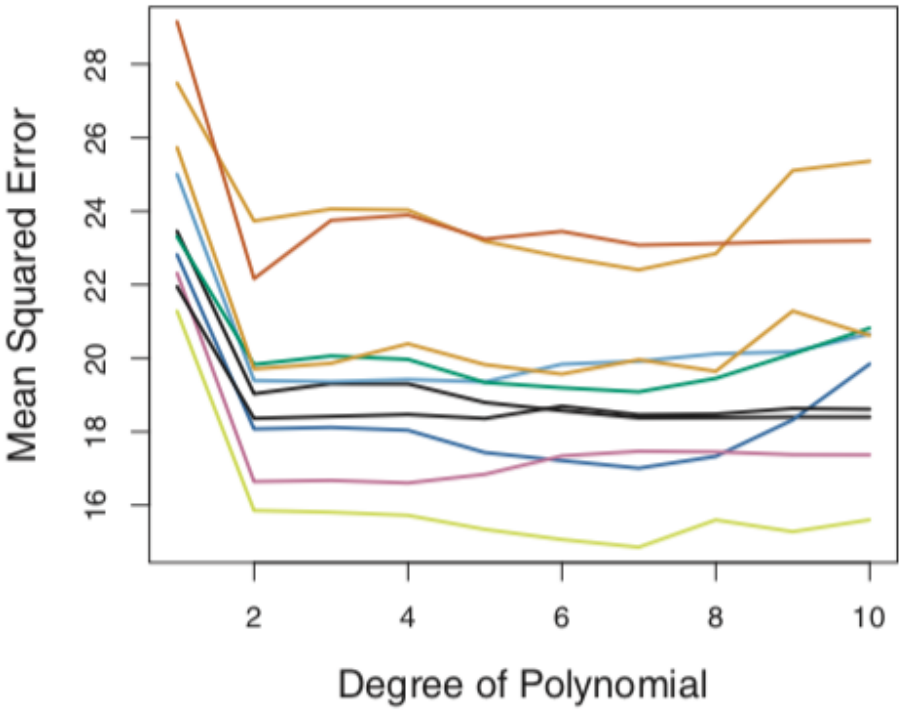
We can answer this question using the validation method

Validation Set Approach on *Auto* Data Set

- Randomly split the 392 observations into two sets,
  - a training set containing 196 of the data points,
  - and a validation set containing the remaining 196 observations



Problems With Validation Set Approach



- Based on the variability among these curves, all that we can conclude with any confidence is that the linear fit is not adequate for this data

Problems With Validation Set Approach

- The validation set approach is conceptually simple and is easy to implement

- Two potential drawbacks:

  - The validation estimate of the test error rate can be highly variable, depending on precisely which observations are included in the training set and which observations are included in the validation set

  - Only a subset of observations are included:
    - Trained on fewer observations implies validation set error rate may overestimate test error rate for the model fit on the entire data set

Leave-One-Out Cross-Validation (LOOCV)

- Attempts to address the above shortcomings
- LOOCV involves splitting the set observations into two parts
  - instead of creating two subsets of comparable size, a single observation (x_1,y_1) is used for the validation set, and the remaining observations $\{(x_2, y_2), \ldots, (x_n, y_n)\}$ make up the training set.
- The statistical learning method is fit on the n − 1 training observations, and a prediction $\hat{y}_1$ is made for the excluded observation, using its value x_1

LOOCV Schema

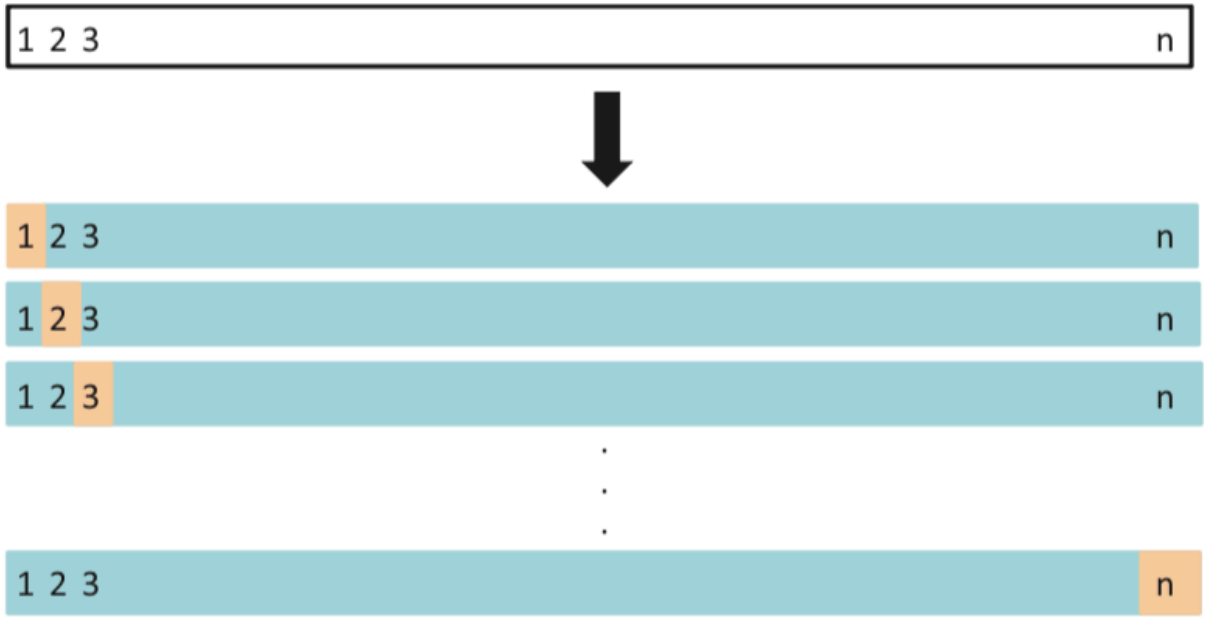


FIGURE 5.3. *A schematic display of LOOCV. A set of n data points is repeatedly split into a training set (shown in blue) containing all but one observation, and a validation set that contains only that observation (shown in beige). The test error is then estimated by averaging the n resulting MSE's. The first training set contains all but observation 1, the second training set contains all but observation 2, and so forth.*

MSE for LOOCV

The LOOCV estimate for the test MSE is the average of *n* test error (MSE) estimates:

$$MSE_1 = (y_1 - \hat{y}_1)^2$$
$$MSE_2 = (y_2 - \hat{y}_2)^2$$
$$\vdots$$
$$MSE_n = (y_n - \hat{y}_n)^2$$

$$LOOCV_{(n)} = \frac{1}{n}\sum_{i=1}^{n} MSE_i$$

Note: Each of these MSE estimates are poor estimates because it is highly variable, since it is based upon a single observation – however the average may not

LOOCV Advantages

- Less bias
  - we repeatedly fit the statistical learning method using training sets that contain $n - 1$ observations, almost as many as are in the entire data set
  - contrast this to the validation set approach, in which the training set is typically around half the size of the original data set
  - consequently, the LOOCV approach tends not to overestimate the test error rate as much as the validation set approach does

LOOCV Advantages

- No randomness
  - performing LOOCV multiple times will always yield the same results: there is no randomness in the training/validation set splits
  - contrast this with other validation approaches

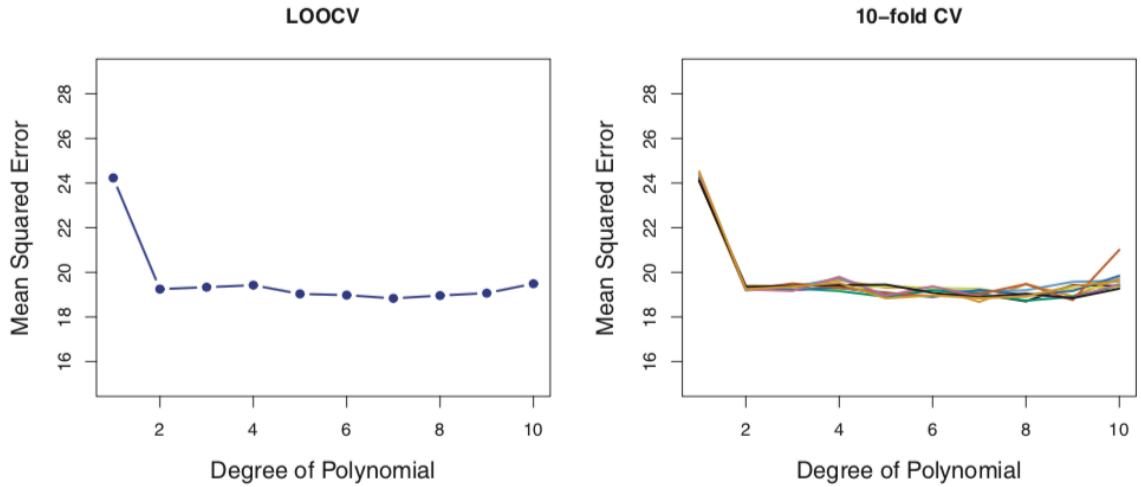


FIGURE 5.4. *Cross-validation was used on the* `Auto` *data set in order to estimate the test error that results from predicting* `mpg` *using polynomial functions of* `horsepower`. Left: *The LOOCV error curve.* Right: 10-*fold CV was run nine separate times, each with a different random split of the data into ten parts. The figure shows the nine slightly different CV error curves.*

k-fold Cross-Validation

- LOOCV requires fitting the statistical learning method n times
- This is computationally expensive
- An alternative to LOOCV is *k-fold* CV
- This approach involves randomly dividing the set of observations into k groups, or folds, of approximately equal size.
- The first fold is treated as a validation set, and the method is fit on the remaining k − 1 folds.

$$CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} MSE_i$$

Training and Test MSE

Training data set - $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$

We obtain the estimate $\hat{f}$

$Training_{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}(x_i)\right)^2$ will be small

We want to know whether $\hat{f}(x_0) \approx y_0$

when (x_0, y_0) is a previously unseen test observation not used to train the statistical learning method.

That is if the $Testing_{MSE} = Ave(\hat{f}(x_0) - y_0)^2$ is small

Training and Test MSE on Simulated Data 1

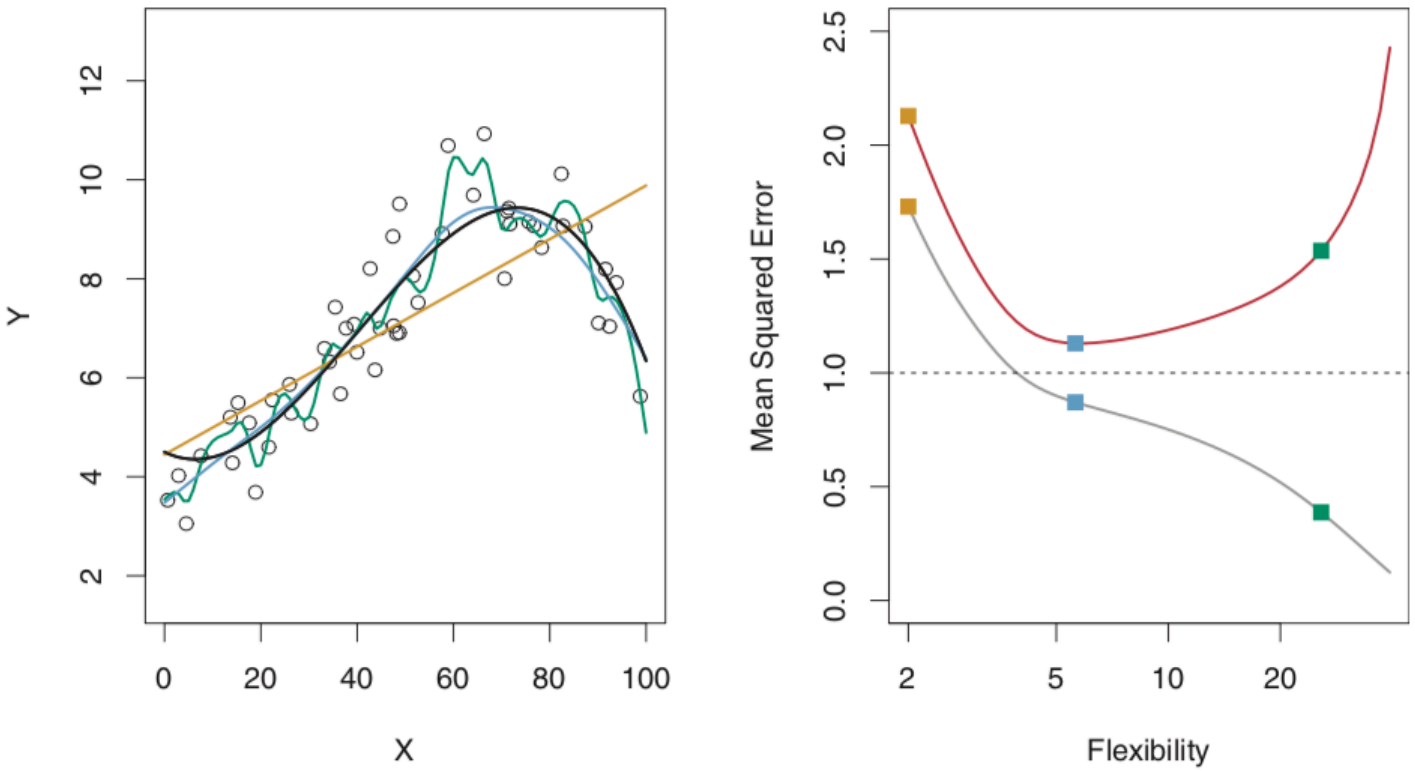


FIGURE 2.9. Left: *Data simulated from f, shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves).* Right: *Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed line). Squares represent the training and test MSEs for the three fits shown in the left-hand panel.*

Training and Test MSE on Simulated Data 2

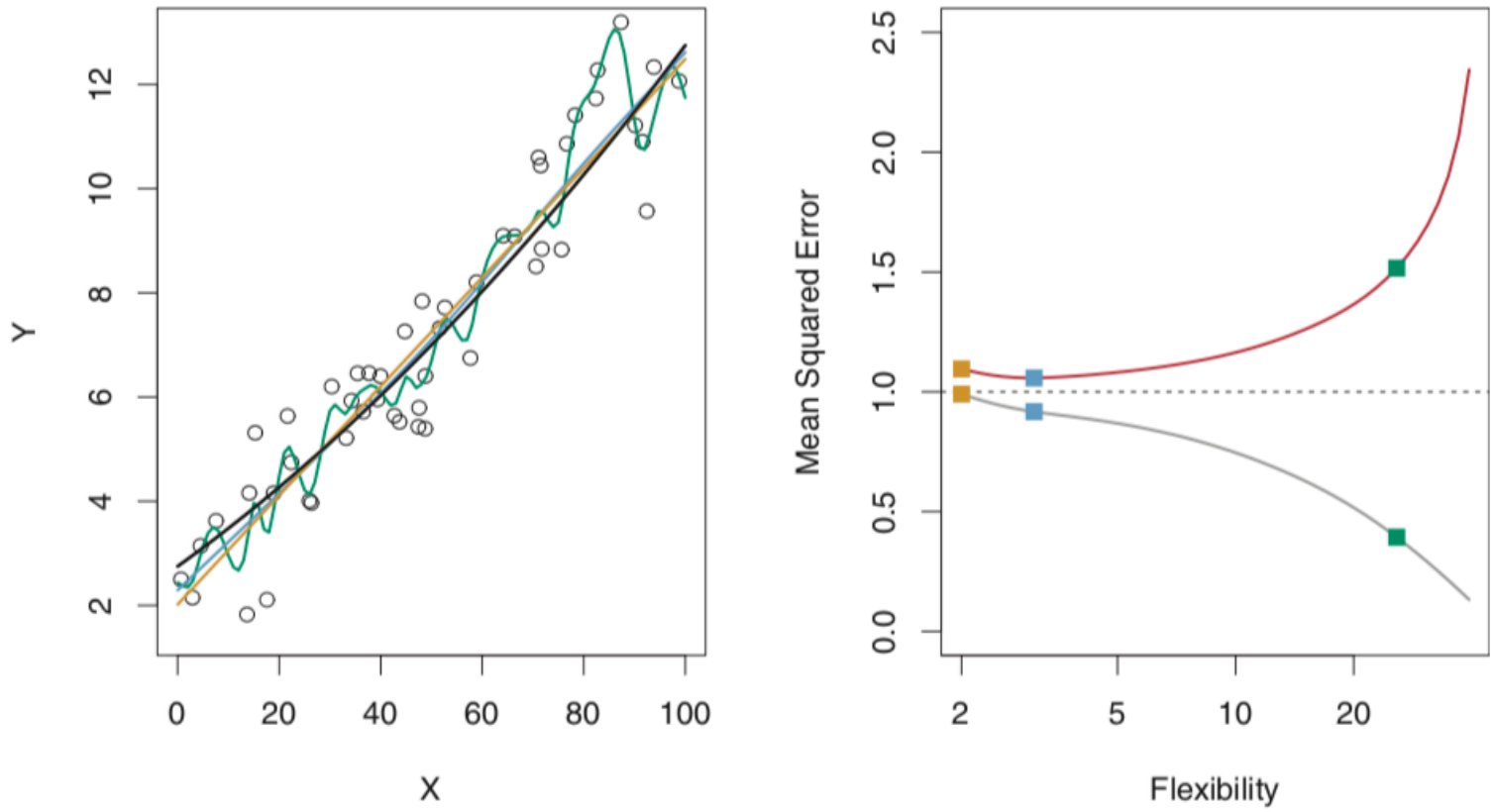


FIGURE 2.10. *Details are as in Figure 2.9, using a different true f that is much closer to linear. In this setting, linear regression provides a very good fit to the data.*

Training and Test MSE on Simulated Data 3

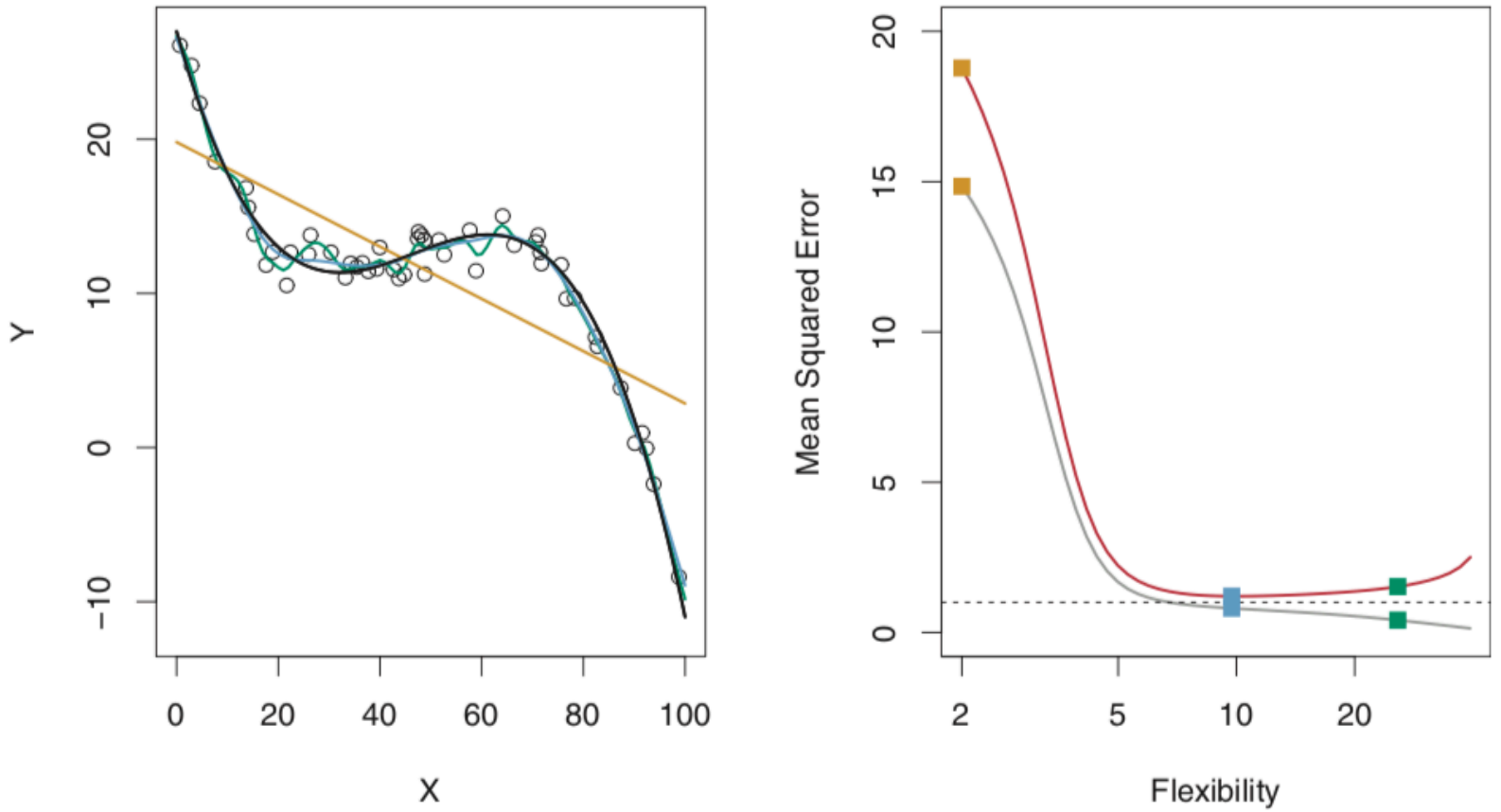


FIGURE 2.11. *Details are as in Figure 2.9, using a different f that is far from linear. In this setting, linear regression provides a very poor fit to the data.*

Training and Test MSE on Simulated Data Using Cross-Validation

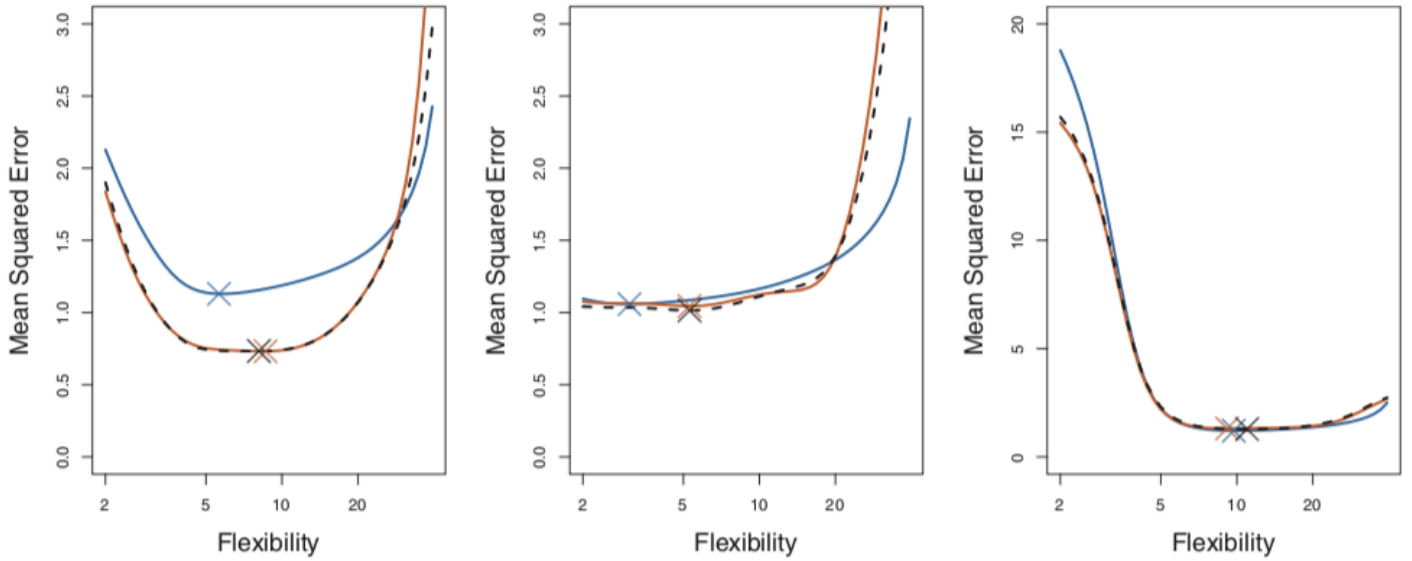


FIGURE 5.6. *True and estimated test MSE for the simulated data sets in Figures 2.9* (left)*, 2.10* (center)*, and 2.11* (right)*. The true test MSE is shown in blue, the LOOCV estimate is shown as a black dashed line, and the* 10*-fold CV estimate is shown in orange. The crosses indicate the minimum of each of the MSE curves.*

Despite the fact that CV underestimate the true test MSE, all of the CV curves come close to identifying the correct level of flexibility

Bias-Variance Tradeoff for k-fold CV

- LOOCV will give approximately unbiased estimates of the test error, since each training set contains n − 1 observations, which is almost as many as the number of observations in the full data set

- And performing k-fold CV for, say, $k = 5$ or $k = 10$ will lead to an intermediate level of bias, since each training set contains $(k - 1)n/k$ observations—fewer than in the LOOCV approach, but substantially more than in the validation set approach

- Therefore, from the perspective of bias reduction, it is clear that LOOCV is to be preferred to k-fold CV

Bias-Variance Tradeoff for k-fold CV

- LOOCV will have higher variance than k-fold CV with $k < n$
- In effect averaging the outputs of n fitted models, each of which is trained on an almost identical set of observations;
  - these outputs will be highly (positively) correlated with each other
- Mean of many highly correlated quantities has higher variance than does the mean of many quantities that are not as highly correlated
- Thus, the test error estimate resulting from LOOCV will tend to have higher variance than the test error estimate resulting from k-fold CV
- Typically *k=5* or *k=10* is used as they show optimal bias-variance tradeoff

Cross-Validation on Classification Problems

- Y is qualitative
- Similar to MSE, instead use misclassified observations.
- For example in the case of LOOCV,

$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n} Err_i$$

$$where$$

$$Err_i = I(y_i \neq \hat{y}_i)$$

- The k-fold CV error rate and validation set error rates are defined analogously.

Cross-validation (Logistic Regression)

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_2 + \beta_4 X_2^2$$

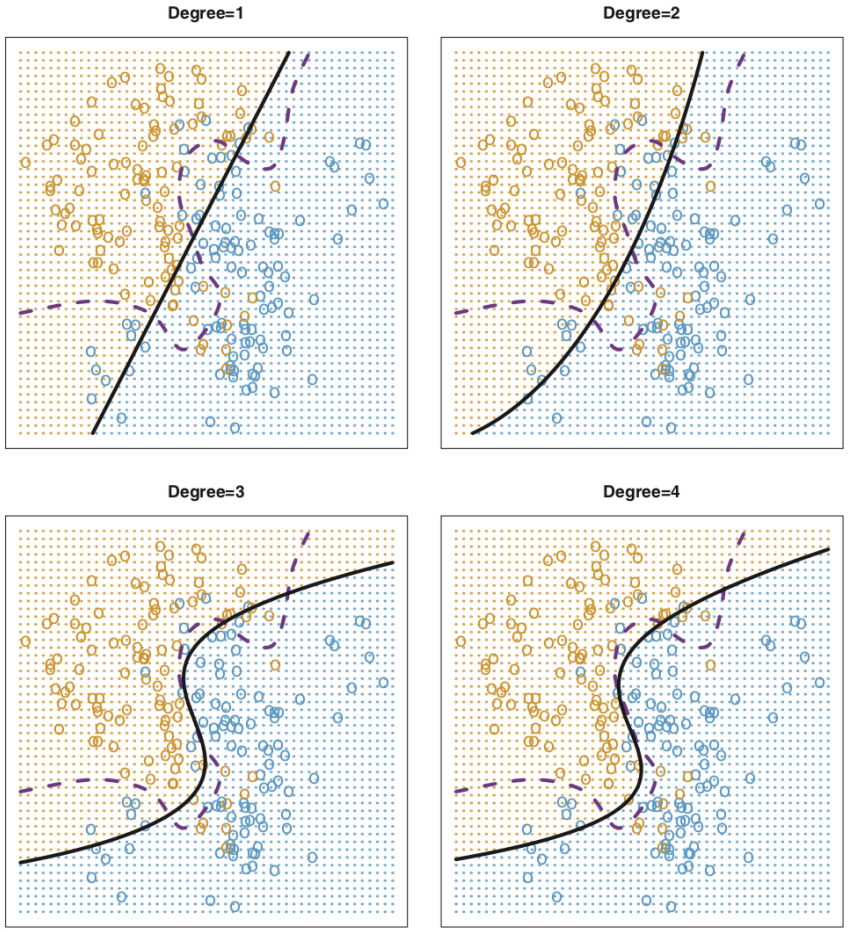


FIGURE 5.7. *Logistic regression fits on the two-dimensional classification data displayed in Figure 2.13. The Bayes decision boundary is represented using a purple dashed line. Estimated decision boundaries from linear, quadratic, cubic and quartic (degrees 1–4) logistic regressions are displayed in black. The test error rates for the four logistic regression fits are respectively* 0.201, 0.197, 0.160, *and* 0.162, *while the Bayes error rate is* 0.133.

CV on Classification (Errors)

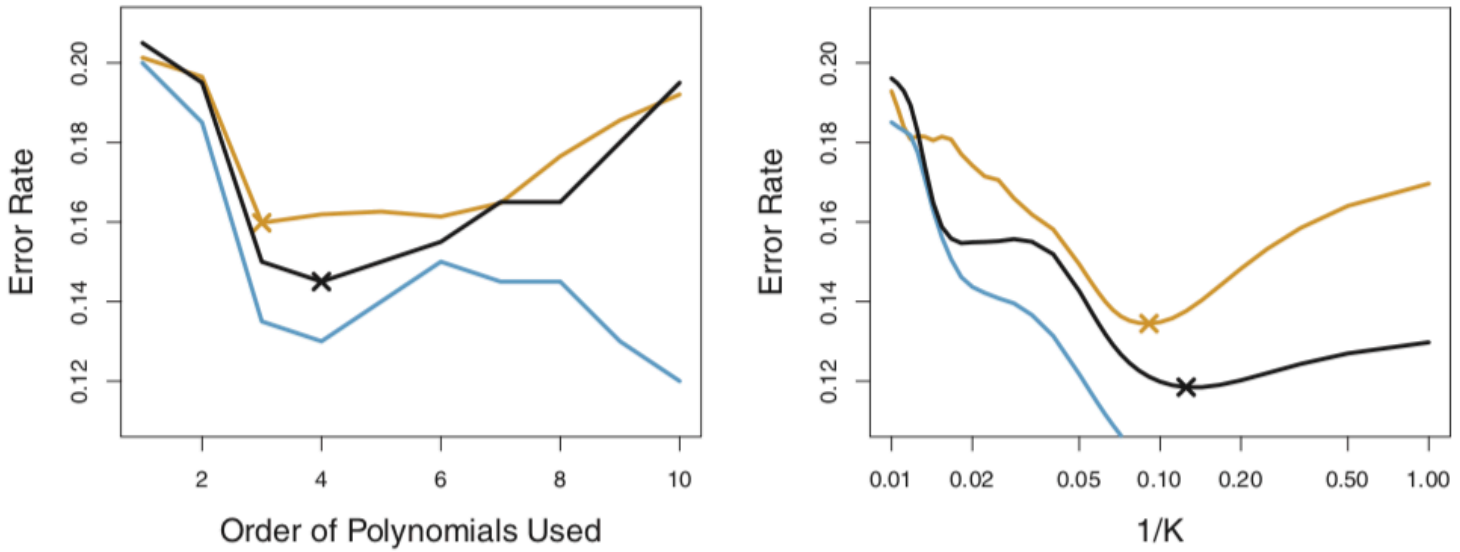


FIGURE 5.8. *Test error (brown), training error (blue), and 10-fold CV error (black) on the two-dimensional classification data displayed in Figure 5.7.* Left: *Logistic regression using polynomial functions of the predictors. The order of the polynomials used is displayed on the x-axis.* Right: *The KNN classifier with different values of K, the number of neighbors used in the KNN classifier.*

10-fold CV errors follow the same trend as test errors

Bootstrapping

The bootstrap is a widely applicable and extremely powerful statistical tool that can be used to quantify the uncertainty associated with a given estimator or statistical learning method.

- **Variance of a random variable, Z**

$$Var(Z) = E[(Z - E[Z])^2]$$
$$= E[Z^2 - 2ZE[Z] + E[Z]^2]$$
$$= E[Z^2] - E[Z]^2$$

- **Properties of $Var(Z)$**

$$Var(aZ) = E[a^2Z^2] - E[aZ]^2 = a^2Var(Z)$$

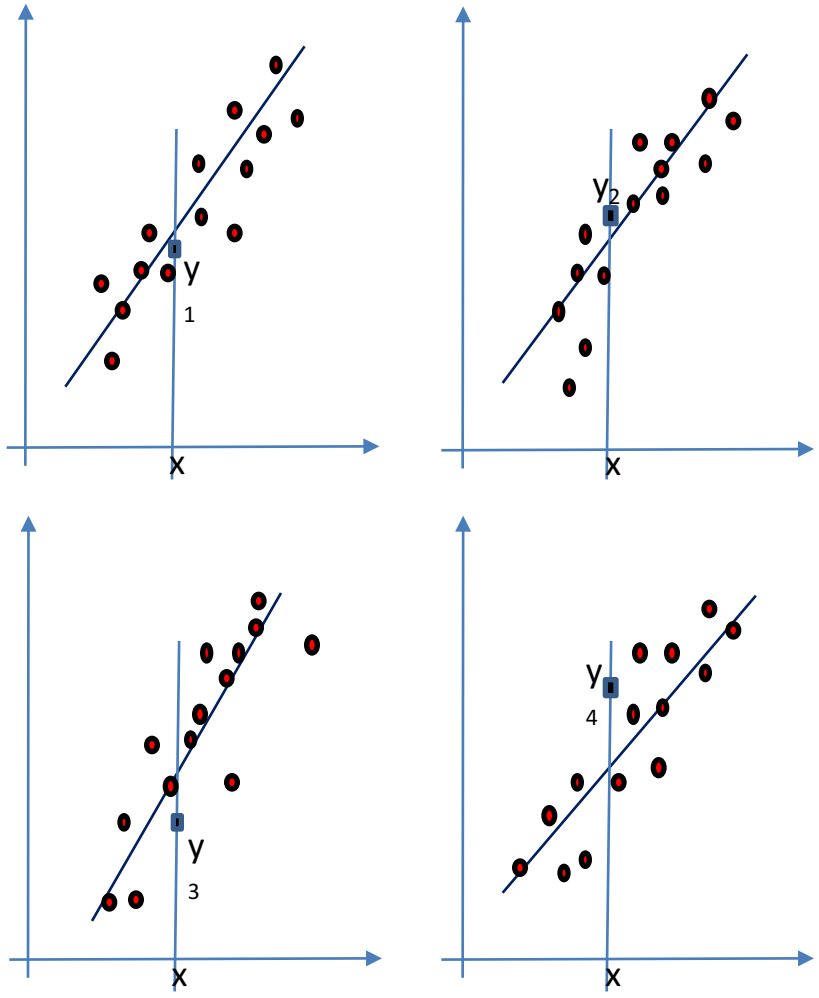
Why Bootstrapping Works?

$$(x_1^1, y_1^1), (x_2^1, y_2^1), \cdots, (x_k^1, y_k^1) \quad \leftrightarrow (x, y_1)$$

$$(x_1^2, y_1^2), (x_2^2, y_2^2), \cdots, (x_k^2, y_k^2) \quad \leftrightarrow (x, y_2)$$

$$\vdots$$

$$(x_1^m, y_1^m), (x_2^m, y_2^m), \cdots, (x_k^m, y_k^m) \leftrightarrow (x, y_m)$$



$$E[y_i] = y \quad Var[y] = E[(y_i - E[y_i])^2] = E[(y_i - y)^2]$$

$$Z = \frac{1}{m}\sum y_i$$

$$E[Z] = \frac{1}{m}\sum E[y_i] = \frac{1}{m}(mE[y]) = E[y]$$

$$Var[Z] = Var[\frac{1}{m}\sum y_i] = \frac{1}{m^2}Var[\sum y_i]$$

If y_i's are **uncorrelated**

$$\frac{1}{m^2}Var[\sum y_i] = \frac{1}{m^2}\sum Var[y_i]$$

$$\frac{1}{m^2}Var[\sum y_i] = \frac{1}{m^2}\sum Var[y_i] = \frac{1}{m^2}\sum (E[(y_i^2 - E[y_i])^2] = \frac{1}{m^2}\sum (E[(y_i^2 - y)^2] = \frac{1}{m^2}\sum Var[y] = \frac{1}{m}Var[y]$$

Why Bootstrapping Works?

$$E[Z] = E[y] = y$$
$$Var[Z] = \frac{1}{m} Var[y]$$

- As m increases, variance is reduced and the aggregated prediction is closer to the true value.
- Unfortunately, we DON'T have several sets of samples!!!
- What should we do?
- Of course, make a uniform sampling from given sample set, with replacement and use each sample set as a bootstrap sample set!!!

Bootstrapping Error

$$\mathrm{SE}_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1}\sum_{r=1}^{B}\left(\hat{\alpha}^{*r} - \frac{1}{B}\sum_{r'=1}^{B}\hat{\alpha}^{*r'}\right)^2}.$$

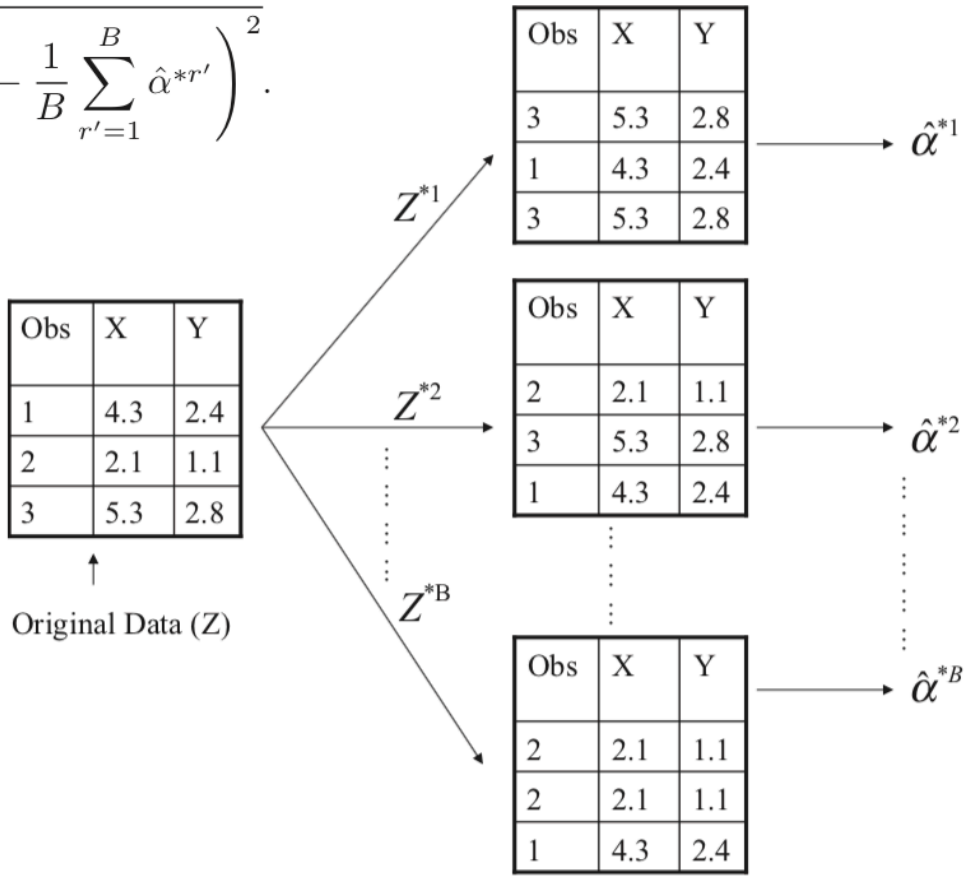


FIGURE 5.11. *A graphical illustration of the bootstrap approach on a small sample containing $n = 3$ observations. Each bootstrap data set contains n observations, sampled with replacement from the original data set. Each bootstrap data set is used to obtain an estimate of α.*

Regularization

Consider the linear model:

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$$

If the β_j's are unconstrained, they can explode

They are susceptible to high variance

To control variance, we might *regularize* the coefficients

- control how large they can grow

These are also known as *shrinkage* methods

Two best-known techniques:

- Ridge regression
- Lasso regression

Ridge Regression

- Very similar to least squares
- In least squares: $\mathrm{RSS} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$.
- In ridge regression: $\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = \mathrm{RSS} + \lambda \sum_{j=1}^{p} \beta_j^2,$
- λ is called the *tuning* parameter and the additive term is called the *shrinkage* penalty
- $\lambda = 0$ is same as least squares, but when it grows, the impact of shrinkage penalty grows
- Unlike least squares that produces only one set of coefficients, for ridge regression we will have several estimates for different λ's
- Note: the penalty does not apply to β_0 as it represents the mean

Selecting tuning parameter λ

- Need disciplined way of selecting λ
- That is, we need to "tune" the value of λ
- In their original paper, Hoerl and Kennard introduced ridge traces:
  - Plot the components of $\hat{\beta}_{\lambda}^{ridge}$ against λ
  - Choose λ for which the coefficients are not rapidly changing
  - and have "sensible" signs
- No objective basis; heavily criticized by many
- Standard practice now is to use cross-validation

Standardized Ridge Regression Coefficients

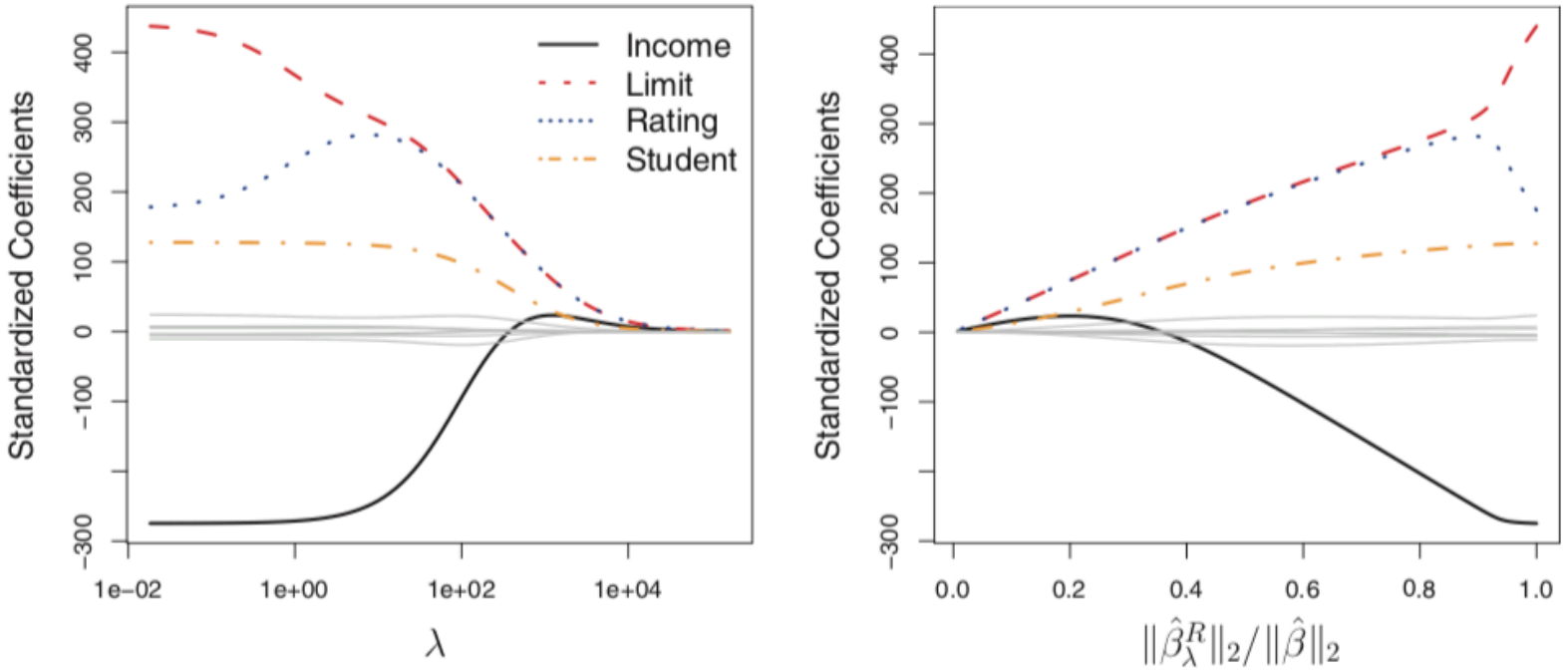


FIGURE 6.4. *The standardized ridge regression coefficients are displayed for the* `Credit` *data set, as a function of* λ *and* $\|\hat{\beta}_\lambda^R\|_2/\|\hat{\beta}\|_2$.

Not scale invariant, therefore need to standardize the predictors, so they are of same scale

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

Does Ridge Regression Improve Over Least Squares?

- Decreased variance but increase in bias

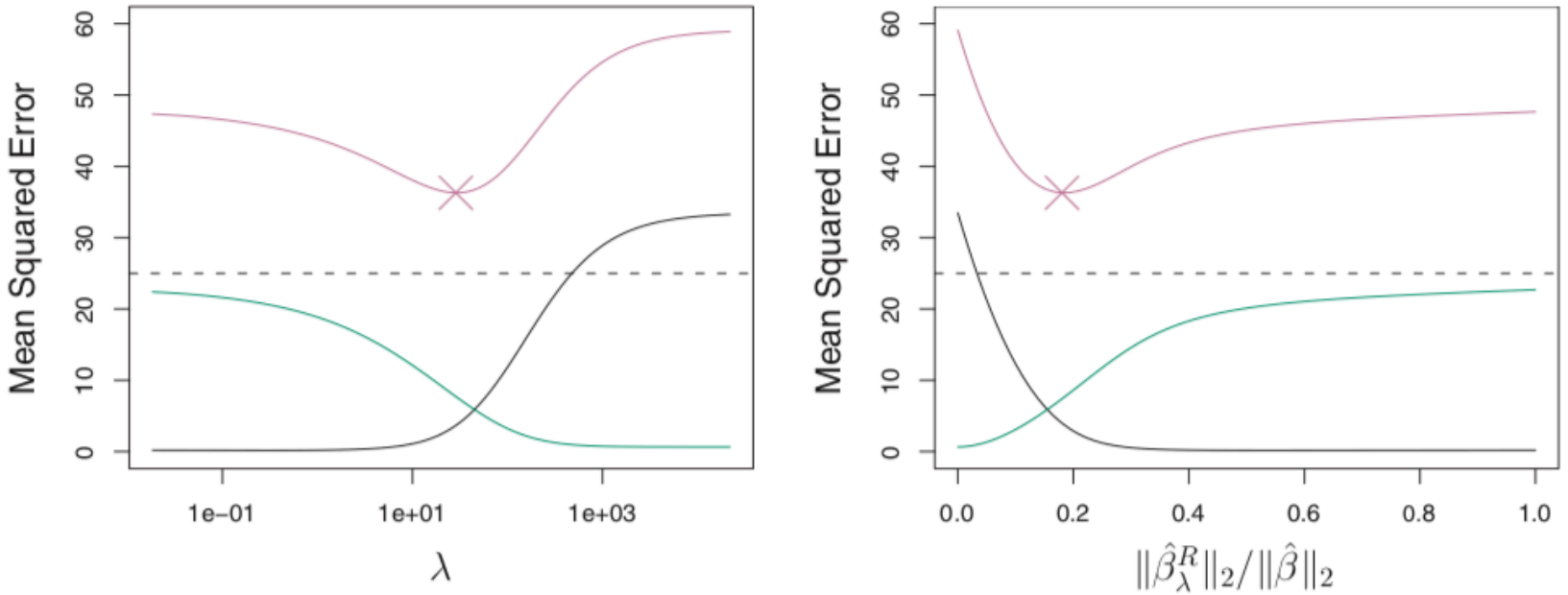


FIGURE 6.5. *Squared bias (black), variance (green), and test mean squared error (purple) for the ridge regression predictions on a simulated data set, as a function of* λ *and* $\|\hat{\beta}^R_\lambda\|_2/\|\hat{\beta}\|_2$. *The horizontal dashed lines indicate the minimum possible MSE. The purple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

Ridge Regression vs. Least Squares

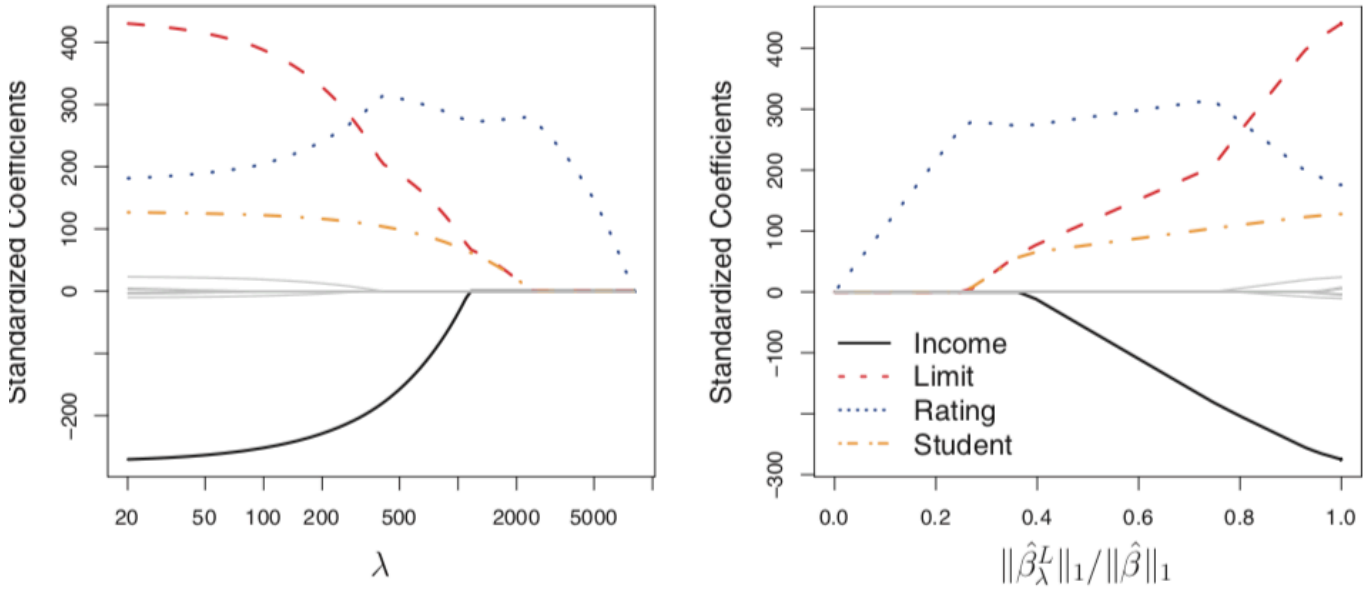
- When relationship between the response and the predictors is close to linear
  - the least squares estimates will have low bias but may have high variance
  - small change in data will lead to big change in coefficients
  - p ~= n, least squares will be highly variable; p > n, least squares will not even have unique solution
  - ridge regression can still perform well by trading off a small increase in bias for a large decrease in variance
- Ridge regression works best in situations where the least squares estimates have high variance.

Ridge Regression – Disadvantage and Lasso

- Will include all *p* predictors even though the coefficients may be negligible
- May not be a problem for model accuracy, but interpretation might suffer
- Therefore, we need a method that excludes unimportant features
- Lasso – also performs *variable selection* but identifying good tuning parameter is essential

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

Lasso



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

Lasso gives the null model where all coefficients are zero when λ becomes sufficiently large

Lasso presents a model in which rating, limit, student and income appear serially

Another Formulation of Ridge and Lasso Regression

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

and

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

It can be proved that

For every value of λ, there is some s such that the Lasso equation and (6.8) will give the same Lasso coefficient estimates.

Similarly, for every value of λ there is a corresponding *s* such that Ridge regression equation and (6.9) will give the same ridge regression coefficient estimates.

Variable Selection Property of Ridge and Lasso Regression

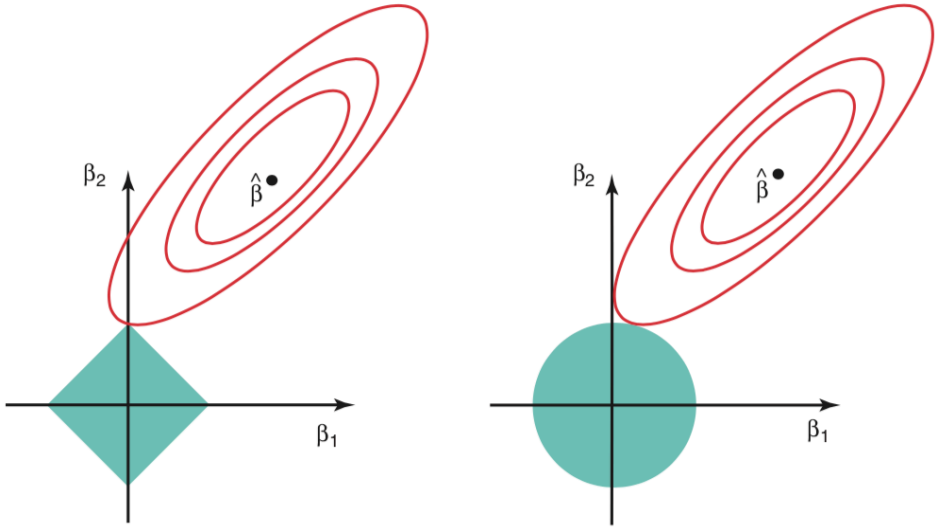


FIGURE 6.7. *Contours of the error and constraint functions for the lasso* (left) *and ridge regression* (right). *The solid blue areas are the constraint regions,* $|\beta_1| + |\beta_2| \leq s$ *and* $\beta_1^2 + \beta_2^2 \leq s$, *while the red ellipses are the contours of the RSS.*

The ellipses that are centered around β^ represent regions of constant RSS.

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.

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.

Ridge Regression vs. Lasso Regression

Using all 45 predictors

Similar behavior – almost identical biases

Variance is ridge is slightly lower

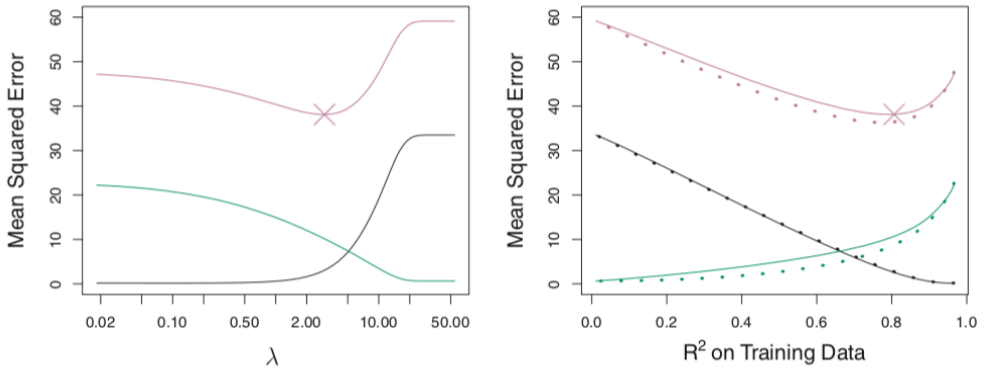


FIGURE 6.8. Left: *Plots of squared bias (black), variance (green), and test MSE (purple) for the lasso on a simulated data set.* Right: *Comparison of squared bias, variance and test MSE between lasso (solid) and ridge (dashed). Both are plotted against their R^2 on the training data, as a common form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

Using 2/45 predictors

Lasso tends to outperform Ridge in Bias, Variance and MSE

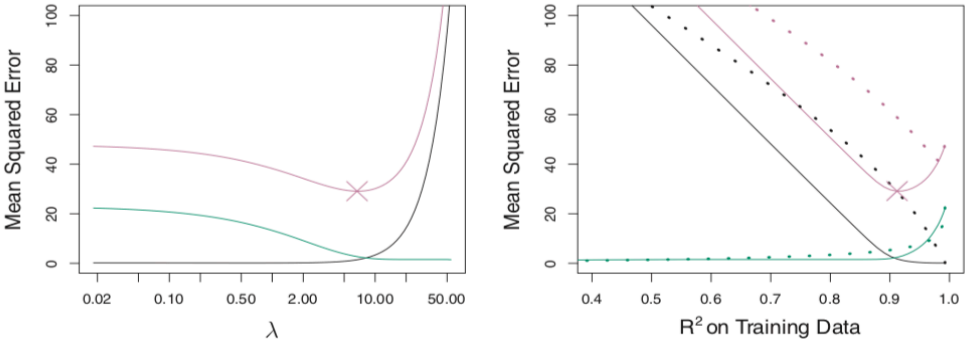


FIGURE 6.9. Left: *Plots of squared bias (black), variance (green), and test MSE (purple) for the lasso. The simulated data is similar to that in Figure 6.8, except that now only two predictors are related to the response.* Right: *Comparison of squared bias, variance and test MSE between lasso (solid)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

Ridge vs. Lasso

- Neither Lasso nor Ridge will universally dominate the other
  - depends on the number of predictor variables used

- 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.

- Unlike ridge regression, the lasso performs variable selection, and hence results in models that are easier to interpret.

Selecting the tuning parameter

- Implementing ridge regression and the lasso requires a method for selecting a value for the tuning parameter λ or the constraint parameter s

- Cross-validation provides a simple way to tackle this problem
  - We choose a grid of λ values, and compute the cross-validation error for each value of λ
  - 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.

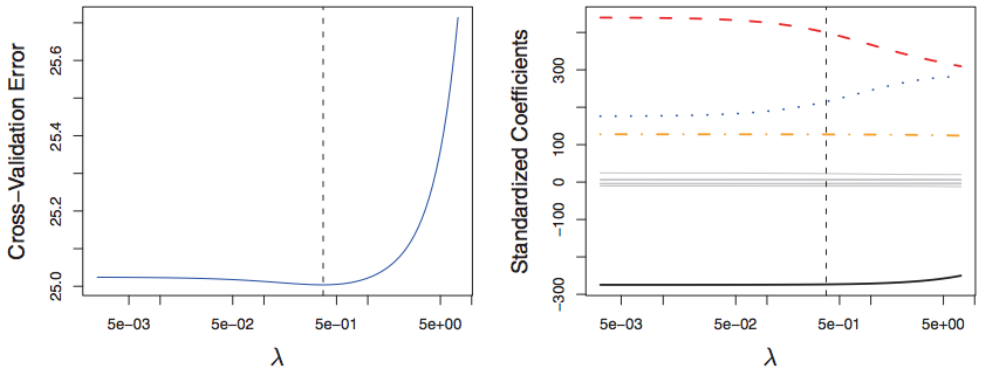


FIGURE 6.12. Left: *Cross-validation errors that result from applying ridge regression to the* `Credit` *data set with various value of* λ. Right: *The coefficient estimates as a function of* λ. *The vertical dashed lines indicate the value of* λ *selected by cross-validation.*

Selecting the tuning parameter

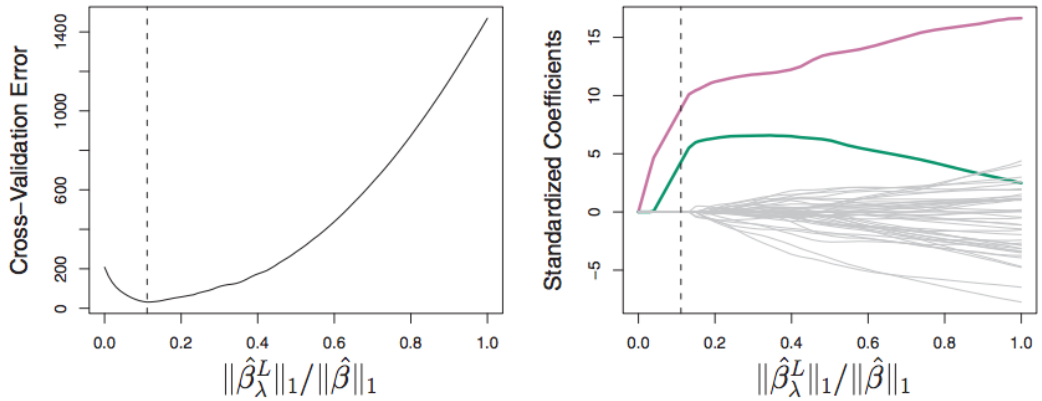


FIGURE 6.13. Left: *Ten-fold cross-validation MSE for the lasso, applied to the sparse simulated data set from Figure 6.9.* Right: *The corresponding lasso coefficient estimates are displayed. The vertical dashed lines indicate the lasso fit for which the cross-validation error is smallest.*

- Vertical line – cross-validation error smallest
- Red and green – two predictor variables
- Lasso correctly given much larger coefficient estimates to the two signal predictors
  - also the minimum cross validation error corresponds to a set of coefficient estimates for which only the signal variables are non-zero
- Cross-validation together with the lasso has correctly identified the two signal variables in the model where as least squares assigns a large coefficient estimate for these variables