DATA 572: Supervised Learning

2023W2

Shan Du

Resampling Methods

- Resampling methods involve repeatedly drawing samples from a training set and refitting a model of interest on each sample in order to obtain additional information about the fitted model.
- Resampling approaches can be computationally expensive, because they involve fitting the same statistical method multiple times using different subsets of the training data.
- Two of the most commonly used resampling methods are *cross-validation* and the *bootstrap*.

Cross-Validation

- There is a distinction between the *test error rate* and the *training error rate*.
- In the absence of a very large designated test set that can be used to directly estimate the test error rate, a number of techniques can be used to estimate this quantity using the available training data.
- One way is to 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.

The Validation Set Approach

- The *validation set approach* involves randomly dividing the available set of observations into two parts, a *training set* and a *validation set* or *hold-out set*.
- The model is fit on the training set, and the fitted model is used to predict the responses for the observations in the validation set.
- The resulting validation set error rate—typically assessed using MSE in the case of a quantitative response—provides an estimate of the test error rate.

The Validation Set Approach

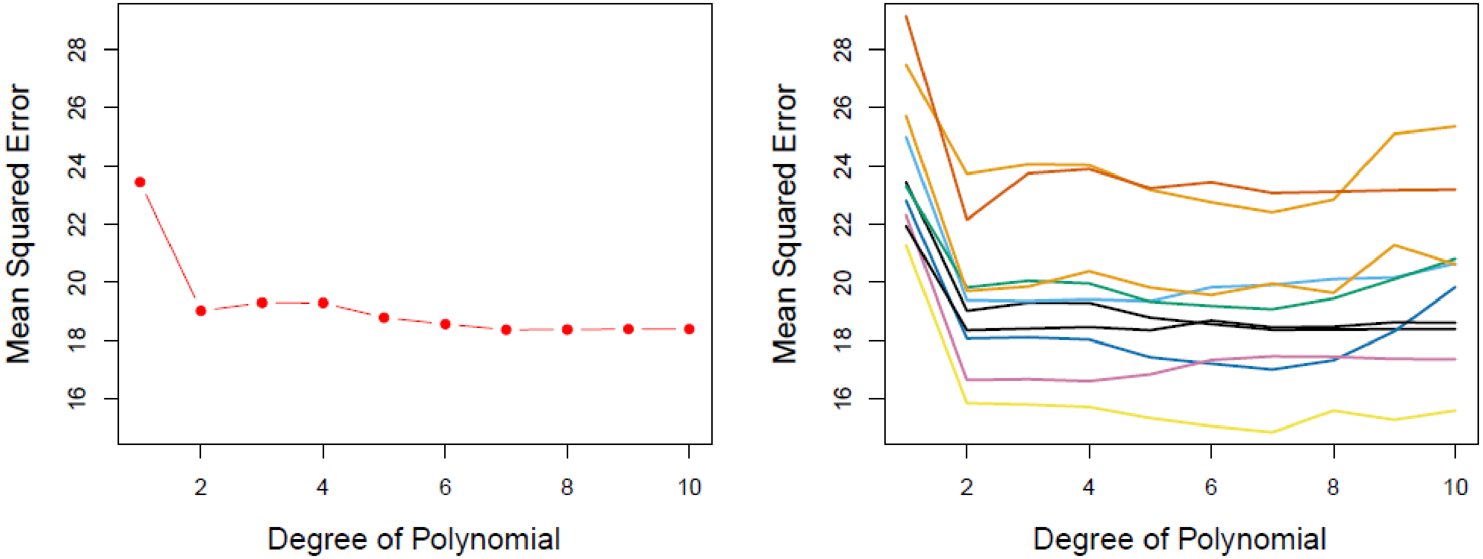


FIGURE 5.2. *The validation set approach was used on the* Auto *data set in order to estimate the test error that results from predicting* mpg *using polynomial functions of* horsepower. Left: *Validation error estimates for a single split into training and validation data sets.* Right: *The validation method was repeated ten times, each time using a different random split of the observations into a training set and a validation set. This illustrates the variability in the estimated test MSE that results from this approach.*

Leave-One-Out Cross-Validation

- *Leave-one-out cross-validation* (LOOCV) is closely related to the validation set approach, but it tries to use data adequately.
- Like the validation set approach, LOOCV involves splitting the set of observations into two parts. However, 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.

Leave-One-Out Cross-Validation

- 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.
- We can repeat the procedure by selecting (x_2, y_2) for the validation data, training the statistical learning procedure on the $n-1$ observations $\{(x_1, y_1), (x_3, y_3), \ldots, (x_n, y_n)\}$.
- Repeating this approach n times produces n squared errors.

k-Fold Cross-Validation

- 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.
- This procedure is repeated k times; each time, a different group of observations is treated as a validation set.

k-Fold Cross-Validation

- It is not hard to see that LOOCV is a special case of k-fold CV in which k is set to equal n.
- In practice, one typically performs k-fold CV using $k = 5$ or $k = 10$.
- What is the advantage of using $k = 5$ or $k = 10$ rather than $k = n$? The most obvious advantage is computational.
  - LOOCV requires fitting the statistical learning method n times.
  - Performing k -fold CV requires fitting the learning procedure only k times.

k-Fold Cross-Validation

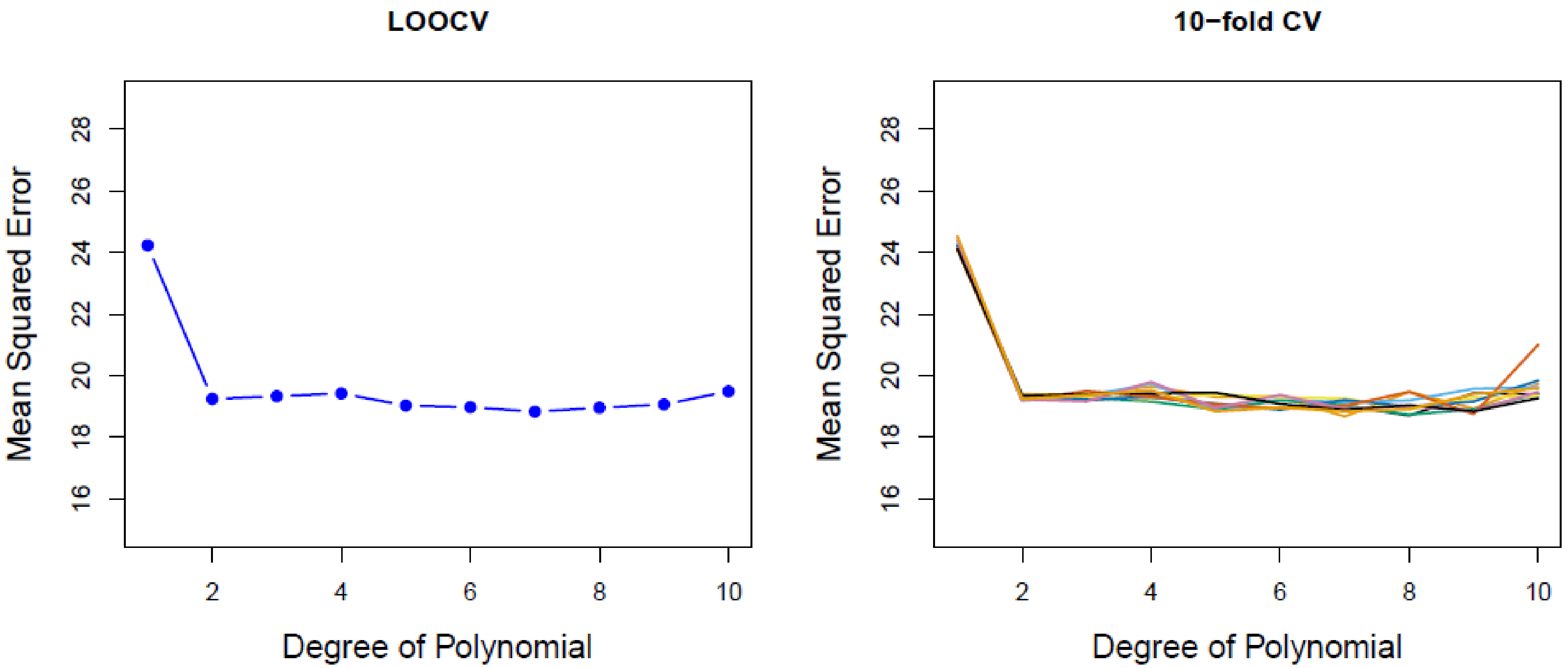


FIGURE 5.4. *Cross-validation was used on the* `Auto` *data set in order to estimate the test error that results from predicting* `mpg` *using polynomial functions of* `horsepower`. Left: *The LOOCV error curve.* Right: 10*-fold CV was run nine separate times, each with a different random split of the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

Cross-Validation on Classification Problems

- In the classification setting, we use the number of misclassified observations to quantify test error. The LOOCV error rate takes the form

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

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

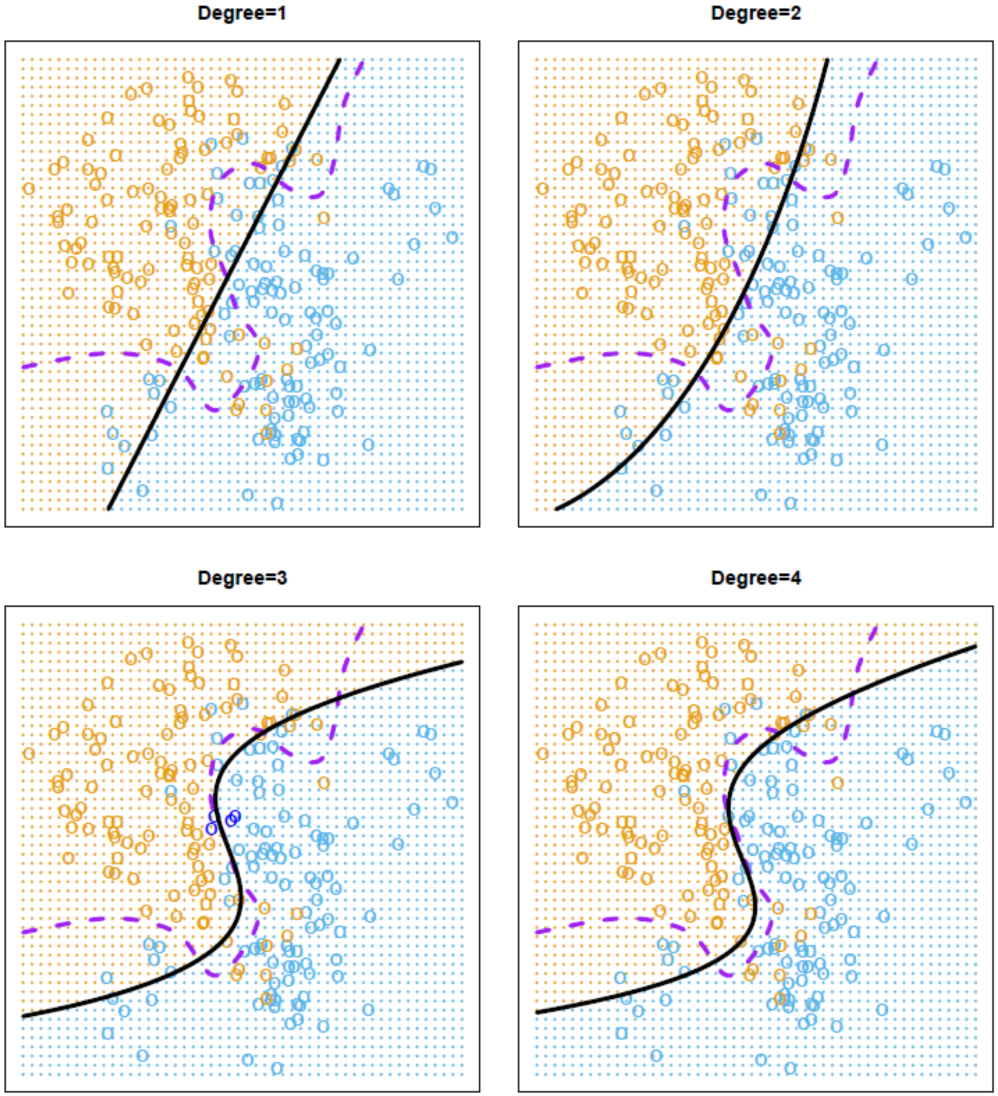


FIGURE 5.7. *Logistic regression fits on the two-dimensional classification data displayed in Figure 2.13. The Bayes decision boundary is represented using a purple dashed line. Estimated decision boundaries from linear, quadratic, cubic and quartic (degrees 1–4) logistic regressions are displayed in 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.

Cross-Validation on Classification Problems

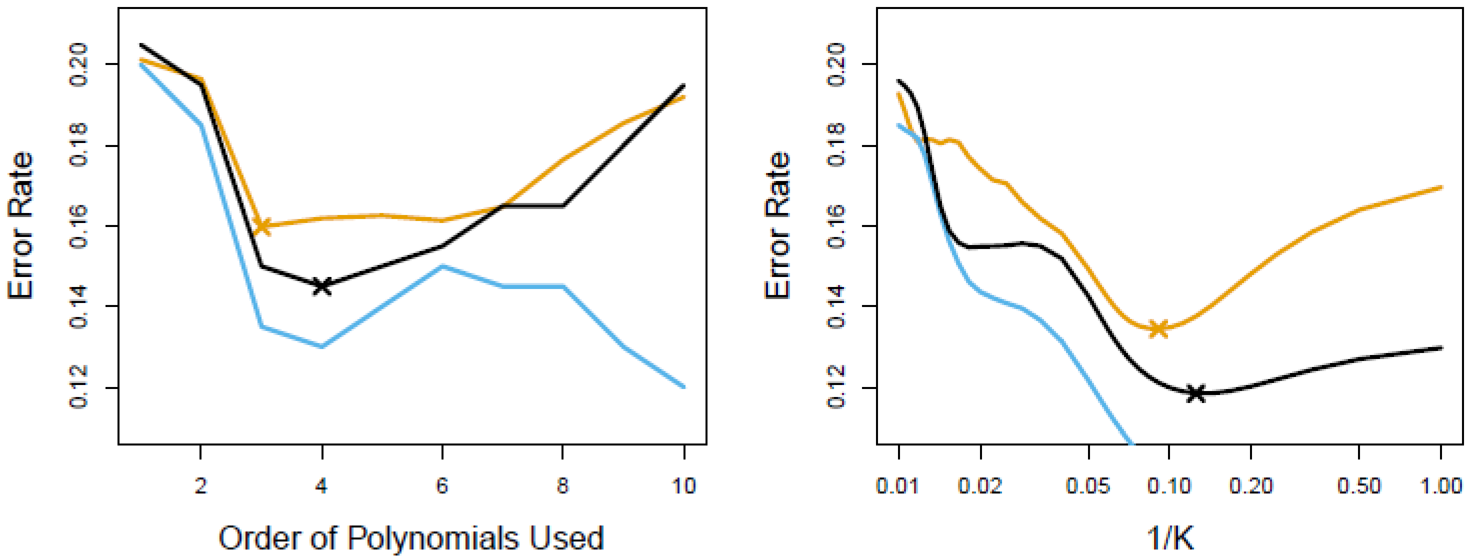


FIGURE 5.8. *Test error (brown), training error (blue), and 10-fold CV error (black) on the two-dimensional classification data displayed in Figure 5.7.* Left: *Logistic regression using polynomial functions of the predictors. The order of the polynomials used is displayed on the x-axis.* Right: *The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

The Bootstrap

- 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.
- It can be easily applied to a wide range of statistical learning methods, including some for which a measure of variability is otherwise difficult to obtain and is not automatically output by statistical software.

The Bootstrap

- Suppose that we wish to invest a fixed sum of money in two financial assets that yield returns of X and Y, respectively, where X and Y are random quantities.
- We will invest a fraction α of our money in X, and will invest the remaining $1-\alpha$ in Y. Since there is variability associated with the returns on these two assets, we wish to choose α to minimize the total risk, or variance, of our investment.

The Bootstrap

- In other words, we want to minimize $\mathrm{Var}(\alpha X + (1-\alpha)Y)$.
- The value that minimizes the risk is given by

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}}$$

where $\sigma_X^2 = \mathrm{Var}(X)$, $\sigma_Y^2 = \mathrm{Var}(Y)$, and $\sigma_{XY} = \mathrm{Cov}(X, Y)$.

The Bootstrap

- In reality, the quantities σ_X^2, σ_Y^2, and σ_{XY} are unknown.
- We can compute estimates for these quantities, $\hat{\sigma}_X^2$, $\hat{\sigma}_Y^2$, and $\hat{\sigma}_{XY}$, using a data set that contains past measurements for X and Y.
- We can then estimate the value of α that minimizes the variance of our investment using

$$\alpha = \frac{\hat{\sigma}_Y^2 - \hat{\sigma}_{XY}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\hat{\sigma}_{XY}}$$

The Bootstrap

- We estimate α on a simulated data set.
- We simulate 100 pairs of returns for the investments X and Y. We used these returns to estimate σ_X^2, σ_Y^2, and σ_{XY} which can be used to estimate α.

The Bootstrap

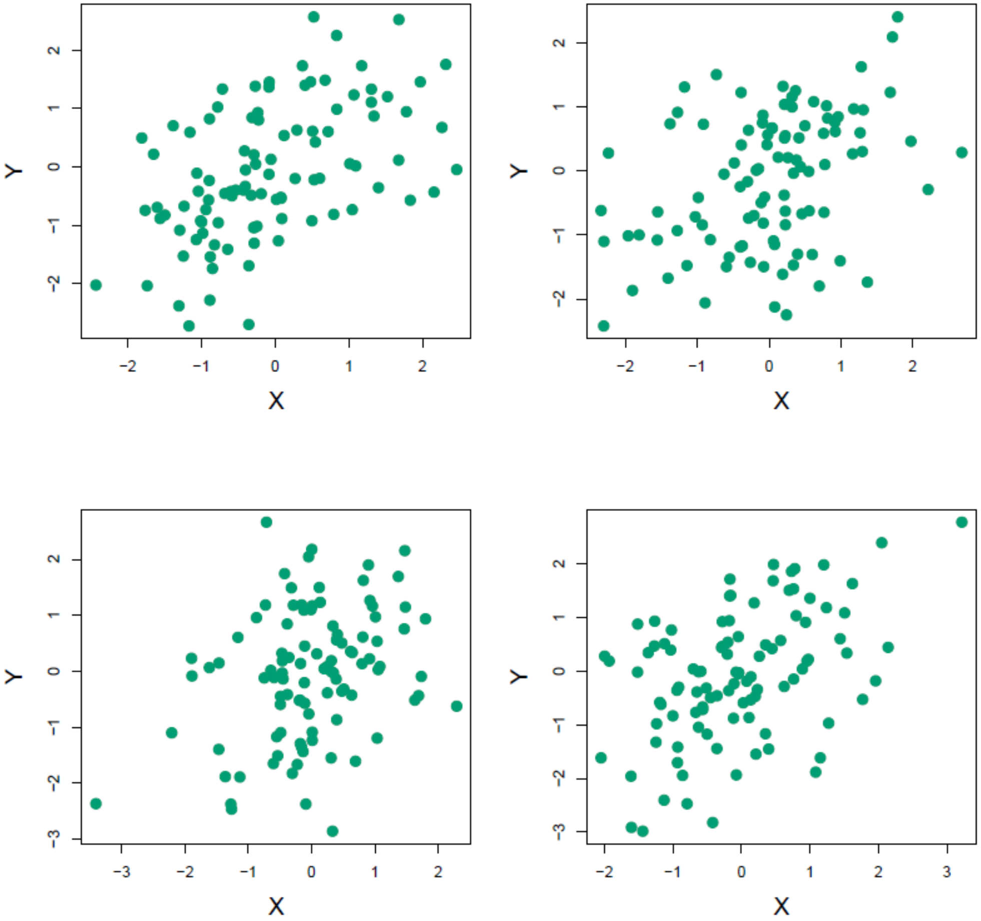
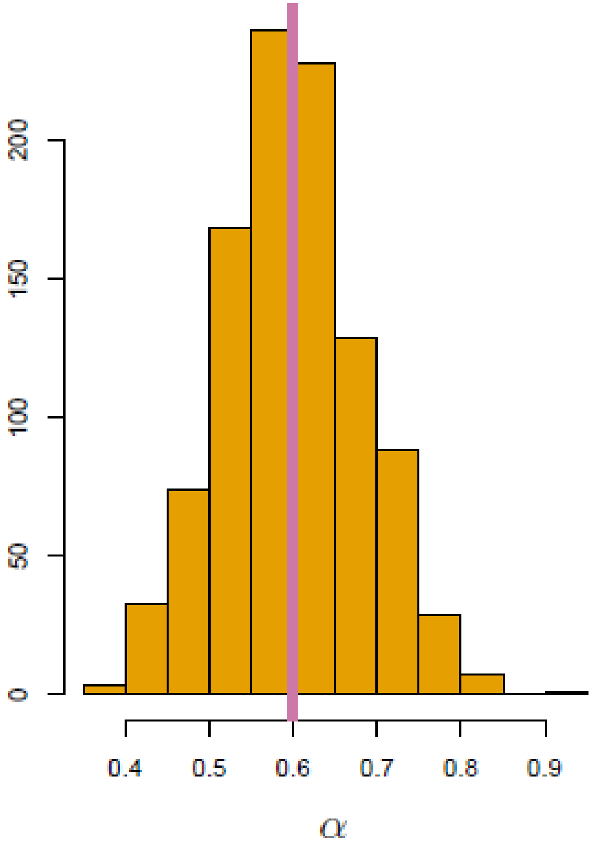


FIGURE 5.9. *Each panel displays* 100 *simulated returns for investments* X *and* Y. *From left to right and top to bottom, the resulting estimates for* α *are* 0.576, 0.532, 0.657, *and* 0.651.

The Bootstrap

- It is natural to wish to quantify the accuracy of our estimate of α. To estimate the standard deviation of $\hat{\alpha}$, we repeat the process 1,000 times.



We set $\sigma_X^2 = 1$, $\sigma_Y^2 = 1.25$, and $\sigma_{XY} = 0.5$, then we know that the true value of α is 0.6.

The mean over all 1,000 estimates of α is

$$\bar{\alpha} = 0.5996$$

The standard deviation of the estimates is 0.083.

The Bootstrap

- In practice, however, the procedure for estimating α cannot be applied, because for real data we cannot generate new samples from the original population.
- The bootstrap approach allows us to use a computer to emulate the process of obtaining new sample sets, so that we can estimate the variability of α without generating additional samples.
- Rather than repeatedly obtaining independent data sets from the population, we instead obtain distinct data sets by repeatedly sampling observations *from the original data set*.

The Bootstrap

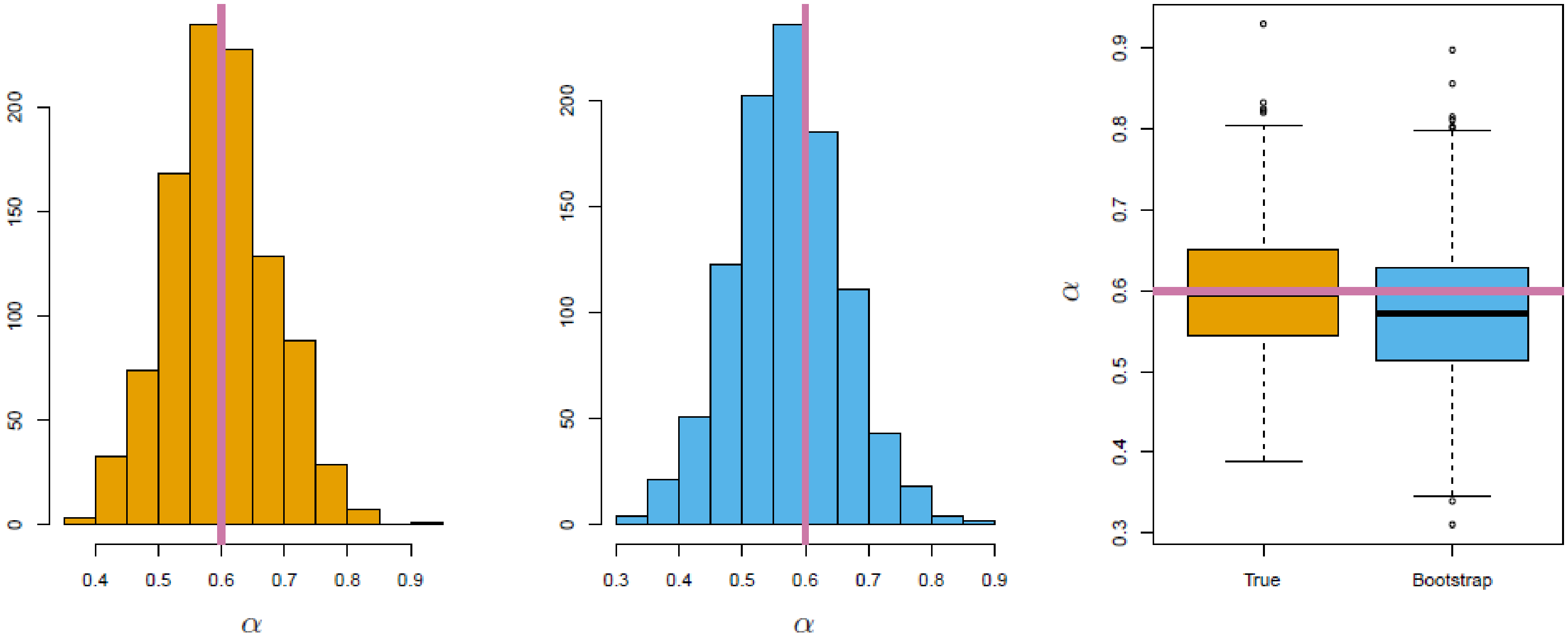
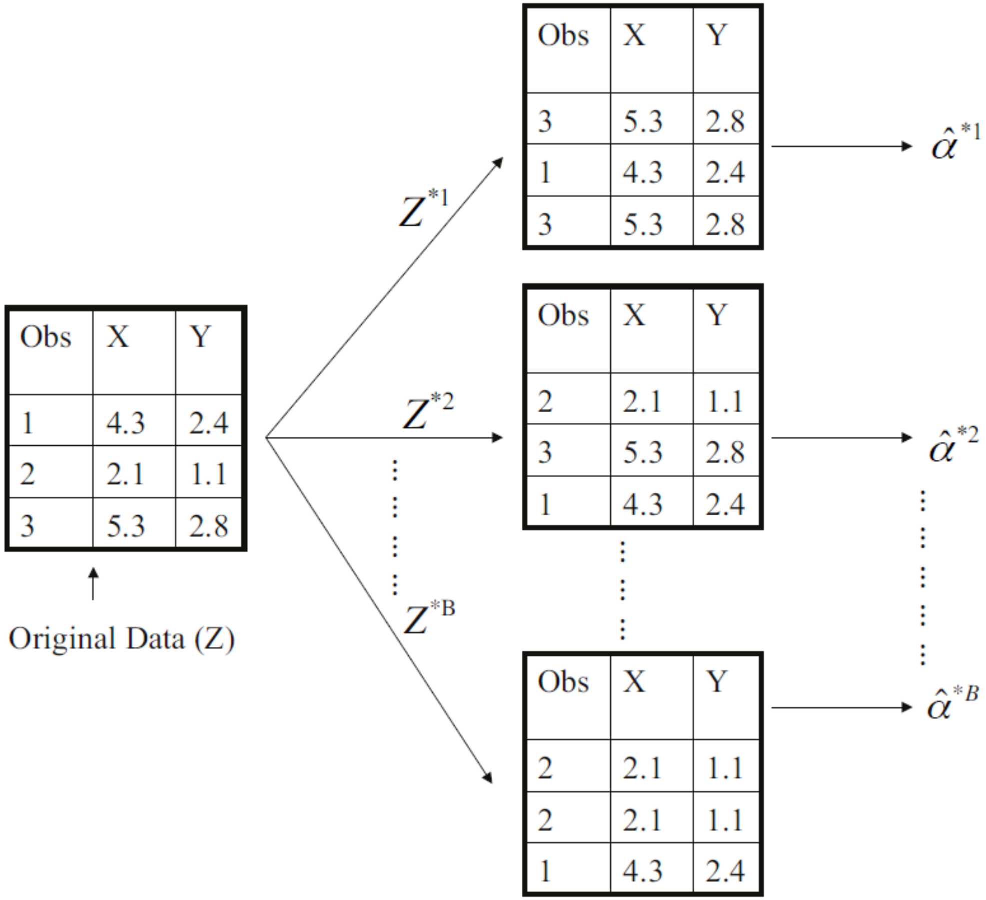


FIGURE 5.10. Left: *A histogram of the estimates of α obtained by generating 1,000 simulated data sets from the true population.* Center: *A histogram of the estimates of α obtained from 1,000 bootstrap samples from a single data set.* Right: *The estimates of α displayed in the left and center panels are shown as boxplots. In each panel, the pink line indicates the true value of α.*

The Bootstrap



Obs	X	Y
1	4.3	2.4
2	2.1	1.1
3	5.3	2.8

Original Data (Z)

Obs	X	Y
3	5.3	2.8
1	4.3	2.4
3	5.3	2.8

Obs	X	Y
2	2.1	1.1
3	5.3	2.8
1	4.3	2.4

Obs	X	Y
2	2.1	1.1
2	2.1	1.1
1	4.3	2.4

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.*

Feature Selection

- Simple linear model can be improved, by replacing plain least squares fitting with some alternative fitting procedures.
- Alternative fitting procedures can yield better *prediction (or classification) accuracy* and *model interpretability*.

Feature Selection

- *Prediction Accuracy*: Provided that the true relationship between the response and the predictors is approximately linear, the least squares estimates will have low bias.
  - If $n \gg p$—that is, the number of observations is much larger than the number of variables—then the least squares estimates tend to also have low variance, and hence will perform well on test observations.
  - However, if n is not much larger than p, then there can be a lot of variability in the least squares fit, resulting in overfitting and consequently poor predictions on future observations not used in model training.
  - And if $p > n$, then there is no longer a unique least squares coefficient estimate: there are infinitely many solutions. Each of these least squares solutions gives zero error on the training data, but typically very poor test set performance due to extremely high variance.

Feature Selection

- Model Interpretability: It is often the case that some or many of the variables used in a multiple regression model are in fact not associated with the response. Including such *irrelevant* variables leads to unnecessary complexity in the resulting model. By removing these variables—that is, by setting the corresponding coefficient estimates to zero—we can obtain a model that is more easily interpreted. Now least squares is extremely unlikely to yield any coefficient estimates that are exactly zero.

Subset Selection

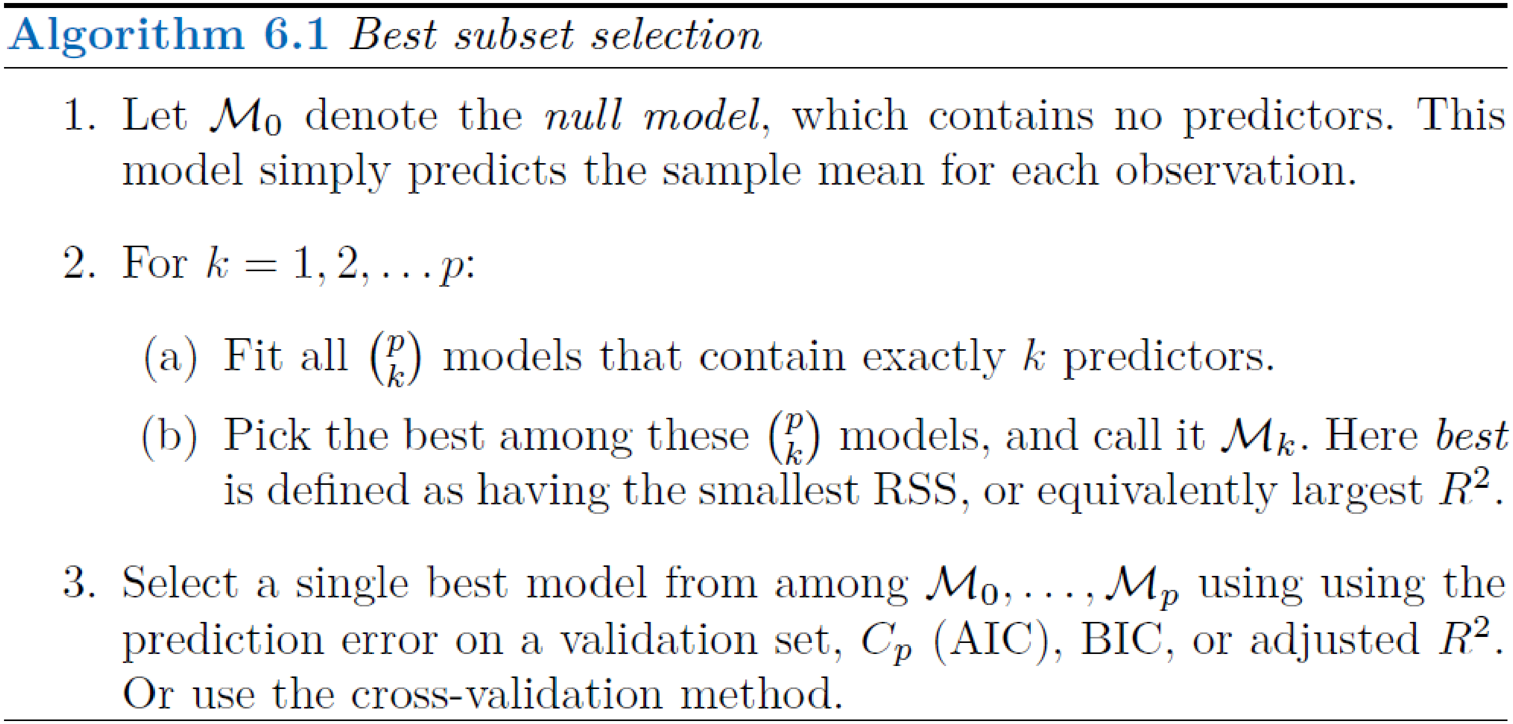
- Subset selection involves identifying a subset of the p predictors that we believe to be related to the response. We then fit a model using least squares on the reduced set of variables.

Best Subset Selection

- To perform *best subset selection*, we fit a separate least squares regression for each possible combination of the p predictors. That is, we fit all p models that contain exactly one predictor, all $\binom{p}{2} = p(p-1)/2$ models that contain exactly two predictors, and so forth.
- We then look at all of the resulting models, with the goal of identifying the one that is *best*.

Best Subset Selection

- The problem of selecting the best model from among the 2^p possibilities considered by best subset selection is not trivial. This is usually broken up into two stages:



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 using the prediction error on a validation set, C_p (AIC), BIC, or adjusted R^2. Or use the cross-validation method.

Best Subset Selection

- In order to select a single best model, we must simply choose among the $p + 1$ options in Algorithm 6.1. This task must be performed with care, because the RSS of these $p + 1$ models decreases monotonically, and the R^2 increases monotonically, as the number of features included in the models increases.
- If we use these statistics to select the best model, then we will always end up with a model involving all of the variables.
- The problem is that a low RSS or a high R^2 indicates a model with a low *training* error, whereas we wish to choose a model that has a low *test* error.

Best Subset Selection

- Therefore, in Step 3, we use the error on a validation set, C_p, BIC, or adjusted R^2 in order to select among $\mathcal{M}_0, \mathcal{M}_1, \ldots, \mathcal{M}_p$.
- If cross-validation is used to select the best model, then Step 2 is repeated on each training fold, and the validation errors are averaged to select the best value of k. Then the model $\mathcal{M}_k$ fit on the full training set is delivered for the chosen k.

Stepwise Selection

- For computational reasons, best subset selection cannot be applied with very large p.
- *Stepwise* methods, which explore a far more restricted set of models, are attractive alternatives to best subset selection.

Forward Stepwise Selection

- Forward stepwise selection is a computationally efficient alternative to best subset selection.
- Forward stepwise selection begins with a model containing no predictors, and then adds predictors to the model, one-at-a-time, until all of the predictors are in the model.
- In particular, at each step the variable that gives the greatest additional improvement to the fit is added to the model.

Forward Stepwise Selection

- Unlike best subset selection, which involved fitting 2^p models, forward stepwise selection involves fitting one null model, along with $p - k$ models in the kth) iteration, for $k = 0, \ldots, p - 1$.
- This amounts to a total of $1 + \sum_{k=0}^{p-1}(p - k) = 1 + p(p + 1)/2$ models.
- If $p = 20$, best subset selection requires fitting 1,048,576 models, whereas forward stepwise selection requires fitting only 211 models.

Backward Stepwise Selection

- Like forward stepwise selection, *backward stepwise* selection provides an efficient alternative to best subset selection.
- However, unlike forward stepwise selection, it begins with the full least squares model containing all p predictors, and then iteratively removes the least useful predictor, one-at-a-time.

Backward Stepwise Selection

- Like forward stepwise selection, the backward selection approach searches through only $1 + p(p+1)/2$ models, and so can be applied in settings where p is too large to apply best subset selection.
- Backward selection requires that the number of samples n is larger than the number of variables p (so that the full model can be fit). In contrast, forward stepwise can be used even when $n < p$, and so is the only viable subset method when p is very large.

Hybrid Approaches

- Another alternative, hybrid versions of forward and backward stepwise selection are available, in which variables are added to the model sequentially, in analogy to forward selection.
- However, after adding each new variable, the method may also remove any variables that no longer provide an improvement in the model fit.
- Such an approach attempts to more closely mimic best subset selection while retaining the computational advantages of forward and backward stepwise selection.

Moving Beyond Linearity

- Linear models are relatively simple to describe and implement, and have advantages over other approaches in terms of interpretation and inference. However, standard linear models can have significant limitations in terms of classification power.
- We can improve upon least squares using ridge regression, the lasso, principal components regression, and other techniques. In that setting, the improvement is obtained by reducing the complexity of the linear model, and hence the variance of the estimates.
- But we are still using a linear model, which can only be improved so far!

Moving Beyond Linearity

- *Polynomial regression* extends the linear model by adding extra predictors, obtained by raising each of the original predictors to a power. For example, a *cubic* regression uses three variables, X, X^2, and X^3, as predictors. This approach provides a simple way to provide a nonlinear fit to data.
- *Step functions* cut the range of a variable into K distinct regions in order to produce a qualitative variable. This has the effect of fitting a piecewise constant function.

Moving Beyond Linearity

- *Regression splines* are more flexible than polynomials and step functions, and in fact are an extension of the two. They involve dividing the range of X into K distinct regions. Within each region, a polynomial function is fit to the data. However, these polynomials are constrained so that they join smoothly at the region boundaries, or *knots*. Provided that the interval is divided into enough regions, this can produce an extremely flexible fit.

Moving Beyond Linearity

- *Smoothing splines* are similar to regression splines, but arise in a slightly different situation. Smoothing splines result from minimizing a residual sum of squares criterion subject to a smoothness penalty.
- *Local regression* is similar to splines, but differs in an important way. The regions are allowed to overlap, and indeed they do so in a very smooth way.
- *Generalized additive models* allow us to extend the methods above to deal with multiple predictors.