An Introduction to Statistical Learning - Notes

Matheus Rosso

Contents

1 Statistical learning

1.1 Statistical learning concepts

- Statistical learning refers to a set of methods and techniques from which it is possible to understand data from the recognition of its patterns.
- Datasets are decomposed into a **response variable**, Y (though, it can be exist more than one), and a set of **predictors**, $X = (X_1, X_2, ..., X_p)$. These variables correspond to n observations, or data points.
- From those variables, it is conceived a model relating the response variable with the predictors, supposing that the variations in these explain the variations of that:

$$Y = f(X) + \epsilon \tag{1}$$

  Where $f(.)$ is a deterministic function relating the inputs (predictors) with the output (response), and ϵ is a random error term. Usually, it is supposed that ϵ has zero mean and is statistically independent from X.

- The main purpose of statistical learning is to use observational data to estimate the systematic relationship between X and Y, $f(.)$.
- Understanding the patterns that rely on a dataset and estimating $f(.)$ are oriented to **predict** values of Y and to produce **inference** for the relationships among predictors and response.
- Given values for X, a **prediction** for Y follows from:

$$\hat{Y} = \hat{f}(X) \tag{2}$$

  - The accuracy of the point estimate $\hat{Y}$ depends on two types of error: the **reducible error**, related with how well a statistical learning method learns from the data; and the **irreducible error**, that occurs even if the precise function $f(.)$ is known − the irreducible error follows from the existence of the random term ϵ in (1).
  - The distinguishing among reducible and irreducible errors are summarized by the following expression for the expected value of the squared difference between Y and the prediction $\hat{Y}$:

$$E[(Y - \hat{Y})^2] = E[(f(X) + \epsilon - \hat{f}(X))^2] = E[(f(X) - \hat{f}(X))^2] + Var(\epsilon) \tag{3}$$

Where $E[(f(X) - \hat{f}(X))^2]$ is the reducible error, whose minimization is the objective of statistical learning methods; and $Var(\epsilon)$ is the irreducible error, upper bound for the prediction of Y.

- **Inference** refers to the identification of the way Y is affected by X_1, X_2, ..., and X_p. Some questions that follow from an inference problem are:
  - Which predictors are statistically associated with the response?
  - What is the relationship between the response and each predictor (in terms of direction and magnitude)?
  - Can the relationship between the response Y and each predictor be adequately summarized using a linear equation, or is the relationship more complicated?
  - The most adequate statistical learning method to be used depends highly on the purpose of the analysis. If this purpose is derived from a prediction problem or from an inference problem, even a same given method can have different uses. The use of a linear regression model for prediction has several differences in terms of variables selection as compared with its use for inference, for example. Also the bias-variance trade-off has different implications depending on prediction or inference is the main goal. The exclusion of a statistically insignificant variable or a multicollinear variable to avoid overfitting in a prediction setting may not apply if one is interested in a partial effect of a given explanatory variable.
- The estimation of $f(.)$ follows from a **training dataset** containing n data points, $(y_i, x_i)_{i=1}^n$, where $x_i = (x_{i1}, ..., x_{ip})$ and both y_i and x_i are realizations of the populational variables Y and X. A method designed to learn from the training data can be of two main kinds: parametric and non-parametric.
  - The **parametric approach** defines a specific functional form for $f(.)$, which depends on arguments X and parameters. The next step of this approach is to use the observational data to estimate the unknown parameters.
  - The main advantages of the parametric approach are its relatively easy estimation procedure and its propensity to immediate interpretation of the parameters. A clear disadvantage is the inflexibility of supposing a specific functional form for $f(.)$ that may not be the correct

one, even though more parameters can be introduced to the model in order to accommodate different possible functional forms (however, this larger number of parameters may lead to an overfitting problem).

  - The **non-parametric approach** do not define a functional form for $f(.)$, thus allowing for a wide set of possible shapes for the systematic relationship between Y and X. This requires a large number of observations in order to obtain an accurate estimate for $f(.)$. Again, another issue concerning non-parametric estimations is the risk of incurring in **overfitting** as the fit to the data became excessively dependent on data points that appear in the training dataset, but that are not likely to also be present in test datasets.

- A **flexible** statistical learning method can fit the data considering several different shapes for the function $f(.)$. Then, a more flexible model tend to produce more accurate predictions. A more restrictive, or less flexible model, in its turn, has the advantage of being more interpretable. Therefore, there is a **trade-off between flexibility and interpretability**, or between prediction accuracy and model interpretability. So, prediction problems are prone to demand the use of more flexible models, while inference problems may require more restrictive models. Besides, even for predictions a less flexible method can be preferred, as highly flexible models can suffer from overfitting. Figure 1.1 indicates how the trade-off applies for some statistical learning methods.

- Statistical learning methods also vary according with the use of a response variable when the model is trained. A **supervised learning method** uses both Y and X to train/fit the model, as there exists a response variable beforehand. An **unsupervised learning method** do not consider a response variable when the model is estimated, mainly because there is not an output variable previously to the learning process. The most common class of method for unsupervised learning is clustering, when data are grouped following patterns of resemblance in the predictors vector X.

- Finally, another distinction among statistical learning methods considers the nature of the response variable Y. In a supervised learning setting with Y being a quantitative variable, a **regression problem** is defined. Alternatively, when Y is qualitative (or, categorical), a **classification problem** emerges.

FIGURE 2.7. *A representation of the tradeoff between flexibility and interpretability, using different statistical learning methods. In general, as the flexibility of a method increases, its interpretability decreases.*

Figure 1.1: Trade-off between interpretability and flexibility for some methods

1.2 Assessing model accuracy

- Considering a regression problem, a first measure to assess the quality of fit is the **Mean Squared Error (MSE)** statistic:

$$MSE = (1/n)\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2 \tag{4}$$

It is important to stress that (4) is a *training MSE*, i.e., it applies for training data only, thus representing a measure for quality of *fit*. Besides, a large number of methods are implemented in order to minimize a measure like that. Therefore, a more relevant measure is the squared of the difference between an unseen observation y_0 and the prediction $\hat{f}(x_0)$ produced by the model:

$$Ave[(y_0 - \hat{f}(x_0))^2] \tag{5}$$

This measure is called a **test set MSE**, and a good statistical learning method should minimize its value. Even though one could think that a model that produces a low training MSE would also produce a low test MSE, nothing statistically guarantees that (see page 32 for more on this).

- The independence between train MSE and test MSE relates to the **overfitting problem**. The more flexible a model is, the better the fit to the observed data it produces, and the lower the train

MSE associated. However, an estimated $\hat{f}(.)$ function that gets very close to the data points (Y) follows excessively the error term component (ϵ) and, therefore, does not adjust well to the actual systematic relationship between X and Y, which is represented by the actual function $f(.)$. Since the error term is random, for an unseen observation (y_0, x_0), a better prediction is produced by a model that fits well the function $f(.)$, which composes y_0 in a deterministic way, thus implying in a larger probability of $\hat{f}(x_0) \approx y_0$ (and a lower test MSE).

- Therefore, very flexible models tend to have very low training MSE, but large test MSE. This is an indication of **overfitting**, suggesting that the flexibility of the model may have passed the optimal point in which test MSE is minimized. Figure 1.2 illustrates the relations among model flexibility and train and test MSE measures.

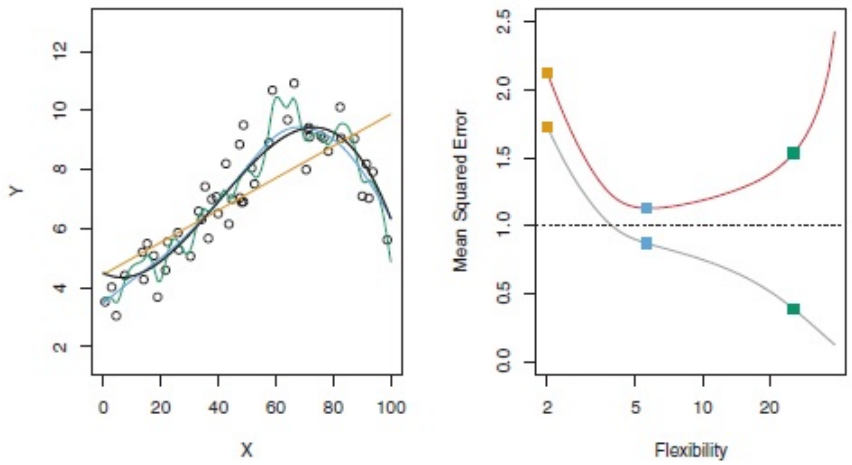

FIGURE 2.9. Left: *Data simulated from f, shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves).* Right: *Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed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

Figure 1.2: Flexibility, train MSE and test MSE

- The **bias-variance trade-off** explains the U-shape of the test set MSE when plotted against the flexibility of a model, therefore, it explains the occurrence of overfitting. The expected test MSE for a given value x_0 can be decomposed as follows:

$$E[(y_0 - \hat{f}(x_0))^2] = Var(\hat{f}(x_0)) + Bias(\hat{f}(x_0))^2 + Var(\epsilon) \tag{6}$$

This average refers to repeating the estimation of $f(.)$ and the prediction of y_0 for different training sets. An overall expected test MSE can be calculated by averaging (6) for all possible values of x_0 in the test set.

- The goal of any statistical learning method is to minimize both the variance of the estimation ($Var(\hat{f}(x_0))$) and the bias, so that the accuracy of the prediction would be the largest possible (with $E[(y_0 - \hat{f}(x_0))^2]$ minimized, given that $Var(\epsilon)$ is irreducible).
- The variance $Var(\hat{f}(x_0))$ computes the expected variation of the estimation of $f(.)$ that would result from different training datasets. More flexible models are expected to have larger variances of $\hat{f}(.)$, since, as discussed previously, they focus excessively on data points Y that are composed also of random error terms, and then changing the observations from the training set is likely to produce large variations in the estimations.
- The bias of the prediction is derived from the fact that the estimation will not precisely get to right functional form or parameters of the actual function $f(.)$. More flexible models tend to have less bias, given that they capture more patterns in the data.
- The bias-variance trade-off shows that it is easy to obtain a model with low bias but that has a high variance, as well as a model with low variance but high bias.
- Moreover, it is expected that from a low level of flexibility, increasing it reduces the bias more than increases the variance, then implying in an initially decreasing test MSE. This occurs until an optimal flexibility point is reached (Figure 1.2).

- When Y is qualitative, instead of a measure such as MSE, an **error rate** is the way to assess the quality of fit:

$$(1/n)\sum_{i=1}^{n} I(y_i \neq \hat{y}_i) \tag{7}$$

Where $I(.)$ is the indicator function. Therefore, the expression (7) consists on the fraction of incorrect classifications produced by the method that led to $\hat{y}_i$. Again, the focus must rely on the expected test error rate for an unseen observation y_0:

$$Ave[I(y_0 - \hat{y}_0)] \tag{8}$$

A good classifier should minimize (8).

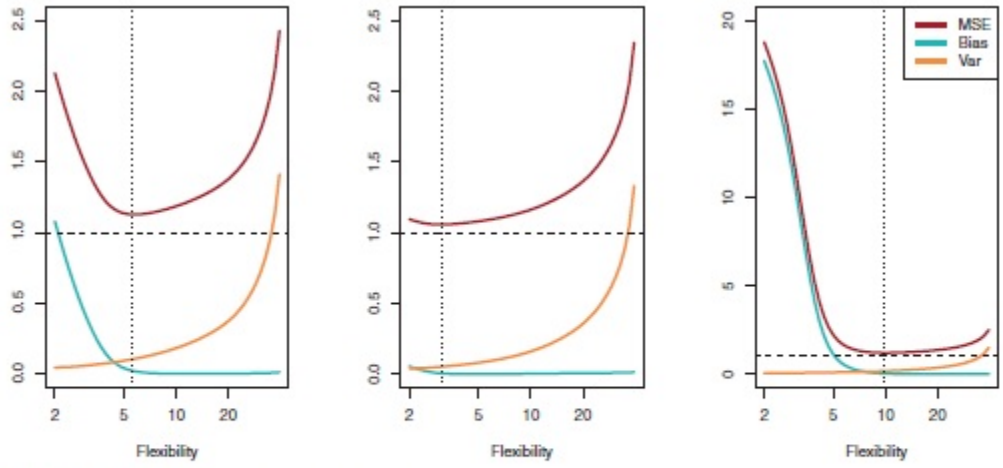

FIGURE 2.12. *Squared bias (blue curve), variance (orange curve), Var(ϵ) (dashed line), and test MSE (red curve) for the three data sets in Figures 2.9–2.11. The vertical dotted line indicates the flexibility level corresponding to the smallest test MSE.*

Figure 1.3: Bias-variance trade-off

– **Bayes classifier:** it is a simple classifier that minimizes (8) by definition. Then, the lower bound for (8) is named **Bayes error rate**. The Bayes classifier assigns $Y = j$ conditional on $X = x_0$ if $P(Y = j|X = x_0)$ is the largest among all possible categories of Y. By doing so, the error rate is necessarily $1 - \max_j P(Y = j|X = x_0)$. Again, an overall test error rate can be calculated by averaging (8) for all possible values of x_0 in the test set.

– **K Nearest Neighbors classifier (KNN classifier):** since the conditional distribution of Y is unknown, many classification methods try to estimate $P(Y = j|X)$. KNN classifier is one of such methods, and rely on the definition of the K nearest neighbors for a data point x_0. If N_0 is the set of those closest points, then $P(Y = j|X = x_0)$ is estimated by:

$$(1/K) \sum_{i \in N_0} I(y_i = j) \tag{9}$$

The prediction associated with (9) is $\hat{y}_0 = j^*$ for which $j^* = \max_j (1/K) \sum_{i \in N_0} I(y_i = j)$.

– The choice of K is determinant for the classification obtained through the KNN classifier. The model is the most flexible when $K = 1$, implying in results highly dependent on a single data point. As K increases, the method becomes more restrictive and converges to a linear decision boundary. As occurs with the regression problem, increasing the flexibility of the model

consistently reduces the train error rate, but reduces the test error rate only until certain level of $(1/K)$.

- As KNN takes the distance between observations concerning their predictors' values, it may be appropriate to standardize the predictors so that scale and unit of measurement does not influence the classification.

2 Linear regression

2.1 Uses for the linear regression model

- Linear regression can be used for predicting a numerical response variable and is particularly useful for inference. Besides, it constitutes a starting point for many advanced techniques.
- Some questions that can be answered by a linear regression model:
  - Is there a relationship between the response and the predictors? If there exists, how strong is it?
  - What are the predictors that effectively contribute to explain the variations in the response variable?
  - What is the partial effect of each predictor over the response variable, and how accurate one can estimate it?
  - For given values of the predictors, what is the prediction for the response, and how accurate is this prediction?
  - Is the relationship between response and predictor linear?
  - Is there an interaction effect among any of the predictors?

2.2 Simple linear regression

- A simple linear regression is composed from a response variable Y and a single predictor X, and defines a specific shape for the systematic relationship between those two variables:

$$Y = \beta_0 + \beta_1 X + \epsilon \tag{10}$$

Implying that $f(X) = \beta_0 + \beta_1 X$, thus resulting in a linear functional form for $f(.)$, where this linearity refers to the relationship between Y and X and mostly to the way how the predictor is related with the parameter β_1.

- Since β_0 and β_1 are unknown, they must be estimated from training data $(y_i, x_i)_{i=1}^n$, where y_i and x_i consist on observations from Y and X, respectively. The method used for estimating β_0 and β_1 is the (ordinary) least squares, which finds $\hat{\beta}_0$ and $\hat{\beta}_1$ such that $y_i \approx \hat{\beta}_0 + \hat{\beta}_1 x_i \ \forall \ i \in \{1, ..., n\}$ as much as possible, i.e., $RSS = \sum_i (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2$ should be minimized with $(\hat{\beta}_0, \hat{\beta}_1)$.

- In estimating the true **population regression line** given by equation (10), a variance emerges as different estimates of β_0, β_1, and even ϵ, follow from different training datasets. Consequently, it is important to check the accuracy of the coefficient estimates.

  - If the usual hypothesis that X and ϵ are (mean) independent sustains, then the least squares estimates for β_0 and β_1 will be **unbiased**. Even with an unbiased estimation of the coefficients, they will always be different from their true values. But what really matters is that the estimations do not systematically overestimate or underestimate the populational parameters, and this is guaranteed by an unbiased estimator.

  - How far away from β_0 and β_1 the estimations tend to be indicates its accuracy. This is measured through the **standard errors** of $\hat{\beta}_0$ and $\hat{\beta}_1$ ($SE(\hat{\beta}_0)$, $SE(\hat{\beta}_1)$), which are usually calculated from a previous estimation of $Var(\epsilon) = \sigma^2$ through the RSE measure, the **residual standard error**, given by $RSE = \sqrt{RSS/(n-2)}$. It is worth notice that this relies on a hypothesis of constant variance for the error term (**homoscedasticity**).

  - From $SE(\hat{\beta}_0)$ and $SE(\hat{\beta}_1)$, one can calculate **confidence intervals** and proceed to **hypothesis testing**, mainly to check the statistical significance of the estimates.

- Besides the accuracy of the individual estimated parameters, it is also important to assess the accuracy of the model as whole, i.e., to quantify how well the estimated model fits to the data. Two main measures exist for this purpose: the previously mentioned residual standard error and the R^2 statistic.

  - The irreducible error $Var(\epsilon)$ implies that even knowing the true β_0 and β_1 it is not possible to perfectly predict y_0 from x_0. This applies even further when $f(.)$ has to be estimated. So, if

$Var(\epsilon)$ indicates the average amount that the response will deviate from the true population regression line, the **residual standard error** (RSE) measure shows the model lack of fit and calculates the aggregate expected distance of the least squares line to the actual data.

– If the RSE is measured in units of Y, the R^2 statistic has the advantage of being a percentage. It calculates the proportion of variation of Y that is due to predictors X. Therefore, $R^2 = (TSS - RSS)/TSS = 1 - (RSS/TSS)$, where RSS is the residual sum of squares and TSS is the total sum of squares, $TSS = \sum_i (y_i - \overline{y})^2$. Since $0 \leq R^2 \leq 1$, as closer R^2 gets to 1, the better is the model fit to the data. However, a low R^2 can mean either that the predictors explain few variations of Y or that $Var(\epsilon) = \sigma^2$ is high as a consequence of a natural high variance of the error term. Thus, R^2 statistics of models for different responses should not be compared.

2.3 Multiple linear regression

- The response variable Y usually can be explained by several different *observable* predictors $X = (X_1, ..., X_p)$. Since the partial effects parameters of each predictor X_j are more precisely estimated when all of the conceivable and observable predictors are present in a same model, the multiple linear regression model is a better choice than the simple linear regression for most empirical tasks that would use such a method. So, the goal in regression problems is to estimate equations as:

$$Y = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p + \epsilon \tag{11}$$

- The equation (11) can as well be estimated through the (ordinary) least squares method. But now $\hat{\beta}_j$ is an estimate of the partial effect of X_j on Y, given that all other X_{-j} are kept constant as an one unit variation in X_j is simulated. Besides, instead of a least squares regression line, the estimation of (11) implies in a least squares regression *plane*.

- The partial effect nature of the parameters in (11) explains why variables that are statistically significant in simple regression models for Y can be non-significant in a multiple linear regression model. One variable can have correlation with Y, but as a result of correlation with another predictor that effectively causes variations in Y.

- Important questions to address in a multiple linear regression setting:

– *Is there a relationship between the response and the predictors?* This can be checked through a F test with the following null hypothesis:

$$H_0 : \beta_1 = \beta_2 = ... = \beta_p = 0 \tag{12}$$

If (12) is rejected, then there is evidence of statistical significant relationship between the response and at least one of the predictors. It is important to stress that parameters highly significant individually are not indicative of joint significance. Finally, if n is small relative to p, then the F test may not be satisfactory to assess joint significance.

– *Variables selection:* defining which predictors should be part of the model is crucial, since including non-relevant variables may lead to an overfitting problem. Only predictors that effectively contribute for fitting the data without increasing excessively the variance of the estimations should be included in the model.

  * One possible proceeding for selecting variables for the multiple linear regression involves estimating several different models each with a different subset of predictors. Then, measures as AIC, BIC, adjusted R^2 should be calculated and compared across the models.
  * Another approach is named **forward selection**. For p possible predictors, p simple models are regressed and the predictor associated with the regression with the lowest RSS is selected. Then, a second variable to be chosen is such that the RSS of the 2-predictors multiple regression model is the lowest. The proceeding continues until some stopping rule is reached.
  * The **backward selection** starts estimating a multiple regression model with all available predictors. Then, the variable with the highest p-value is excluded. The model with $(p-1)$ predictors is estimated and again the variable with the highest p-value is removed from the model. This process ends when all remaining variables have a p-value below some predefined threshold.
  * **Mixed selection:** variables are added progressively into the model, as with the forward selection, but when a previously included predictor has its p-value increased above some threshold, it is removed from the model. This continues until all the variables in the model have a sufficiently low p-value.

– *Model fit:* the main measures for assessing the quality of a multiple linear regression model to fit the data are the residual standard error (RSE) and the R^2 statistic, as it was the case

for the simple regression. Both measures can be used to check whether the inclusion of a particular predictor could contribute to overfit the model. For example, if the inclusion of X_j implies in a very small increase in the R^2 statistic, then a more accurate estimation can be produced without that predictor. The analysis of changes in the adjusted R^2 can help with this conclusion. Therefore, an overfitting problem can be avoided by considering if the inclusion of a predictor into the model really improves its fit to the true population regression plane.

– *Prediction:* given estimations for the populational parameters, $\hat{\beta}_0$, $\hat{\beta}_1$, ..., $\hat{\beta}_p$, a prediction for an unseen observation x_0 is given by:

$$\hat{y}_0 = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + ... + \hat{\beta}_p x_{ip} \tag{13}$$

Predicting using (2) involves two types of uncertainty: one that relates to the error of estimating the true function $f(.)$ (a reducible error), and other that concerns the fact that even knowing the true parameters, predictions would imply in errors because of the existence of the random error term. (an irreducible error). In order to account for the first type of uncertainty, one can calculate **confidence intervals** for the predictions. Account for the second type involves the construction of **prediction intervals**, which are always wider than the first as a consequence of also considering the irreducible error.

∗ A third source of uncertainty in the estimation and prediction procedures refers to the possibility of mistakenly defining $f(.)$ to have a linear form.

2.4 Other considerations in the regression model

- **Qualitative predictors:** when a predictor is a categorical variable with more than 3 levels, the test for the existence of difference in the (conditional) mean of Y among the existing categories should be done through a joint significance test over the parameters of the dummies created.

- The linear regression model usually supposes that the relationship between Y and X is additive and linear. While the first implies that the partial effect of X_j does not depend on the values of other predictors, the second results in constant partial effects. The introduction of **interaction terms**, $X_j.X_k$, is an alternative to relax the hypothesis of additivity. In order to make the functional form of $f(.)$ more flexible, it is possible to introduce polynomial terms of a predictor X_j into the model. The

order of the polynomial should be defined with caution, to avoid the occurrence of the overfitting problem.

- The linear regression model and its procedures rely on some hypothesis that may not apply when a training dataset is used to estimate the model parameters. Some of the problems that can emerge are listed below.
  - **Non-linearity of the data:** if the true relationship between Y and X is not linear, the estimations will be highly biased. A procedure to identify this mistake in the definition of the functional form of $f(.)$ is to analyze the *residual plot*. When the model is correctly defined, plotting the residuals $\hat{e}_i = y_i - \hat{y}_i$ against the fitted values $\hat{y}_i$ produce no identifiable pattern. Therefore, in the case of $\hat{e}_i$ being related with $\hat{y}_i$ in some deterministic way, it may be appropriated to estimate a polynomial regression with the order revealed by the residual plot.
  - **Correlation of the error terms:** the estimation of $Var(\epsilon) = \sigma^2$ and, therefore, of $SE(\hat{\beta}_j)$ will be biased and will underestimate the actual values of the standard errors when there exists correlation among the error terms. Thus, all statistical inference procedures are invalid in this context (besides, the least squares estimation will not be efficient). Again, plotting $\hat{e}_i$ against $\hat{y}_i$ can indicate the violation of the hypothesis of uncorrelated error terms: any pattern suggesting persistence in the series of residuals is an evidence of serial correlation. Besides some techniques for modelling this correlation, one correction procedure is to estimate a robust variance/covariance matrix for the least squares estimates.
  - **Heteroskedasticity:** if $Var(\epsilon_i) = \sigma^2$ is not constant across i, then the calculation of RSE and $SE(\hat{\beta}_j)$ will be incorrect, also making invalid all usual statistical inference procedures. The heteroskedasticity is identified through statistical tests and visually by plotting the residuals $\hat{e}_i$ (or even $\hat{e}_i^2$) against $\hat{y}_i$ (again, if some pattern emerges, it may be a sign of violation of the constant variance assumption). To solve this issue, modelling the variance of ϵ may produce more efficient estimates (weighted least squares, GLS). Applying concave transformations over the response variable can also help to attenuate the effects of heteroskedasticity. Finally, the estimation of a heteroskedastic-robust variance/covariance matrix (White variance matrix) can turn the statistical inference procedures valid again.
  - **Outliers:** an outlier is an extreme value for the response variable Y. Since the predicted value for Y is far from the actual value when the observation is an outlier, the residual standard

error (RSE) will be inflated for the model estimated with the outlier. The inference, therefore, is affected, even though the estimates are not prone to vary considerably with and without the outlier. An adjusted residual plot may be used to distinguish an outlier from an ordinary observation: the *studentized* residuals ($\hat{e}_i$ divided by the RSE) should be plotted against $\hat{y}_i$, and possible outliers are those for which the studentized residual is greater than 3 in absolute value.

- **High leverage points:** a high leverage point is an extreme value for a predictor X_j. Differently from outliers, a high leverage point can substantially distort the estimated model, thus justifying an identification and later exclusion more than the case of an outlier. It should be calculated a *leverage statistic* for each observation i whose mean is $(p+1)/n$, which implies a rule for detecting a high leverage point: if the statistic greatly exceeds $(p+1)/n$, then the corresponding observation may be treated with caution.
- **Collinearity:** there is collinearity when a pair of predictors are correlated to each other, and multicollinearity when one predictor is a function (though not deterministic) of a set of other predictors. Intuitively, this amplifies the variance of the partial effects because it makes harder to isolate the variations of one variable from those of the other predictors. By increasing the standard errors, multicollinearity distorts the results of the inference procedures. The power of the individual statistical significance tests (probability of correctly rejecting the null hypothesis), for example, can be considerably reduced. Besides the analysis of the correlation matrix for the predictors, the calculation of *variance inflation factors* (VIF) is an alternative. If $VIF(\hat{\beta}_j)$ is greater than 5 or 10, then X_j can be highly dependent of the other predictors. As for outliers and high leverage points, dropping a variable that is collinear to other is an option. In fact, an easy way to incur in overfitting is to use an excessively large set of predictors such that some of these are highly collinear to other variables, thus being redundant for the fitting.

2.5 Comparisons of linear regression with KNN

- Differently from the linear regression and other parametric methods, a non-parametric method does not define a functional form for $f(.)$. Therefore, this class of statistical learning methods are inherently more flexible.
- The **K nearest neighbors regression (KNN regression)** method is very similar to the KNN

classifier. Given a value for K and a prediction point x_0, the KNN regression method defines a set N_0 with the K closest points to x_0 in the training dataset. Then, a prediction for y_0 is produced as follows:

$$\hat{f}(x_0) = (1/K) \sum_{i \in N_0} y_i \tag{14}$$

With $K = 1$, the KNN regression method is the most flexible, thus relying on just one data point and resulting in a fit with low bias but high variance.

- A parametric approach as linear regression will outperform a non-parametric approach as KNN regression when the true functional form of $f(.)$ is close to linear. Moreover, a KNN method with small value of K will perform even worse than a more restricted KNN method when $f(.)$ is roughly linear.

- True shapes of the relationship between response and predictors that are non-linear tend to imply in better performances for KNN method than linear regression. However, when there is a very large set of predictors, the **curse of dimensionality** makes linear regression perform better even with non-linear populational relationship between Y and X. This phenomenon is due to the absence of effectively near neighbors for x_0 when there is a large number of predictors available. Therefore, parametric methods are preferable when there is a relatively small number of observations per predictor.

3 Classification

3.1 The classification problem

- When prediction or inference is to be conducted having as reference a qualitative response variable, a *classification problem* is established. Considering an objective of prediction, a statistical learning method should define $\hat{y}_0 = j$ for an observation x_0, where j is selected among a set $\{1, 2, ..., J\}$. Even so, some classification methods first estimate a probability $P(Y = j|X)$ previously of selecting a specific j, thus implying in regression-like procedures. The main classifiers available are logistic regression, linear discriminant analysis, K nearest neighbors and other more computer-intensive methods.

- As for regression models, a set $\{(y_i, x_i\}_{i=1}^n$ of training observations is used to estimate a classification model.

- The linear regression model is not appropriate for classification problems when the categorical response variable has more than 2 levels. This because the numerical encoding to convert a categorical variable into a numerical one would result in a possible inadequate ordering of the levels.

- In the case that the categorical response variable has only two levels, the linear regression estimation will be valid, though can result in predicted probabilities outside the interval $[0, 1]$.

3.2 Logistic regression

- The logistic regressions models the probability of the response variable belonging to each possible category. This probability $P(Y = 1|X) = p(X)$, for a binary response, is set to be equal to the *logistic function*, which necessarily relies in the interval $[0, 1]$:

$$P(y = 1|X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}} \tag{15}$$

  Considering only one predictor. The expression in (15) is estimated through the *maximum likelihood* method.

- A relevant measure that follows from (15) is the **odds ratio**:

$$\frac{p(X)}{1 - p(X)} = e^{\beta_0+\beta_1 X} \tag{16}$$

That has the *log-odds* or *logit* expression below as alternative:

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X \tag{17}$$

- Differently from the use of linear regression to model $p(X)$, given the expression (15) for the logistic regression, the partial effect of X on the probability $P(Y = 1|X)$ depends on the level of X, and not only on β_1, even though this parameter provides the sign of the partial effects.

- From the estimation of the parameters β_0 and β_1 in (15), all statistical inference procedures are valid to assess the accuracy of the coefficient estimates and the statistical significance of predictors. Moreover, predictions for the probability $P(Y = 1|X)$ for a given test observation can be made, and from this, a prediction for the categorical response Y can be defined from any decision rule that specifies a threshold for the estimated probability.

- Extending the variables in (15) leads to a multiple logistic regression:

$$P(y = 1|X) = \frac{e^{\beta_0+\beta_1 X+...+\beta_p X_p}}{1 + e^{\beta_0+\beta_1 X+...+\beta_p X_p}} \tag{18}$$

  Again, the maximum likelihood method estimates the parameters in (18).

- Similarly to what applies for multiple linear regression, the inclusion of different predictors in a logistic regression can avoid the occurrence of **confounding factors**, which consists on correlated predictors that must be used together in order to isolate their partial effects.

- Logistic regression has extensions for the case in which the categorical response variable has more than 2 levels (multinomial logit and ordered logit).

3.3 Linear discriminant analysis

- Logistic regression tries to model $P(Y = k|X = x)$ directly. An alternative approach is to first model $P(X = x|Y = k)$ for all $k \in \{1, ..., K\}$ and then use Bayes' theorem to get to $P(Y = j|X = x)$. **Linear discriminant analysis (LDA)** is a method that follows this indirect way to estimate the probabilities of interest.

- LDA is relevant as an alternative to logistic regression since their estimates may be highly unstable when the classes of Y are well-separated and when n is small and the predictors X come from normal distributions.

- **Bayes' theorem and classification:** supposing that the response variable Y has K possible classes to assume, each with *prior probability* π_k. Given predictors X, their *density functions* $f_k(X) = P(X = x|Y = k)$ and those priors lead to the *posterior probability* of interest $P(Y = k|X = x)$ through the Bayes' theorem:

$$P(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)} \tag{19}$$

  In order to simplify notation, it can be defined $p_k(X) = P(Y = k|X)$. To estimate π_l, if a random sample of Y is available, the simple relative frequency of $Y = l$ provides a consistent estimator for the prior π_l. When it comes to estimate $f_k(X)$, more advanced procedures must be used.

- The LDA method supposes that each predictor in X has a normal distribution conditional on $Y = k$. Consequently, $f_k(x)$ is given by the following expression when $p = 1$ (X unidimensional):

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\Big(- \frac{1}{2\sigma_k^2}(x - \mu_k)^2\Big) \tag{20}$$

  Since this distribution of X is conditional on the class assumed by Y, all the parameters $E(X|Y = k) = \mu_k$ and $Var(X|Y = k) = \sigma_k^2$ depends on the class k considered. As discussed further, the *linear* aspect of LDA is due to an assumption of common variance $\sigma_1^2 = ... = \sigma_k^2 = \sigma^2$. Using (20) on (19):

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} \exp\Big(- \frac{1}{2\sigma}(x - \mu_k)^2\Big)}{\sum_l \pi_l \frac{1}{\sqrt{2\pi}\sigma_l} \exp\Big(- \frac{1}{2\sigma}(x - \mu_l)^2\Big)} \tag{21}$$

  Taking the log of (21):

$$\delta_k(x) = x.\frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k) \tag{22}$$

  Given μ_k and σ^2, the Bayes classifier principle proposes that an observation x is assigned to the class k^* for which (22) is the largest.

  - The **Bayes decision boundary** is an expression derived from (22) that geometrically classifies test observations from their positions in the space relatively to the Bayes decision boundary. This expression, therefore, is constructed by setting $\delta_k(x) = \delta_l(x)$ for all $k \neq l$.
  - The LDA method can also be applied from inputting x into (22) and assessing the k^* for which $\delta_{k^*}(x)$ is maximum. However, in practice it is necessary to estimate the unknown parameters μ_k, σ^2 and π_k. If $\hat{\pi}_k$ can be obtained from relative frequencies, $\hat{\mu}_k$ and $\hat{\sigma}^2$ follows the usual

statistical formulas. Once $(\hat{\pi}_k, \hat{\mu}_k, \hat{\sigma}^2)$ is calculated, they must be plugged into (22) in order to produce predictions for the response variable Y.

  - From (22) being a linear function of x follows the term *linear* in LDA. This, in its turn, is a result of the assumption that X has common (conditional) variance along all classes k.

- When $p > 1$ (X composed by multiple predictors), the LDA method supposes a multivariate normal distribution for the vector $X = (X_1, ..., X_p)$ conditional on $Y = k$, $X \sim N(\mu_k, \Sigma)$, where μ_k is a p-dimensional vector of class-specific means and Σ is a pxp variance/covariance matrix common to all classes k. The multivariate version of (22) is given by:

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log(\pi_k) \tag{23}$$

  Again, the Bayes classifier defines for an observation x a class $Y = k^*$ such that (23) is largest. In the same way, a Bayes decision boundary can be defined to geometrically assign a class to an observation x. Regarding the estimation of parameters μ_k, Σ and π_k, it requires the usual statistical formulas for sample means and variances and for relative frequency, but now the covariances present in Σ should also be estimated.

- The classification problem differs from the regression problem by a special circumstance: in the cases where $Y \in \{0, 1\}$ and $P(Y = 1|X)$ is very small (less than 5%, for example), then the *null classifier*, i.e., predicting $Y = 0$ to all test observations will have a test error rate also very small (particularly, equal to $P(Y = 1|X)$). Therefore, it is challenging for any relatively complex statistical learning method to compete with such a trivial classifier.

- Even so, depending on the context of the classification problem, more important than a low overall test error rate is a low class-specific test error rate. This means that mistakenly predicting $Y = 0$ may be more problematic than mistakenly predicting $Y = 1$.

  - This brings into discussion two measures: the **sensitivity** is the percentage of $Y = 1$ correctly classified ($\hat{Y}_1|Y_1/Y_1$, where $\hat{Y}_1|Y_1$ refers to $\hat{Y} = 1$ when $Y = 1$ and Y_1 corresponds to $Y = 1$). On the other hand, the **specificity** is the percentage of $Y = 0$ correctly classified ($\hat{Y}_0|Y_0/Y_0$).

  - Since the Bayes classifier minimizes the overall test error rate by construction, one classification method based on it can have a low overall error rate, but a high class-specific error rate. This may be implied also by the construction of the Bayes classifier, which assigns an observation

x to the class $Y = k^*$ for which $P(Y = k^*|X)$ is largest. In a binary case, this represents the following threshold for assigning x to $Y = 1$:

$$P(Y = 1|X = x) > 0.5 \tag{24}$$

When the fraction of $Y = 1$ is small, such a threshold will be larger than the appropriate, implying in a small sensitivity, and a large proportion of **false negatives** (**type II error**).

– A better class-specific performance can be achieved by reducing the threshold (24). This should reduce the proportion of false negatives and increase the sensitivity, even though at the cost of an increase of the overall error rate by increasing the **false positive** (**type I error**) fraction. This trade-off between false negative proportion and overall error rate is illustrated by figure 3.3.

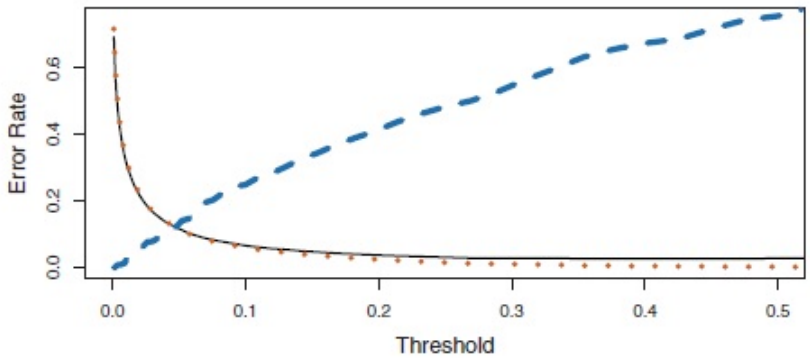

FIGURE 4.7. *For the* `Default` *data set, error rates are shown as a function of the threshold value for the posterior probability that is used to perform the assignment. The black solid line displays the overall error rate. The blue dashed line represents the fraction of defaulting customers that are incorrectly classified, and the orange dotted line indicates the fraction of errors among the non-defaulting customers.*

Figure 3.1: False negative fraction and overall error rate trade-off

– The appropriate threshold depends on domain knowledge and considers the relative costs of incurring in false negatives and false positives.

– A measure that accounts for this inverse relationship between false negative and false positive fractions is the **ROC curve**, which is constructed considering all different thresholds possible. The larger the area under the ROC curve (**AUC**) the better is the classifier. A classifier as good as classifying by chance has an AUC of 0.5 (having test observations as reference).

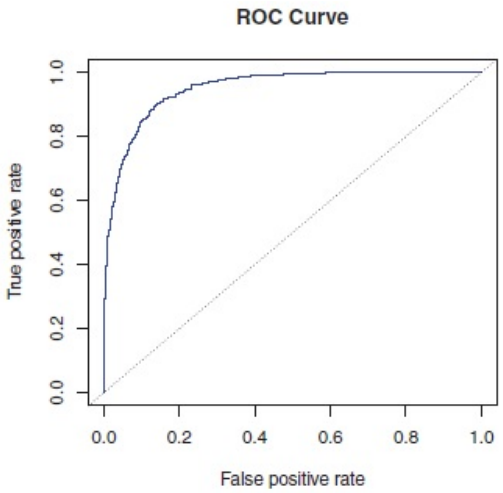

FIGURE 4.8. *A ROC curve for the LDA classifier on the* `Default` *data. It traces out two types of error as we vary the threshold value for the posterior probability of default. The actual thresholds are not shown. The true positive rate is the sensitivity: the fraction of defaulters that are correctly identified, using a given threshold value. The false positive rate is 1-specificity: the fraction of non-defaulters that we classify incorrectly as defaulters, using that same threshold value. The ideal ROC curve hugs the top left corner, indicating a high true positive rate and a low false positive rate. The dotted line represents the "no information" classifier; this is what we would expect if student status and credit card balance are not associated with probability of default.*

Figure 3.2: ROC curve

- The following **confusion matrix** summarizes the most relevant measures concerning classification problems:

True class/Predicted class	**Negative (0)**	**Positive (1)**	**Total**
Negative (0)	True negative (TN)	False positive (FP)	N
Positive (1)	False negative (FN)	True positive (TP)	P
Total	$\hat{N}$	$\hat{P}$	T

Table 3.1: Confusion matrix

- From the components of table 3.3, one can define:
  - False negative (type II error) rate (miss rate): $FN/P = FN/(FN + TP)$.
  - False positive (type I error) rate (fall-out): $FP/N = FP/(FP + TN)$.

- Sensitivity (*recall*): TP/P.
- Specificity: TN/N.
- Positive predictive value (*precision*): $TP/\hat{P} = TP/(FP + TP)$.
- Negative predictive value: $TN/\hat{N} = TN/(FN + TN)$.
- False discovery rate: $FP/(\hat{P}) = FP/(FP + TP)$.
- Accuracy rate: $(TN + TP)/(N + P) = (TN + TP)/(TN + FN + FP + TP)$.
- F1 score: $2TP/(2TP + FP + FN)$.

- Selecting which measure from the above to serve as reference in the modelling procedures depends highly on the context of application. If from the total of positives ($Y = 1$) is harmful to produce false negatives (FN), then maximizing the sensitivity, or minimizing the false negative rate may be an option. But, if from the total positives predicted ($\hat{Y} = 1$) is harmful to get few true positives (TP), then maximizing the positive predictive value (precision) may be preferable.

- **Quadratic discriminant analysis (QDA):** this alternative to LDA follows from the assumption of variance/covariance matrices class-specific, so that $X \sim N(\mu_k, \Sigma_k)$. This implies in altering (23) to:

$$\delta_k(x) = -\frac{1}{2}x^T\Sigma_k^{-1}x + x^T\Sigma_k^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma_k^{-1}\mu_k - \frac{1}{2}\log|\Sigma_k| + \log(\pi_k) \tag{25}$$

  Thus, $\delta_k(x)$ in (25) is a quadratic function of x.

  - Since (25) requires the estimation of a larger number of parameters given by the variances and covariances class-specific, QDA models are more flexible and, therefore, tend to have larger variances and lower bias than it applies for LDA models.
  - LDA is more appropriate than QDA if there is evidence of common variance for X across classes of Y, and when the number of observations is relatively small.

3.4 A comparison of classification methods

- So far four different classification methods were presented: KNN, logistic regression, LDA and QDA. While the first is non-parametric, all the others suppose a functional form for the deterministic relationship between predictors X and response Y.

- Logistic regression and LDA share a common characteristic of linear decision boundaries, which is related to the fact that both have linear odds ratios. Basically, the main differences between logistic regression and LDA rely on the fitting procedures, as the first uses a non-linear method (maximum likelihood) and the second is far more simple, using usual statistical formulas for the relevant parameters. Even with this proximity, LDA tends to outperform logistic regression if the predictors X follow a conditional normal distribution with common variance matrix across all classes.

- Differently from logistic regression and LDA, QDA assumes non-linearity for the deterministic relationship between X and Y, since its Bayes decision boundary is quadratic instead of linear. KNN, in its turn, is even more flexible, given that does not assume any functional form for the decision boundary. Consequently, QDA is a midpoint between parametric linear approaches (logistic regression and LDA) and non-parametric approaches (KNN), providing a wider range of possible functional forms for the true relationship between X and Y.

  - If the predictors are drawn from normal distributions with common variance, irrespective of the correlation between the predictors the LDA method outperforms the others, followed by the logistic regression. KNN would be the worst method in such a restrictive scenario.
  - If the predictors are not normally distributed, but follow a t-distribution, for example, with common variance, then logistic regression is better than LDA as a consequence of the non-normality. Similarly, the absence of normal distribution for the predictors would even imply in QDA performing worse than KNN.
  - If predictors follow a normal distribution, but with class-specific variances, then QDA will be the best approach, followed by KNN given its more flexibility.
  - If predictors are normally distributed, but the responses follow a relatively complex distribution as a function of X_j, X_j^2 and $X_j.X_k$, then QDA would still be the best method, again followed by KNN and with the linear approaches having a particularly bad performance.
  - Finally, if responses Y have very complex distributions, highly non-linear related to X, then KNN is preferable, with QDA being only slightly better than logistic regression and LDA.
  - Therefore, with linear decision boundaries, logistic regression and LDA are better choices for classification. When boundaries are moderately non-linear, QDA may perform better. In a

more extreme case where decision boundaries are very complex in its functional form, a non-parametric approach as KNN can be superior.

– As was the case with linear regression, both logistic regression and LDA have more flexible versions where transformations of the predictors (X_j^2, X_j^3, X_j^4, and so on) are used in the estimations. This may improve the performance of the models if the increase in variance due to the acquired more flexibility is more than compensated by a reduction in bias.

4 Resampling methods

- Given the limitation imposed by having only one sample from the population, resampling methods consists of extracting sub-samples from the entire available sample in order to produce additional information about some model, as different estimations are produced from each sub-sample.

- There are two main resampling methods: *cross-validation* and *bootstrap*. Cross-validation is used to estimate the test error from predictions of a given statistical learning model, either to assess its performance (**model assessment**) and to select the appropriate level of flexibility (**model selection**). Bootstrap is used to obtain estimates for the accuracy of a parameter or model estimation.

4.1 Cross-validation

- From fitting a model, it is straightforward to calculate the training error. Instead of this measure, what is of interest is the test error that indicates how far away from true values tend to be the predictions associated with observations not used when training the model. Training error is eventually very different from test error, and besides is prone to underestimate it.

- Since test observations with their response values usually are not available, it is necessary to estimate test error from the data set that would be used for training the model. In order to reproduce the situation in which prediction accuracy is assessed from observations not used when the model is fitted, the original dataset is split into training sets, used for estimations, and **hold-out** sets, used for predictions and, therefore, for computing test error estimates.

- **Validation set approach:** involves only one random splitting of all observations into a training set and a **validation set**. The first is used to fit the statistical learning method and the second

leads to an estimation of the test error, named **validation set error rate**, as predictions for its observations are opposed to the true responses.

– One possible application of the validation set approach is to estimate models with different levels of flexibility, calculate their validation set errors and plot them against the level of flexibility (figure 4.1). Repeating this procedure several times will produce somewhat different curves for validation set errors. Even though the validation set errors curves have different levels, they tend to evolve similarly as flexibility increases, thus suggesting approximate optimal levels of flexibility.

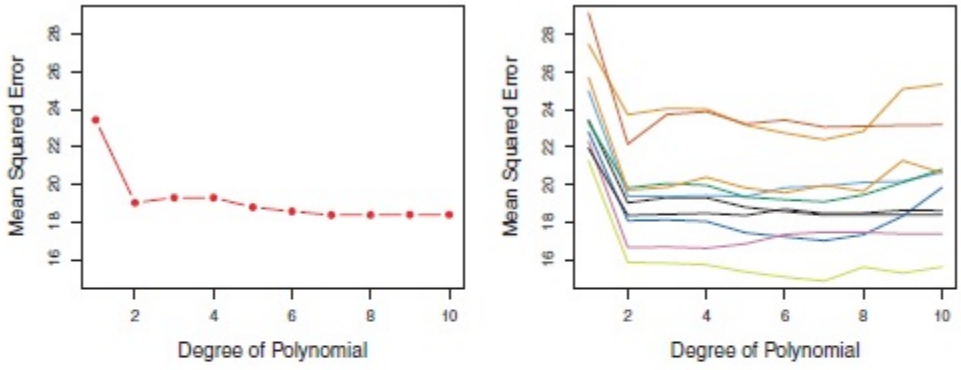

FIGURE 5.2. *The validation set approach was used on the* `Auto` *data set in order to estimate the test error that results from predicting* `mpg` *using polynomial functions of* `horsepower`. Left: *Validation error estimates for a single split into training and validation data sets.* Right: *The validation 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.*

Figure 4.1: Validation set errors and level of flexibility

– Those differences in the levels of validation set errors as the validation approach is repeated show that this procedure for estimating the test error has high variance, since it depends on which observations are included in the training set and which are assigned to the validation set.

– Another issue with the validation set approach is that it tends to overestimate the test error, since statistical methods perform worse when they rely on fewer observations, which is the case as a relatively large share of observations are kept out of the estimation procedure.

- **Leave-one-out cross-validation (LOOCV):** this approach produces n splits and n model fitting, where n is the number of observations of the entire dataset. Each time $n-1$ observations are assigned to the training set and the remaining data point is used for assessing the prediction accuracy, for which a test error estimate $MSE_i = (y_i - \hat{y}_i)^2$ is calculated. This procedure is repeated for each observation i in the dataset.

  – Each individual test error estimate MSE_i is a poor, though unbiased, estimation for the test error, since it depends highly on a single observation, thus having a very large variance. This is attenuated as all MSE_i are averaged into the **LOOCV estimate** for the test error (measured by MSE in the regression problem):

$$MSE_{(n)} = \frac{1}{n}\sum_{i=1}^{n} MSE_i \tag{26}$$

  The estimation provided by (26) has less bias than that given by the validation set approach, since each MSE_i is calculated from a prediction based on a model fitted from all but one observation. Thus, the LOOCV procedure will not overestimate the test error. Besides, there is no randomness in this approach, given that only one curve for LOOCV against model flexibility is produced (figure 4.1).

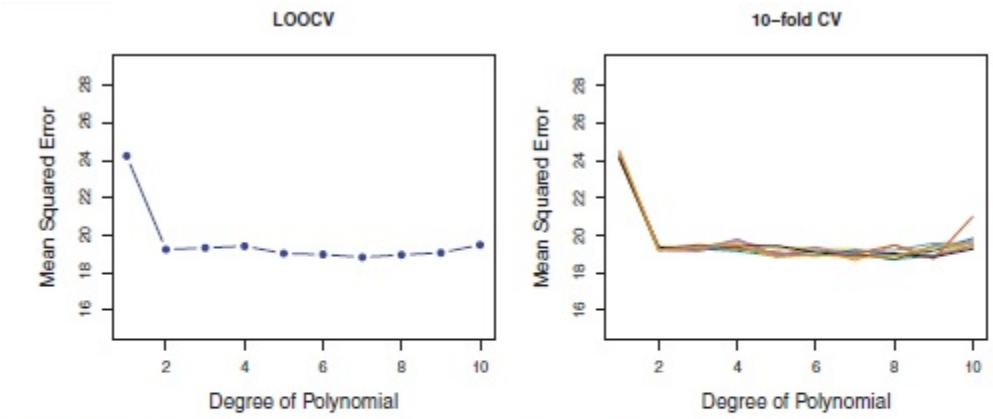

FIGURE 5.4. *Cross-validation was used on the* `Auto` *data set in order to estimate the test error that results from predicting* `mpg` *using polynomial functions of* `horsepower`. Left: *The LOOCV error curve.* Right: 10*-fold CV was run nine separate times, each with a different random split of the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

Figure 4.2: LOOCV and k-fold errors and level of flexibility

- One disadvantage of LOOCV approach to estimate test error relies on the fact that it is computationally expensive, since n different models have to be fitted. However, for the linear regression model, there is a shortcut for calculating (26). Instead of regressing n different models, only one regression can be performed from the entire dataset, followed by the calculation of the expression below:

$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - h_i}\right)^2 \tag{27}$$

  Where $\hat{y}_i$ is the fitted value for the observation i and h_i is the leverage statistic.

- **k-fold cross-validation:** randomly divides the entire dataset into k folds of approximately the same size (k/n), and proceeds k times as follows: one fold with k/n observations is hold out for predictions, while the remaining $k-1$ folds with a total of $(k-1)k/n$ observations are used for fitting the model. The estimated model is used to predict the response for the hold out observations, thus leading to an estimation MSE_i for the test error.

  - The **k-fold estimate** for the test error averages all MSE_i for the k estimations, and consequently sets of predictions, produced:

$$MSE_{(k)} = \frac{1}{k}\sum_{i=1}^{k} MSE_i \tag{28}$$

  - Therefore, k-fold CV has the LOOCV approach as a special case when $k = n$. Empirically, it was defined that $k = 5$ or $k = 10$ implies a k-fold estimate with good properties, at the same time it has a moderate computational cost.
  - Since the approach starts randomly dividing the entire dataset into k folds, the k-fold CV estimate for the test error varies as this proceeding is repeated several times. However, the k-fold CV estimate has lower variance than the validation set approach (figure 4.1).
  - Whether the approach to estimate the test error is validation set, LOOCV or k-fold CV, their errors plotted against model flexibility tend to produce curves with similar format as compared to the true error curve. Therefore, it is expected that those procedures are valid to define the appropriate level of flexibility for a given statistical learning method, or even to choose the best method from a range of options, considering that the minimum of the test error and the minimum of the estimated errors are likely to occur together. Even so, those validation and cross-validation approaches may not produce good estimations of the precise level of the test error, when model performance is the main goal of the validation or cross-validation procedure.

- **Bias-variance trade-off for k-fold CV:** the LOOCV approach produces predictions based on models estimated from larger sets of observations than validation set and k-fold CV. This helps LOOCV estimates for test error to have less bias than those of validation set and k-fold CV procedures. Concerning validation set and k-fold, the later uses more observations than the first, so k-fold is expected to lead to less biased estimates for test error.

  - However, when producing estimates, one is usually not only concerned with bias, but also with variance. k-fold, as mentioned, produces estimates for test error with lower variance than that of validation set approach.
  - Besides, as LOOCV averages estimated test errors that are highly correlated (training sets very similar to each other), its variance is expected to be higher than that of k-fold CV (training sets relatively more distinct to each other), which averages estimated test errors less correlated.
  - Therefore, k-fold CV presents intermediate levels of bias and variance, thus being a more biased, but with less variance alternative for LOOCV. Moreover, $k = 5$ and $k = 10$ are considered parameters for which the estimates for test error are expected to have a moderate amount of bias and a sufficient low variance.

- The above discussion focused on regression problems, even that all conclusions apply for classification problems as well, the only difference being the test error measure of reference, which now is error rate. Therefore, LOOCV estimate for test error rate is given by:

$$CV_{(n)} = \frac{1}{n} \sum_{i=1}^{n} I(y_i \neq \hat{y}_i) \tag{29}$$

  Where a similar expression holds for k-fold CV and validation set approach.

4.2 The bootstrap

- Bootstrap is used to assess the accuracy of a given estimation, whether it refers to an estimated parameter, a whole statistical learning method or even a predicted value.

- In order to assess the variability of an estimation, the bootstrap reproduces the idea under which the outcome of an estimation varies as the sample used changes. Since obtaining new samples from the population is mostly unfeasible, bootstrap produces sub-samples from the available dataset.

- From the whole set of n observations, bootstrap randomly selects *with replacement* n' observations, estimates the parameter, fits the model or predicts values for the response variable based on those n' observations, and then repeat this process B times (for a sufficiently large value for B). Finally, standard errors are calculated from the results of all estimations.

5 Linear model selection and regularization

5.1 Extending the linear regression model

- The linear regression model, even though very restrictive when compared to advanced non-linear models, is particularly useful for inference and may have competitive performance for prediction purposes, besides being more easily interpretable.

- Extending the linear regression model can improve it both reducing the variance of the estimates, so that prediction accuracy is increased, and helping with model interpretability, by selecting some predictors to be part of the model, which eliminates unnecessary complexities.

- There are three main approaches to improve prediction accuracy and interpretability of linear regression models: *subset selection*, where only the best predictors are selected to estimate the model, *shrinkage methods*, which estimate the model with all predictors available, but doing it in such a way that some of their coefficients are constrained towards zero, and *dimension reduction*, which creates a smaller set of linear combinations of all predictors.

5.2 Subset selection

- Subset selection may also be defined as *model selection*, or *variables selection*, since only a subset of the entire set of available predictors is selected to be used in the linear regression model estimation. This simplifies model interpretability as unnecessary variables are removed from the model, attenuating the complexity of the underlying relationship between response and predictors. Besides, this may reduce the variance of the coefficient estimates and, therefore, increase the accuracy of predictions, given that the test error decreases with the exclusion of variables that inflate the variance, but modestly contribute to reduce bias.

- **Best subset selection:** this approach for model selection estimates all possible combinations from

the p predictors available. From all these 2^p models estimated, one is selected as the preferable for each model size (i.e., number of predictors), given its performance when fitting the data, and then the best among all model sizes is selected considering its performance predicting the response for hold-out observations.

  - **Procedure for best subset selection:**

    1. The null model M_0 is estimated, considering only an intercept for the equation.
    2. For each $k \in \{1, 2, ..., p\}$ and considering the whole set of predictors $X = (X_1, X_2, ..., X_p)$, all models with k predictors are estimated. The best model for each model size k is selected, thus being defined as M_k, when the criterion for this selection involves the smallest RSS or the largest R^2.
    3. The best model among M_0, M_1, M_2, ..., M_p is selected considering cross-validation estimated error, C_p, AIC, BIC or adjusted R^2.

  - The best subset selection applies equivalently for statistical learning procedures other than linear regression, the difference being in the metrics for selection both in step 2 and step 3. For logistic regression, for example, instead of RSS or R^2 in step 2, the deviance statistic ($-2\mathcal{L}_{ML}$) can be used. In step 3, each statistical learning method considered has its own formulations for C_p, AIC, BIC and adjusted R^2. Besides, cross-validation estimated error must also varies according with the context of estimation.

  - The main advantages of best subset selection is that it is conceptually simple and searches among all possible specifications of the model under construction. This precise fact implies in its main disadvantage, the computational complexity involved in estimating 2^p models.

- **Stepwise selection:** besides its computation complexity, best subset selection may also suffer from statistical problems when p is large. The larger the space where the approach is searching over appropriate models in step 2, the larger the chance for selecting models that perform well only in training data, but poorly when new observations are applied to the model. Therefore, best subset selection are prone to suffer from overfitting when p is very large, thus implying in estimates highly variable and predictions with few accuracy.

- **Forward stepwise selection:** consists on the first stepwise approach for variable selection. As best subset selection, it starts by estimating a null model, which implies, in this case, in the possibility

of specifying models when $n < p$. Then, predictors are added to the model sequentially given their contribution for fitting the data.

  - **Procedure for forward stepwise selection:**

    1. The null model M_0 is estimated with only an intercept in the equation.
    2. For each $k \in \{1, 2, ..., p-1\}$, all models that increases M_k in one additional predictor are estimated, and that one with the lowest RSS or the largest R^2 is chosen as the best and defined as M_{k+1}.
    3. The best model among M_0, M_1, ..., M_p is selected having as reference the cross-validation estimated error, C_p, AIC, BIC or adjusted R^2.

  - Instead of 2^p models, forward stepwise selection only requires the estimation of only $1 + p(p+1)/2$ models. This smaller computational requirement implies, however, in a uncertainty about whether the set of models M_0, M_1, ..., M_p truly presents the best model for each model size. It can be the case that, for example, the best model with one variable contain the predictor X_1, but the best model with two variables contains X_2 and X_3. The forward selection, by starting with X_1 in this context, will lead to a best model for two variables that necessarily contains X_1.

- **Backward stepwise selection:** this approach starts by estimating the full model M_p, and sequentially removes the variables that less contribute to fit the data.

  - **Procedure for backward stepwise selection**:

    1. The full model M_p is estimated using all available predictors in the equation.
    2. For $k \in \{p, p-1, ..., 1\}$, estimate all k models that contains all predictors present in M_k except for only one. The best model among these k models is selected considering the lowest RSS or the largest R^2, and is defined as M_{k-1}.
    3. The best model among M_0, M_1, ..., M_p is chosen given criteria as cross-validation estimated error, C_p, AIC, BIC or adjusted R^2.

  - Again, a total of $1 + p(p+1)/2$ models are estimated, and also there is no guarantee that the best models for each model size are selected.

- **Hybrid approaches:** as forward selection, additional predictors are sequentially added to the

model, but as with backward selection, those variables that are no longer useful for fitting the data may also be sequentially removed from the model.

- **Choosing the optimal model:** if RSS and R^2 are used to specify models for each model size in step 2 of all variables selection processes discussed above, they can not be seen as alternatives for selecting the best model among all available. Though low RSS and high R^2 indicate a good fit to the data, this means that a low *training* error is achieved. As presented in section 1, training set error is not a good estimate for test set error, which is the true measure for defining if a given model is appropriate for prediction purposes.
  - Statistical learning methods usually produce estimates having as objective the minimization of the training set error. So, they are implemented in order to define the training error to be as small as possible, while test error is calculated from observations which were not part of that minimization process.
  - Moreover, training error always decreases as model flexibility increases, or, as is the case in linear regression setting, training error always decreases as the number of variables increases.
  - This follows from a persistent decrease in bias as the model fit to the data consistently better as more information about the observations is added to the model.
  - While this means decreasing training error, the test error may increase as the *noise information* included in the model does not refer to new observations not present in the training set, which may strongly increase the prediction variance, more than compensating the reduction in bias.
- From two possibilities to estimate test set error, one does so indirectly by adjusting training error measures, whereas the other directly estimates the test error through validation set approach or cross-validation approaches.
- Considering the linear regression model, the least squares method for fitting a linear regression equation to the data by definition defines coefficient estimates so as the RSS is minimized. Besides, the RSS is a decreasing function of the number of variables in a model. As a decrease in RSS always means a decrease in training error, but may not be followed by a decrease in test error, if RSS is to be used as a measure of test error, then it must be adjusted for the model size.
  - The C_p estimate for test set MSE for a linear model fitted by least squares with d predictors

is given by:

$$C_p = \frac{1}{n}(RSS + 2d\hat{\sigma}^2) \tag{30}$$

Where $\hat{\sigma}^2$ is an estimate for the variance of the error term. If $\hat{\sigma}^2$ is unbiased when estimating the true σ^2 parameter, then C_p is unbiased to estimate the true test MSE. Then, one model is best relative to others if (30) is minimum.

- The AIC and the BIC measures for test error MSE are slighly different from the expression in (30):

$$AIC = \frac{1}{n\hat{\sigma}^2}(RSS + 2d\hat{\sigma}^2) \tag{31}$$

$$BIC = \frac{1}{n}(RSS + \log(n)d\hat{\sigma}^2) \tag{32}$$

- While (30), (31) and (32) should be minimized when selecting the best among a set of models, the adjusted R^2 should be maximized:

$$\overline{R}^2 = 1 - \frac{RSS/(n-d-1)}{TSS/(n-1)} \tag{33}$$

- It is clear that all four expressions (30), (31), (32) and (33) penalizes models with a large number of predictors (large d). Therefore, a low training MSE (thus, low RSS) will not necessarily imply in low C_p, AIC, BIC measures, or in a high $\overline{R}^2$ measure, since these statistics introduce a penalty for the inclusion of noise variables, which are not able to significantly reduce RSS.
- There are versions of (30), (31), (32) and (33) that can be applied for statistical learning methods different from linear regression estimated through least squares.

- Though computationally more complex, direct estimates of test error obtained through validation set approach or cross-validation approaches (LOOCV, k-fold CV) rely on fewer assumptions than those necessary to the indirect estimations, which require even that the model is correctly specified.
  - The best model among M_0, M_1, ..., M_p can be identified from a validation set error estimate or a cross-validation error estimate curve which plots the test MSE estimations for each model size against the model size d. Repeating the validation set approach or the cross-validation approaches may imply in different definitions of the best model. Therefore, one possible procedure is to calculate the standard deviation of the test error estimates (for the different model sizes), and then select the smallest model for which the test error estimate is within one standard deviation from the lowest test error estimate among all model sizes.

5.3 Shrinkage methods

- The subset selection approach defines only some of the predictors available to be part of the linear regression model. Increased model accuracy and interpretability may also be achieved through estimation methods different from least squares, where these methods use all predictors available, but constrain the coefficient estimates so that they are driven towards zero, working as if the associated predictor was not included in the model.

- **Ridge regression:** this method has a different objective function to be minimized. Instead of considering only RSS, also a term aggregating all coefficient estimates composes the objective function:

$$\sum_{i=1}^{n}\Big(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\Big)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = RSS + \lambda\sum_{j=1}^{p}\beta_j^2 \tag{34}$$

  The element λ is called **tunning parameter**, being defined separately. The **ridge regression coefficient estimates** $\hat{\beta}_\lambda^R$ minimize the expression (34), and thus not only seek to fit the data as best as possible, but also penalize the absolute magnitude of each coefficient estimate.

  - Therefore, as compared to the least squares coefficient estimates $\hat{\beta}^{OLS}$, the ridge regression method drives the estimates of β_j towards zero.
  - Besides, the expression (34) shows that least squares is a particular case of ridge regression, since $\hat{\beta}^{OLS}$ is obtained when $\lambda = 0$. However, when $\lambda \to \infty$, then $\hat{\beta}_\lambda^R \to 0$ in order to (34) be minimized.
  - Even though all coefficient estimates considered together are driven to zero when λ increases, it can be that some individual estimates get higher for short intervals of λ. This aggregate decreasing of ridge regression coefficient estimates is captured by the l_2-norm of the vector $\hat{\beta}_\lambda^R$. The expression $||\hat{\beta}_\lambda^R||_2 = \sqrt{\hat{\beta}_1^2 + \hat{\beta}_2^2 + ... + \hat{\beta}_p^2}$ always decreases as λ increases, while $||\hat{\beta}_\lambda^R||_2$ tends to $||\hat{\beta}^{OLS}||_2$ when $\lambda \to 0$ (figure 5.3).
  - If the least squares estimate for β_j is divided by c when the corresponding predictor is multiplied by this constant c, the ridge regression coefficient estimates are even more sensible to scale. If x_j is multiplied by c, including estimates for other predictors may change considerably. Therefore, as it was the case for the KNN statistical learning method, also when ridge regression is to be applied one should previously standardize the predictors, so that all variables have the same scale.

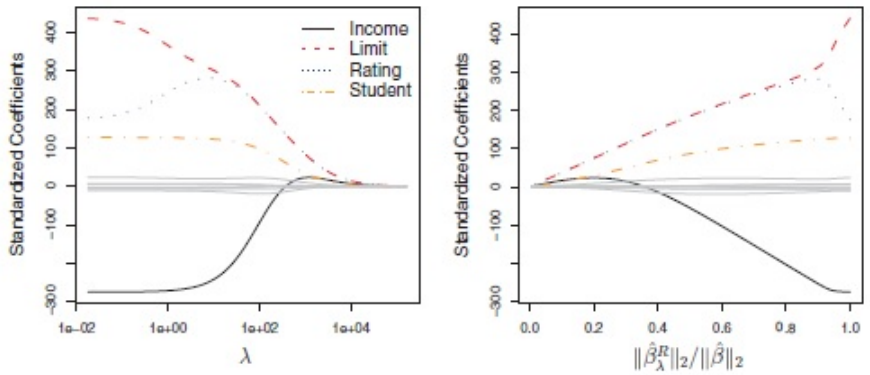

FIGURE 6.4. *The standardized ridge regression coefficients are displayed for the* `Credit` *data set, as a function of* λ *and* $\|\hat{\beta}^R_\lambda\|_2/\|\hat{\beta}\|_2$.

Figure 5.1: Ridge regression coefficient estimates as a function of λ

– The improvement provided by ridge regression over least squares relies on the bias-variance trade-off. Giving less weight to the coefficient estimates works as reducing the flexibility of the model, since this simulates reducing the number of variables. Therefore, as λ increases, the flexibility of the linear model decreases, thus reducing the variance of the estimates and of the underlying predictions for test observations, at the same time, however, the bias increases, attenuating the reducing effect over the test MSE (figure 5.3).

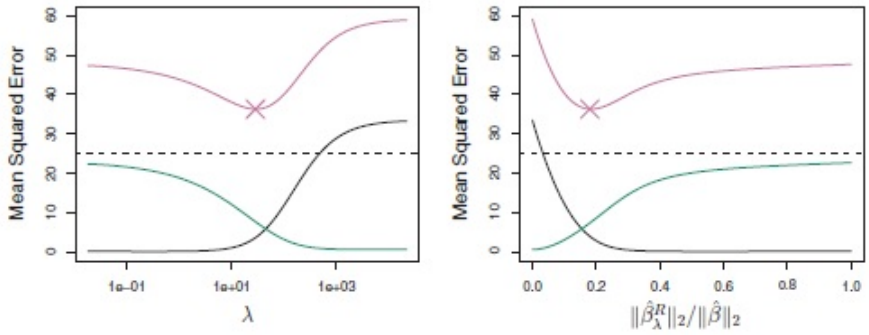

FIGURE 6.5. *Squared bias (black), variance (green), and test mean squared error (purple) for the ridge regression predictions on a simulated data set, as a function of* λ *and* $\|\hat{\beta}^R_\lambda\|_2/\|\hat{\beta}\|_2$. *The horizontal dashed lines indicate the minimum possible MSE. The purple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

Figure 5.2: Ridge regression and the bias-variance trade-off

– As the figure 5.3 shows, intermediate levels of λ may help improving the prediction accuracy of linear regression models. Therefore, ridge regression is an attractive alternative to least squares, specially in circumstances where least squares is prone to suffer from highly variable estimates and predictions, as is the case when n is considerably small comparative to p.

– Comparing ridge regression to subset selection, the first has evident computational advantages over the later, given that ridge regression estimates only one model.

- **Lasso:** differently from ridge regression, the lasso method is constructed in a way that the coefficient estimates can be set precisely equal to zero. This follows from the objective function minimized by the **lasso coefficient estimates** $\hat{\beta}_\lambda^L$:

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = RSS + \lambda\sum_{j=1}^{p}|\beta_j| \tag{35}$$

– Figure 5.3 shows that for sufficiently high values of λ, some of the coefficient estimates equal zero, thus implying in a variables selection property for the lasso method.

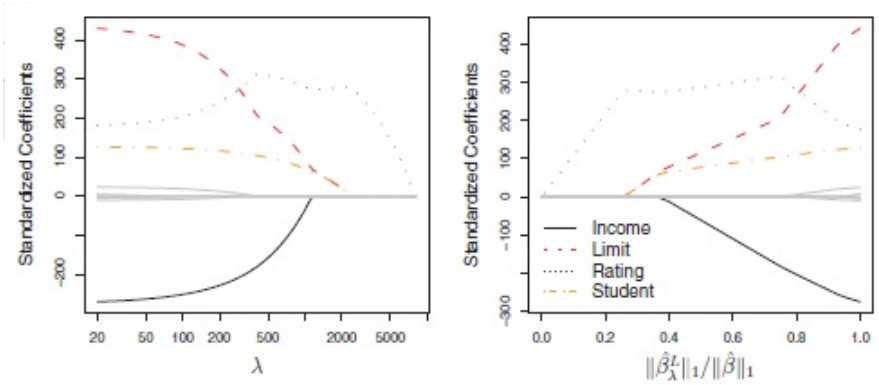

FIGURE 6.6. *The standardized lasso coefficients on the* `Credit` *data set are shown as a function of* λ *and* $\|\hat{\beta}_\lambda^L\|_1/\|\hat{\beta}\|_1$.

Figure 5.3: Lasso coefficient estimates as a function of λ

– Not only expressions (34) and (35) lead to $\hat{\beta}_\lambda^R$ and $\hat{\beta}_\lambda^L$, but also the following problems when resolved imply in the ridge regression and the lasso coefficient estimates:

$$\min_{\beta}\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 \text{ subject to } \sum_{i=1}^{p}\beta_j^2 \leq s \tag{36}$$

$$\min_{\beta} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \text{ subject to } \sum_{i=1}^{p} |\beta_j| \leq s \tag{37}$$

The solutions of (36) and (37) are equivalent to those of (34) and (35) considering that, for each λ, there are appropriate values for s. This alternative formulation for obtaining $\hat{\beta}_{\lambda}^{R}$ and $\hat{\beta}_{\lambda}^{L}$ presents ridge regression and lasso as *constrained* versions of the minimization problem that leads to the least squares estimates. Finally, there is also an alternative formulation that is equivalent to best subset selection:

$$\min_{\beta} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \text{ subject to } \sum_{i=1}^{p} I(\beta_j \neq 0) \leq s \tag{38}$$

Therefore, best subset selection defines an estimate for β in which RSS is minimized constrained to the fact that no more than s predictors are included in the model.

– The fact that the lasso objective function may imply in coefficient estimates equal to zero is geometrically better understood. The expression $\beta_1^2 + \beta_2^2 \leq s$ defines a circle in the two-dimensional space, a sphere in the three-dimensional space and a hypersphere in the p-dimensional space with $p > 3$. In its turn, the expression $|\beta_1| + |\beta_2| \leq s$ represents a diamond, a polyhedron and a polytope in the two-dimensional, three-dimensional and p-dimensional ($p > 3$) spaces (figure 5.3).

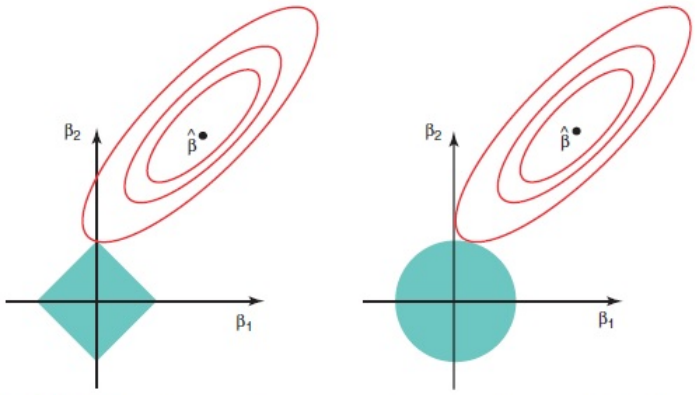

FIGURE 6.7. *Contours of the error and constraint functions for the lasso* (left) *and ridge regression* (right). *The solid blue areas are the constraint regions,* $|\beta_1| + |\beta_2| \leq s$ *and* $\beta_1^2 + \beta_2^2 \leq s$, *while the red ellipses are the contours of the RSS.*

Figure 5.4: Geometrical representation of lasso and ridge regression coefficient estimates

The ridge regression and lasso coefficient estimates are found in the tangency between an iso-RSS curve and the region formed by the constraint $\sum_j \beta_j^2 \leq s$ and $\sum_j |\beta_j| \leq s$, respectively. Given the corners in the constraint region for the lasso method, it is likely that the tangency occurs through an iso-RSS curve that intercepts at least one of the axes.

  - Comparing ridge regression with lasso first allows to conclude that the later has an interpretability advantage over the first, given that lasso yields less complex models as some variables have their parameters set to zero.
  - Considering prediction accuracy, both imply similar patterns for test MSE as λ increases (i.e., as flexibility decreases), in particular, the decrease in variance overcomes the increase in bias until some optimal point is reached, after which bias increasing more than compensate variance reduction. Therefore, both ridge regression and lasso can provide more accurate predictions from estimates less variable than those implied by least squares.
  - Lasso is expected to outperform ridge regression when some (standardized) predictors have substantial coefficients, while the remaining have coefficients with small magnitude, or even equal to zero. Alternatively, ridge regression would be preferred when all (standardized) predictors have coefficients similar in magnitude.

- Shrinkage methods require the definition of the tunning parameter λ. As is the case when defining the best model in subset selection, validation set approach and cross-validation approaches can be applied for determining the value of λ. Once defined a grid of possible values for λ, an estimation of test MSE is calculated for each these values, where the minimum estimated test MSE indicates the optimal tunning parameter value. Then, the procedure of validation or cross-validation continues as usual with the model being refitted using all observations available and the value selected for λ.

- After discussing subset selection and shrinkage methods, it follows that standard linear regression estimated through least squares may be implemented with the support of model selection, where prediction accuracy and model interpretability are improved by the selection of *signal variables* to be part of the model, avoiding the inclusion of *noise variables* that would lead the model to overfit. Shrinkage methods as ridge regression and lasso represent alternatives to least squares, given that prediction accuracy and model interpretability are improved by modifying the least squares procedure.

5.4 Dimension reduction methods

- Both selection of a subset of predictors and shrinkage methods that drive the coefficient estimates toward zero are based on the original variables X_1, ..., X_p. Alternatively, dimension reduction methods transform original predictors into a smaller set of variables Z_1, ..., Z_M, where $M < p$, which are then used to fit a least squares model.

- Dimension reduction methods define linear combinations from the predictors X_1, ..., X_p:

$$Z_m = \sum_{j=1}^{p} \phi_{jm} X_j \tag{39}$$

  Where $m \in \{1, ..., M\}$. Then, a linear regression model can be defined using the linear combinations from (39):

$$y_i = \theta_0 + \sum_{m=1}^{M} \theta_m z_{im} + \epsilon_i \tag{40}$$

  By defining appropriates values for ϕ_{jm}, the equation (40) not only provides a simpler model for the response, but also may lead to more accurate predictions.

- The theoretical equivalence between (40) and the original linear regression model follows from combining (39) with (39):

$$\sum_{m=1}^{M} \theta_m z_{im} = \sum_{m=1}^{M} \theta_m \sum_{j=1}^{p} \phi_{jm} x_{ij} = \sum_{j=1}^{p} \sum_{m=1}^{M} \theta_m \phi_{jm} x_{ij} = \sum_{j=1}^{p} \beta_j x_{ij}$$

  Where:

$$\beta_j = \sum_{m=1}^{M} \theta_m \phi_{jm} \tag{41}$$

  Therefore, dimension reduction methods estimate a linear regression model with constraints over the coefficients that follow (41). Even if these constraints imply in more biased estimations, the smaller set of predictors may reduce the variance of the estimates and, therefore, the variance of predictions, since a less flexible model is estimated.

- Dimension reduction methods are implemented in two steps: first, linear combinations (39) are constructed from the original predictors. Then, a linear regression model for the response is fitted using those linear combinations as predictors. Therefore, alternative dimension reduction methods mainly differ from each other in the ways that parameters ϕ_{jm} are defined.

- **Principal components analysis (PCA):** this technique is used both for dimension reduction and for unsupervised learning problems. The identification of principal components from a dataset considers the directions along with the data vary the most. By doing so, a set of variables can be reduced to only one that captures as best as possible all information contained in those variables.

  – Figure 5.4 represents this dimension reduction from a bi-dimensional case. The green solid line clearly indicates the direction in which the data vary the most, given the amplitude of the line. This is the *first principal component* concerning the two variables plotted in figure 5.4.

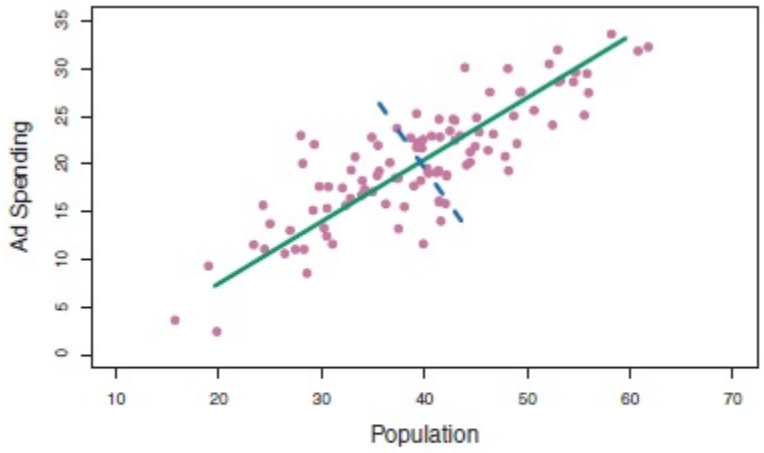

FIGURE 6.14. *The population size (pop) and ad spending (ad) for 100 different cities are shown as purple circles. The green solid line indicates the first principal component, and the blue dashed line indicates the second principal component.*

Figure 5.5: First two principal components directions

  – Mathematically, projecting variables into just one that captures as best as possible all information that follows from those variables requires the definition of ϕ_{11} and ϕ_{12} so that the variance of the linear combination between the two variables $(\phi_{11}(X_{i1} - \overline{X}_1) + \phi_{12}(X_{i2} - \overline{X}_2))$ is maximized subject to a stability condition under which $\phi_{11}^2 + \phi_{12}^2 = 1$.

  – The direction for which the data variability is the largest is equivalent to that defined by the line which is as close as possible to all data points considered, i.e., a first principal component line should minimize the sum of squared distances between the points and the line. Therefore, a first principal component projects data points into a line that is as close as possible to them (figure 5.4). See subsection 9.2 for more on geometrical interpretation of PCA.

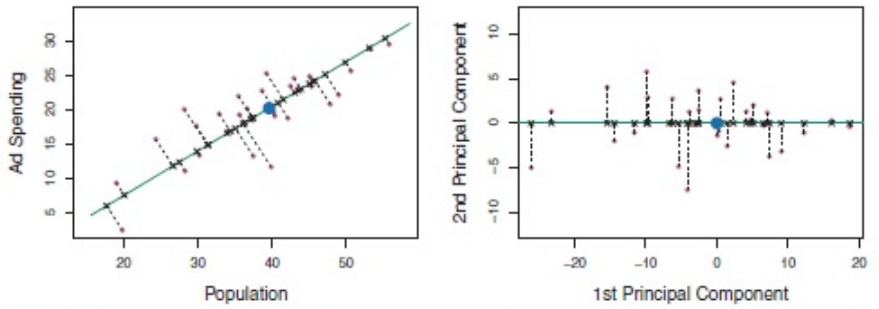

FIGURE 6.15. *A subset of the advertising data. The mean* pop *and* ad *budgets are indicated with a blue circle.* Left: *The first principal component direction is shown in green. It is the dimension along which the data vary the most, and it also defines the line that is closest to all* n *of the observations. The distances from each observation to the principal component are represented using the black dashed line segments. The blue dot represents* $(\overline{\texttt{pop}}, \overline{\texttt{ad}})$. Right: *The left-hand panel has been rotated so that the first principal component direction coincides with the* x*-axis.*

Figure 5.6: First principal component direction and first two principal components projections

- The capacity for reducing the information provided by two distinct variables into only one scalar depends on how strongly related are those variables. If two variables are highly correlated, then defining principal components from them will lead to linear combinations that capture very well the information present in the original variables.
- Besides, given that a first principal component Z_1 was constructed, then a second linear projection can be defined also following the principle of maximizing its variance, but now with an additional constraint being established: this second principal component Z_2 should be independent from the first Z_1. This maximized variability and independence from other principal components apply to all further principal components to be constructed Z_3, ..., Z_M, where $M \leq p$.
- Geometrically, the second principal component is represented by a direction that is orthogonal to that of the first principal component, as illustrated by the blue dashed line in figure 5.4.
- Irrespective of the original feature space dimension (p), all variance-maximizing directions implied by principal components are summarized in the vectors $\phi_m = (\phi_{m1}, ..., \phi_{mp})$. Thus, each of these vectors defines a line in the p-dimensional space which is the closest possible to the data points, given the orthogonality conditions.
- Since all principal components beyond the first are obtained by maximizing the variance of

the linear combination subject to an independence constraint, it follows that the first principal component always contains the most information related to the original variables.

- **Principal components regression (PCR):** the PCR approach to estimate a linear regression model with dimension reduction is based on the calculation of the first M principal components Z_1, ..., Z_M from the original set of variables X_1,, ..., X_p. Then, the response Y is fitted using the principal components as predictors. Implicitly, this assumes that only the directions along with the data vary the most are sufficient to efficiently define the response.
  - If this assumption underlying PCR is valid, then the bias of using projections of the predictors instead of its effective values will be more than compensated by the reduction in variance that follows from the less flexible model estimated. Thus, PCR implies an attenuation to overfit given that $M < p$ variables are used to fit the data.
  - As in general, increasing the number of principal components used in the linear regression leads to a reduction in bias, but also to an increasing in variance. Beyond to some optimal point, the increase in variance may compensate the reduction in bias, defining a U-shaped test set MSE.
  - Comparing PCR to ridge regression or lasso results that PCR can outperform shrinkage methods if only a few principal components are necessary for good prediction results. Besides, given that PCR uses all predictors to construct principal components, this approach does not produce feature selection, and thus is more related to ridge regression than to lasso.
  - In practice, the number of principal components to be used is defined through validation set approach or cross-validation approaches. Finally, as principal components parameters ϕ_{mj} are defined from maximizing the variance of linear combinations of original predictors, all variables should be standardized prior to PCA proceedings, since otherwise high-variance variables or variables with large scales would be over-weighted.
- **Partial least squares (PLS):** PCA defines linear combinations from the original predictors without considering the values of the response variable. Even that an implicit assumption is made by PCR, there is no guarantee that the directions in which the predictors vary the most are those that better predict the response.
  - Thus, PCA produces unsupervised dimension reduction, differently from partial least squares,

which defines parameters ϕ_{mj} in such a way that the resulting linear projection is as correlated with the response as possible. This follows from defining a first direction $Z_1 = \sum_{j=1}^{p} \phi_{1j} X_j$ where ϕ_{1j} is the estimated coefficient from the single linear regression of Y against X_j.

– As figure 5.4 shows, the fact that first PLS direction has also as an objective defining a direction that is the most correlated with the response implies that it not necessarily adjusts as well as possible to the data, which is provided instead by the first PCA direction.

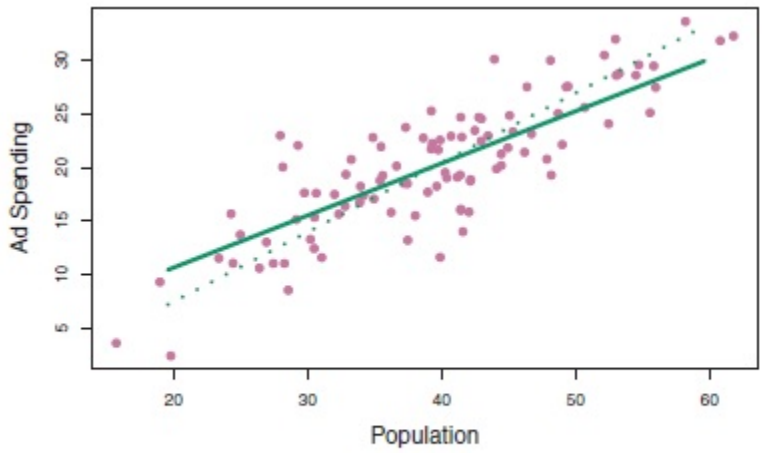

FIGURE 6.21. *For the advertising data, the first PLS direction (solid line) and first PCR direction (dotted line) are shown.*

Figure 5.7: First direction defined by PLS

– The second PLS direction Z_2 follows from the regression of each X_j against Z_1, from which the residuals $\hat{u}_j$ are taken and then used as predictors in single linear regressions of Y on $\hat{u}_j$, resulting in ϕ_{2j}. Not only the estimated coefficient of this variable is correlated with Y, but also captures the effect of X_j on Y that is uncorrelated with the first direction Z_1.

– A procedure like this is repeated for all further PLS directions Z_3, ..., Z_M. The final step of partial least squares approach is to use those PLS directions as predictors in a linear regression model for the response.

  * Check page 81 of The Elements of Statistical Learning for more on the PLS algorithm.

– As for PCR, the tunning parameter M is usually defined through validation set approach or cross-validation set approaches. Besides, it is also recommended to previously standardize all predictors and the response.

- **High-dimensional data:** even though most of statistical methods were designed to low-dimensional settings where the number of observations n is far larger than the number of predictors p, recently it became feasible in many contexts the storage of an incredibly large number of features from a fixed number of observations, implying that n may be approximately equal to p or even that n may be smaller than p. Settings like this are defined to have *high-dimensional* data.
  - The application of statistical learning methods to high-dimensional datasets makes more relevant discussions on bias-variance trade-off and overfitting.
  - Irrespective of the existence of a relationship between predictors and response, in high-dimensional datasets necessarily linear regression models estimated through least squares will result in a perfect fit. This is more easily seen when $n = 2$. Even if Y has no correlation with X, a perfect fit can be obtained by defining a line that crosses the two data points (figure 5.4).

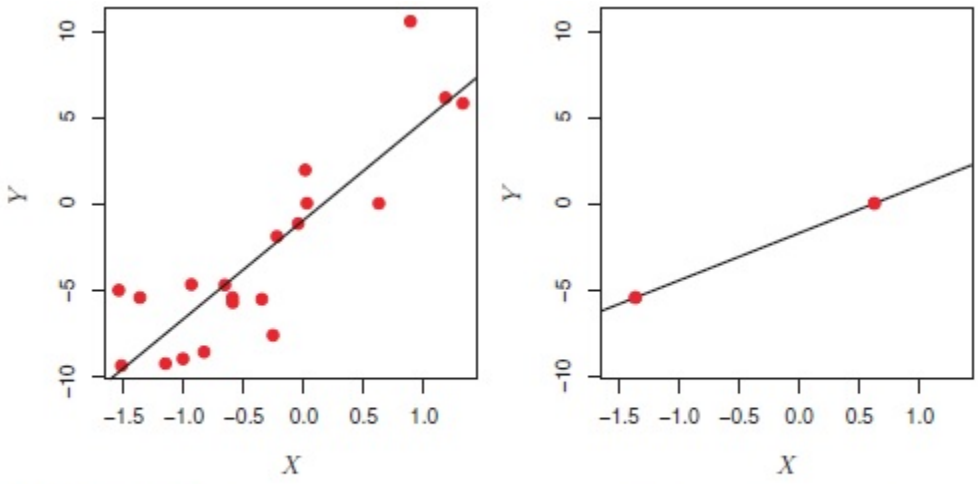

FIGURE 6.22. Left: *Least squares regression in the low-dimensional setting.* Right: *Least squares regression with* $n = 2$ *observations and two parameters to be estimated (an intercept and a coefficient).*

Figure 5.8: Perfect fit with high-dimensional data

  - The overfitting that may be implied by a linear regression model with high-dimensional data is understood when a new data point is considered, given that it may be located far away from the fitted line, resulting in a large test set MSE. Therefore, when $p > n$ or even if $p \approx n$, the linear regression model may be far more flexible than the necessary.
  - In more general high-dimensional settings, the R^2 will be approximate to 1 and the training

set MSE (or, RSS) very low, while the test set MSE may be extremely large. This makes even more important to never draw conclusions from a model considering training set performances. Besides, measures as C_p, AIC, BIC and adjusted R^2 are also not recommended in high-dimensional settings.

- With high-dimensional datasets, procedures as subset selection, ridge regression, lasso, PCR and PLS can be used to avoid overfitting as less flexible models are estimated. Despite of the use of shrinkage methods, for example, predictive performance gets worse when the number of predictors increases, except if the additional predictors are truly related with the response.
- This relates to the *curse of dimensionality*, incurred when a large set of predictors is available, containing only some *signal features* that contribute to reduce test set MSE, while the others are just *noise features* that only lead to overfitting the model as the flexibility goes beyond the optimal level. In fact, even the inclusion of signal variables increase the estimation and, thus, the prediction variances, which may more than compensate the reduction in bias.
- When interpreting linear regression models with high-dimensional data, not only performance measures must be read with caution. The selection of a model becomes more likely to be dependent of the training observations used to fit the model. This means that by training the model with a different set of observations, it may the case that a very distinct model shows to be the best for this different dataset. Thus, when a model is selected in high-dimensional settings, it should not be seen as the best among all, but only as one of many possible models that perform well to predict the response.

5.5 Practical issues

- While model selection when performed through statistics such as AIC, BIC, C_p or adjusted R^2 can be made from the entire dataset, when validation set approach or cross-validation approaches are used the dataset must be separated, either on train and test sets or on k different folds.
  - Considering k-fold CV as an example, after creating k folds from the entire dataset, for each $j \in \{1, 2, ..., k\}$ the best set of variables is selected for all model sizes, implying in models $M_{j0}, M_{j1}, ..., M_{jp}$.
  - This selection uses as reference training set metrics (RSS, R^2), and is based on all folds except for j. Besides, both best subset selection and stepwise selection can perform searches within

each model size.

  - Then, those models are used to predict the response for observations in fold j, and an estimate of MSE is calculated. Averaging all these MSE estimates across $j \in \{1, 2, ..., k\}$, the k-fold CV estimates for test error by model size are obtained, helping to select the optimal number of predictors.
  - Instead of selecting $M_{j0}, M_{j1}, ..., M_{jp}$ for $j \in \{1, 2, ..., k\}$ inside of k-fold CV algorithm, another possibility is to define just one set $\{M_0, M_1, ..., M_p\}$ from the entire dataset, and then apply k-fold CV only to define the best among all of those models.

- After selected the optimal number of predictors in subset selection, or having been defined the optimal tuning parameter in ridge regression or lasso (shrinkage methods), or even after defining the number of principal components or directions in PCR or PLS (dimension reduction methods), the model should be refitted using the entire dataset.

  - Considering validation set approach and CV approaches for subset selection, just the number of predictors should then be used. Therefore, the final model in this context is given by the best model for the optimal number of predictors, being the variables selection based now on the entire dataset.

- Selection of the best approach for linear regression modelling (subset selection, ridge regression, lasso, PCR, or PLS), or, more generally, selection of the best statistical learning method:

  - Option 1: comparison among the test MSE CV estimates for the best alternative within each approach or statistical learning method, where the estimates used are calculated from k-fold CV applied on the entire dataset.
  - Option 2: a prior dataset split can be done, defining prior-train and prior-test sets. Then, validation set approach or CV approaches are performed on prior-train set, revealing the best alternative within each approach or statistical learning method. Finally, predictions are made for prior-test set observations, from which follow definite test MSE estimates used as a reference for each approach or method.

6 Moving beyond linearity

6.1 Extending the linear regression model

- Linear models may perform well for predictive purposes, and are specially relevant in the context of inference. However, adding more flexibility may as well help to enhance even further the accuracy of predictions, given that in various circumstances the linearity assumption leads to large bias when fitting the data.

- If subset selection, shrinkage methods and dimension reduction methods still relate linearly each predictor with the response and improve model accuracy mostly by reducing variance in the estimates, here non-linear approaches are discussed mostly seeking to reduce bias, while interpretability is keep as closest as possible to that implied by linear models.

- The first natural approach for breaking the linearity assumption of the relationship between predictor X and response Y is to use as predictors of an extended linear model *polynomial functions* of X, X^2, X^3, and so on. Another simple way to increase the flexibility of a linear model is by creating intervals for the predictor X, making its contributions for explaining Y dependent on the levels assumed by X. This approach are implemented differently through *step functions* and *local regressions*. More complex approaches for making the fitted curve more sensible to the data points are *regression splines* and *smoothing splines*. Finally, *Generalized Additive Models* extend these approaches to the multiple linear regression setting with predictors X_1, X_2, ..., X_p.

6.2 Polynomial regression

- The simple linear regression that relates a response Y with a predictor X can be extended by adding monomials up to a degree d into the regression model:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + ... + \beta_d x_i^d + \epsilon_i \tag{42}$$

  The coefficients on (42) are estimated through least squares as usual. As d gets higher, the curve fitted to the data becomes increasingly more non-linear. Even so, it is common in practice to not define d to be more than 3 or 4, in order to avoid overfitting as the curve assumes wiggly shapes when trying to get closer to each training observation.

- The analysis of the output of polynomial regressions considers plots of the actual and fitted responses for a given interval of the predictor, since individual coefficient estimates may contain few practical information.
- The fitted curve usually is accompanied by confidence intervals constructed from the prediction variance at each data point, which in its turn follows from the variances of coefficient estimates.
- Similarly to regression models, classification modeled by logistic regression can also be modified so as to accommodate polynomial functions. Instead of $\beta_0 + \beta_1 x_i$, a function as $\beta_0 + \beta_1 x_i + \beta_2 x_i^2 + ... + \beta_d x_i^d$ is inserted in the cumulative logistic function that consists on the probability $P(y_i = 1|x_i)$.

6.3 Step functions

- Another approach to fit a curve that gets as close as possible to the data points without a linearity assumption, so that the curve can take very general shapes, is to convert a numerical feature X into a ordered categorical variable. This requires the definition of K cutpoints c_1, c_2, ..., c_K that break the range of values of X into $K + 1$ segments. Consequently, $K + 1$ different variables are defined:

$$\begin{aligned} C_0(X) &= I(X < c_1) \\ C_1(X) &= I(c_1 \leq X < c_2) \\ &\vdots \\ C_{K-1}(X) &= I(c_{K-1} \leq X < c_K) \\ C_K(X) &= I(c_K \leq X) \end{aligned} \tag{43}$$

Dropping one of these variables so perfect linear multicolinearity is avoided, a regression model estimated through least squares is constructed from the step functions in (43):

$$y_i = \beta_0 + \beta_1 C_1(x_i) + \beta_2 C_2(x_i) + ... + \beta_K C_K(x_i) + \epsilon_i \tag{44}$$

Each β_j represents the change in conditional expected value of y_i when x_i rises from $x_i < c_1$ to the interval $c_j \leq x_i < c_{j+1}$. As it was the case for polynomial regression, also for step function regression it is possible to use this structure instead of the usual linear model inside the cumulative logistic function that defines $P(y_i = 1|x_i)$. The named piecewise constant regression in (44) produces a

curve with constant values during predefined intervals of X, adjusting itself according with the levels of the response (figure 6.3):

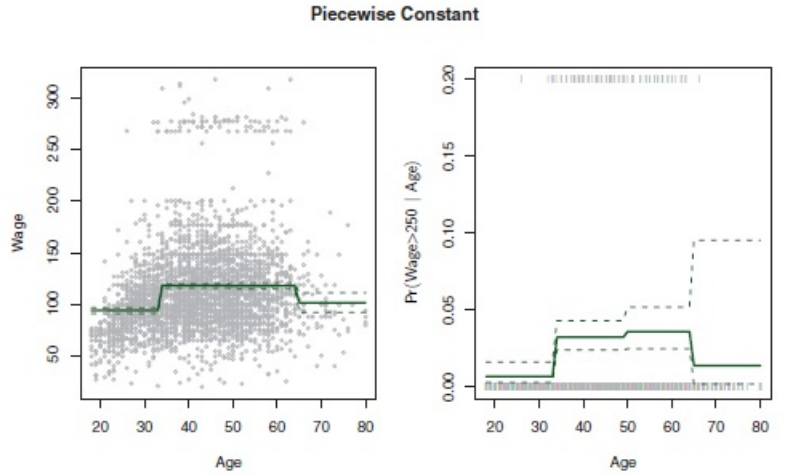

FIGURE 7.2. *The* Wage *data.* Left: *The solid curve displays the fitted value from a least squares regression of* wage *(in thousands of dollars) using step functions of* age*. The dotted curves indicate an estimated 95 % confidence interval.* Right: *We model the binary event* wage>250 *using logistic regression, again using step functions of* age*. The fitted posterior probability of* wage *exceeding* $250,000 *is shown, along with an estimated 95 % confidence interval.*

Figure 6.1: Piecewise constant regression

6.4 Basis functions

- Both equations (42) and (43) share a common structure in which d or K different *basis functions* of x_i are defined. A general version of these equations is as follows:

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + ... + \beta_K b_K(x_i) + \epsilon_i \tag{45}$$

In polynomial regression, $b_j(x_i) = x_i^j$, while for piecewise constant regression $b_j(x_i) = I(c_j \leq X < c_{j+1})$. Coefficients in equation (45) are estimated through least squares, and more complex definitions for $b_j(x_i)$ lead to alternative approaches for non-linear regression.

6.5 Regression splines

- **Piecewise polynomial regression** combines aspects of both polynomial regression and step functions, since it defines one polynomial function for each segment of the range of values of the predictor

X. For instance, piecewise cubic polynomial regression with a single *knot* at $X = c$ where the regression curve changes is given by:

$$y_i = \begin{cases} \beta_{01} + \beta_{11}x_i + \beta_{21}x_i^2 + \beta_{31}x_i^3 + \epsilon_i & \text{if } x_i < c \\ \beta_{02} + \beta_{12}x_i + \beta_{22}x_i^2 + \beta_{32}x_i^3 + \epsilon_i & \text{if } x_i \geq c \end{cases} \tag{46}$$

Least squares can be used to estimate the two equations in (46) as the dataset is divided into two parts according with the value assumed by X, and two different regression models are fitted. Defining more than one knot makes the piecewise cubic polynomial more flexible, and the fitted curve gets even more close to the training observations.

- The curve formed by joining those two that follows from (46) may not necessarily be continuous at the knots. However, the piecewise polynomial function can be estimated under a constraint that the resulting curve is continuous at each knot. Moreover, additional constraints can be set to also define first and second derivatives continuous at the knots, making even more smooth the transition between the curves. The resulting curve is named *cubic spline*, or, more generally, *degree-d spline*, where the first $d - 1$ derivatives are required to be continuous at each knot. Figure 6.5 illustrates the differences between piecewise polynomial regression and degree-d regression splines.

- Estimating a cubic spline with K knots that fits different curves for each segment of the range of values of X can be accomplished through a single equation that uses appropriate basis functions:

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + ... + \beta_{K+3} b_{K+3}(x_i) + \epsilon_i \tag{47}$$

Where $b_1(x_i)$, $b_2(x_i)$ and $b_3(x_i)$ are given by x_i, x_i^2 and x_i^3, respectively. The remaining K basis functions in (47) have the following form:

$$h(x, \xi_k) = (x - \xi_k)_+^3 = \begin{cases} (x - \xi_k)^3 & \text{if } x_i > \xi_k \\ 0 & \text{otherwise} \end{cases} \tag{48}$$

Where ξ_k is a given knot, and $h(x, \xi_k)$ is named *truncated power basis function*. The insertion of K truncated power basis functions in (47) guarantees that the fitted curve and its first two derivatives are continuous at each knot ξ_k.

  - As mentioned previously, more knots lead to a more flexible fit. A measure of this flexibility is given by the *degrees of freedom* of the cubic spline (47). Since this equation presents $4 + K$ coefficients to estimate, this model has $4 + K$ degrees of freedom.

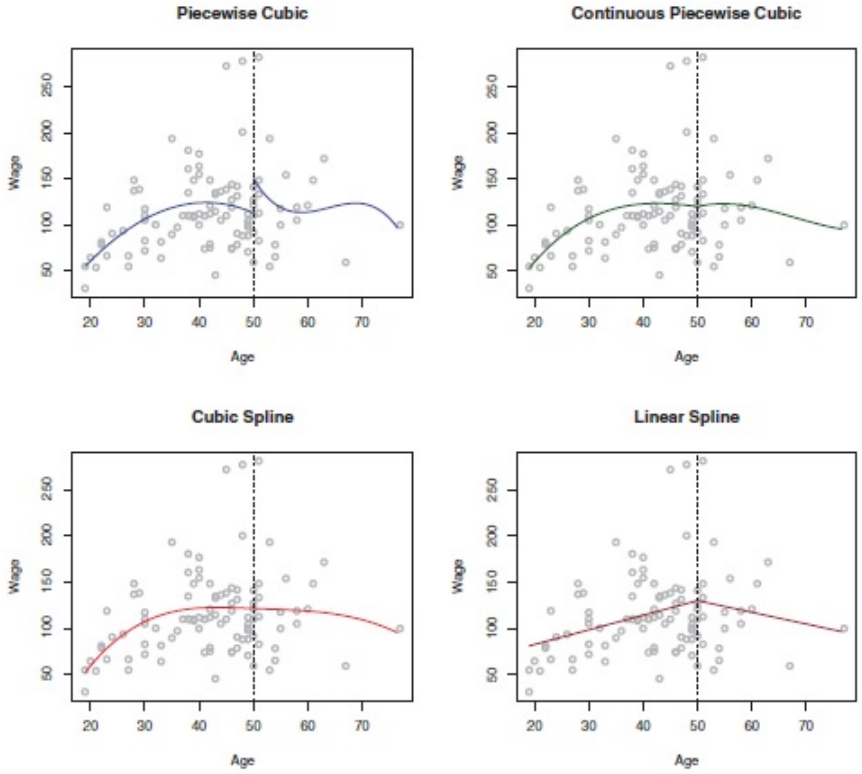

FIGURE 7.3. *Various piecewise polynomials are fit to a subset of the* Wage *data, with a knot at* age=50. Top Left: *The cubic polynomials are unconstrained.* Top Right: *The cubic polynomials are constrained to be continuous at* age=50. Bottom Left: *The cubic polynomials are constrained to be continuous, and to have continuous first and second derivatives.* Bottom Right: *A linear spline is shown, which is constrained to be continuous.*

Figure 6.2: Piecewise polynomial regression and regression splines

– Another improvement upon a simple piecewise polynomial regression is to impose that the fitted curve is linear in the boundaries. Since there are usually only few points in the segments $X < \xi_1$ and $X > \xi_K$, the variance of the estimates is high when trying to fit all the points in those regions. Consequently, more accurate estimates follow from defining that the fitted curve is linear at the outer range of X. These *boundary constraints* imply in **natural splines** whose estimates tend to be more stable than those from cubic (or, degree-d) splines (figure 6.5).

– In regression splines, the main choices to be made are the number and location of the knots $\xi_1, \xi_2, ..., \xi_K$. Few knots may not capture all patterns in the non-linear data, at the same time a large number of knots may lead to overfitting. One possibility is to define more knots where

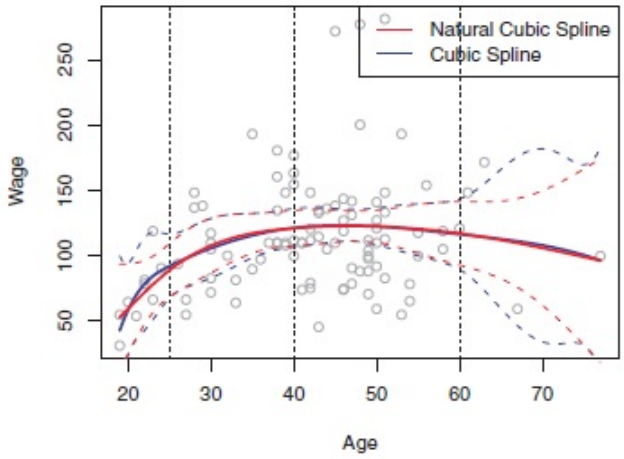

FIGURE 7.4. *A cubic spline and a natural cubic spline, with three knots, fit to a subset of the* Wage *data.*

Figure 6.3: Regression splines and natural splines

the relationship between Y and X changes strongly, and only few knots in more stable regions. Another approach is to define the desired degrees of freedom and to distribute the resulting knots uniformly across the quantiles of X. For example, if one wants to estimate a natural cubic spline with 4 degrees of freedom, it follows 3 knots to be inserted in the percentiles $25th$, $50th$ and $75th$: $4 + K = 4 + 1 + 2$, more generally, $4 + K = df + int + bound$, where df is the desired degrees of freedom, int corresponds to the intercept and $bound$ indicates the 2 additional constraints that follow from the natural splines approach.

– The choice of df, or, equivalently, K can be made on the basis of cross-validation procedures. A grid of values for df or K is created and cross-validated RSS is calculated for each value of df or K. That one which minimizes CV-RSS should be chosen. If more than one variable is present, then one general df value can be set to all predictors, instead of applying the CV procedure for each variable.

– Polynomial regression usually incorporates a large number of monomials in order to achieve a more flexible fit. Cubic natural spline, in its turn, still uses only three polynomial degrees, where the flexibility is increased by the use of knots dividing the range of values of X and capturing more of the non-linear pattern in the data. Consequently, natural splines tend to produce more stable estimates when compared to polynomial regression.

6.6 Smoothing splines

- Polynomial regression, piecewise constant regression and regression splines all use least squares as the estimation method, modifying the standard linear regression so as to the fitted curve approximates the most non-linear data points. Smoothing splines use more sophisticated estimation techniques, and are based on the search for a function $g(x)$ that minimizes RSS, but that also avoids overfitting, since it would be possible to interpolate all training observations, at the cost of highly variable estimates and, therefore, highly variable predictions.

- The function $g(x)$ that fits smoothly the data points should minimize RSS under the penalty of not being too wiggly, which would produce inaccurate estimates given the low-bias, but extremely high variable estimation. The following objective function summarizes this approach:

$$\sum_{i=1}^{n}(y_i - g(x_i))^2 + \lambda \int g''(t)^2 dt \tag{49}$$

  As for ridge regression and lasso, the parameter λ is a *tunning parameter*.

  - The second derivative $g''(t)$ of a function $g(t)$ measures how strongly the change in this function is varying. A second derivative large in absolute value indicates a wiggly-shape curve, given the jumpy nature of $g(t)$. A second derivative close to zero indicates an approximately constant first derivative (as with linear functions), so that $g(t)$ changes smoothly as y varies.

  - Therefore, the larger λ is, the more smooth $g(t)$ should be. If $\lambda = 0$, then $g(t)$ will interpolate all training observations, given that no restriction will be set on how $g(t)$ should change. It is clear how λ controls the bias-variance trade-off, as for high values of λ the variance is low, but the bias is relatively high, and for low values of λ the bias is small, but the variance will increase.

  - The function $g(t)$ that minimizes (49) can be shown to be a piecewise cubic polynomial with knots at each data point $x_1, x_2, ..., x_n$, with continuous first and second derivatives at each knot and that is linear in the boundaries. Thus, a smoothing spline is a natural cubic spline with knots at each data point $x_1, x_2, ..., x_n$. Since this would imply in very high degrees of freedom, it is estimated a shrunken version of a natural cubic spline.

  - In this context, an *effective degrees of freedom* measure is used as reference. This alternative degree varies from n to 2 as λ goes from 0 to ∞.

– The choice of λ follows from cross-validation. Similar to (27) in the context of multiple linear regression, there is also an expression for the LOOCV-RSS as a function of λ:

$$RSS_{CV}(\lambda) = \sum_{i=1}^{n}(y_i - \hat{g}_{\lambda}^{(-i)}(x_i))^2 = \sum_{i=1}^{n}\left(\frac{y_i - \hat{g}_{\lambda}(x_i)}{1 - \{S_{\lambda}\}_{ii}}\right)^2 \tag{50}$$

Where $\hat{g}_{\lambda}^{(-i)}(x_i)$ is the fitted value of the response for observation i calculated from all observations but i, while $\hat{g}_{\lambda}(x_i)$ is the fitted value calculated from all observations. $\{S_{\lambda}\}_{ii}$ is the i-th element of the main diagonal of matrix S_{λ}, which follows from the following relationship:

$$\hat{g}_{\lambda} = S_{\lambda}y \tag{51}$$

Where $\hat{g}_{\lambda}$ is the vector of fitted values and y is the vector of observed responses.

6.7 Local regression

- Local regression works similarly to step function in the sense that curves are fitted to the data considering only successive short intervals of the predictor X.

- **Procedure for local regression at** $X = x_0$:

  1. Define the fraction $s = k/n$ of training points whose x_i are closest to x_0.
  2. Calculate weights $K_{i0} = K(x_i, x_0)$ to each point in this neighborhood in a way that the closest point receives the highest weight and the furthest receives zero. All points outside the neighborhood also receive zero as weight.
  3. Estimate a *weighted least squares regression* of y_i against x_i with K_{i0} as weights. Thus, $\hat{\beta}_0$ and $\hat{\beta}_1$ are calculated from minimizing the following expression:
  $$\sum_{i=1}^{n} K_{i0}(y_i - \beta_0 - \beta_1 x_i)^2$$
  4. The local regression fitted values are given by $\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0$.

- Local regression estimation is a memory-based procedure, since for predicting the response for each different data point (step 4) it is necessary to recalculate weights K_{i0} (step 2) and refit a model through weighted least squares (step 3) using all training observations.

- Local regression implies that the prediction for a given observation x_0 uses only the nearby data points, which represents a flexible approach because attention is devoted only to a specific segment of the range of values of X at each time a prediction is to be made (figure 6.7).

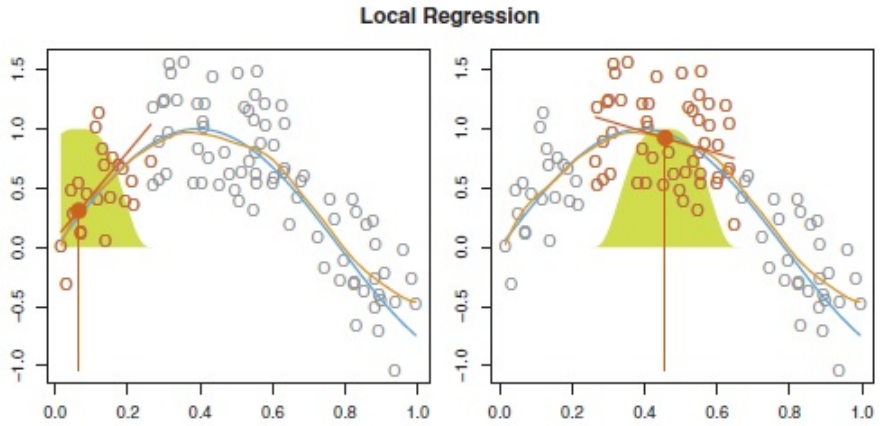

FIGURE 7.9. *Local regression illustrated on some simulated data, where the blue curve represents $f(x)$ from which the data were generated, and the light orange curve corresponds to the local regression estimate $\hat{f}(x)$. The orange colored points are local to the target point x_0, represented by the orange vertical line. The yellow bell-shape superimposed on the plot indicates weights assigned to each point, decreasing to zero with distance from the target point. The fit $\hat{f}(x_0)$ at x_0 is obtained by fitting a weighted linear regression (orange line segment), and using the fitted value at x_0 (orange solid dot) as the estimate $\hat{f}(x_0)$.*

Figure 6.4: Local regression

- Some choices have to made when implementing this method, such as defining the weighting function $K(.)$ and the degree of the polynomial in step 3. A choice that more clearly indicates a flexibility selection is the fraction s of nearest neighbors. The larger s, less flexible the estimation and more smooth the fitted curve. The smaller s, more flexible the estimation and more wiggly the fitted curve. Cross-validation again shows as an option when defining s.

- In multiple linear regression settings, local regression also applies, and a two or three-dimensional neighborhood can be defined in step 1. When p is large, however, the curse of dimensionality may result in a poor performance for this method.

6.8 Generalized additive models (GAM's)

- All approaches discussed so far to flexibly estimate non-linear relationships between X and Y were focused on only one predictor. GAM's have additivity as their main assumption, using it to aggregate general non-linear functions of each predictor $f_j(x_{ij})$ together into a single equation:

$$y_i = \beta_0 + f_1(x_{i1}) + f_2(x_{i2}) + ... + f_p(x_{ip}) + \epsilon_i \tag{52}$$

The functions $f_j(x_{ij})$ can follow from any of the previously discussed approaches, polynomial regressions, step functions, regression splines, smoothing splines or local regressions.

- When (52) has smoothing splines as one of its components, least squares can not be used to estimate the entire GAM equation. Instead, an approach named *backfitting* is applied. It is worth to notice that generally natural splines and smoothing splines have similar performances.

- A first advantage of using GAM's is that it represents an automatic approach for fitting non-linear relationships between Y and X_j, since a function $f_j(X_j)$ can be estimated for each predictor. The less restrictive functional form assumption also tends to generate more accurate predictions. Besides, the additive assumption still allows the use of the model in (52) for inference purposes. Finally, there is a simple measure of the flexibility of each possible model given by the degrees of freedom used. Therefore, GAM consists on an intermediate approach between low flexible approaches such as linear regression and high flexible approaches such as those non-parametric methods.

- The main limitation of GAM's is precisely their additivity assumption, which can make the model in (52) loose interactive relationships between different predictors, where this is specially likely to happen when the set of predictors is very large. Even so, predictors $X_j.X_k$ can be inserted in the GAM, as well as low-dimensional interaction functions $f_{jk}(X_j, X_k)$.

- There is also a classification problem version of GAM when logistic regression is to be used together with non-linear approaches. The following logit function can be defined using non-linear functions of predictors $f_j(X_j)$:

$$\log\Big(\frac{P(Y=1|X)}{1-P(Y=1|X)}\Big) = \beta_0 + f_1(X_1) + f_2(X_2) + ... + f_p(X_p) \tag{53}$$

7 Tree-based methods

7.1 The basics of decision trees

- Tree-based methods apply both for regression and classification problems. Their procedures involve stratifying the predictor space into non-overlapping regions and then predicting the value or class for the response according with the training set mean or most occurring class of the region to which a given test observation belongs.

- A decision tree is composed from *nodes* that lead to splits of the predictor space into different regions, where the splits are represented by *branches*. This applies to all nodes except for the *terminal nodes*, or *leaves*, which indicates final regions R_j, $j \in \{1, 2, ..., J\}$, to which all data points must belong. The predicted response of a given test observation is equal to the mean of the response for training observations in the region R_j in which the test observation is located (figure 7.1).

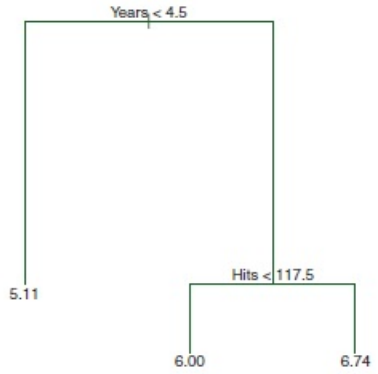

FIGURE 8.1. *For the* `Hitters` *data, a regression tree for predicting the log salary of a baseball player, based on the number of years that he has played in the major leagues and the number of hits that he made in the previous year. At a given internal node, the label (of the form $X_j < t_k$) indicates the left-hand branch emanating from that split, and the right-hand branch corresponds to $X_j \geq t_k$. For instance, the split at the top of the tree results in two large branches. The left-hand branch corresponds to* `Years<4.5`*, and the right-hand branch corresponds to* `Years>=4.5`*. The tree has two internal nodes and three terminal nodes, or leaves. The number in each leaf is the mean of the response for the observations that fall there.*

Figure 7.1: Regression tree

- As mentioned above, predictions from a decision tree involve dividing the predictor space into J different non-overlapping regions R_1, R_2, ..., R_J. Then, for each observation that lies on region R_j (training or test observations), the predicted value for Y (regression setting) will be the average response across all training observations in R_j, $\hat{y}_{R_j}$. Therefore, the main task when training a decision tree is given by the appropriate split of the predictor space.

  – As for linear regression model, the choice of R_1, R_2, ..., R_J seeks to minimize a RSS measure:

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_J})^2 \tag{54}$$

– The feasible method for minimizing (54) is named **recursive binary splitting**. It involves choosing a first X_j variable and a cutpoint s such that the split into $R_1(j,s) = \{X|X_j < s\}$ and $R_2(j,s) = \{X|X_j \geq s\}$ minimizes the following expression:

$$\sum_{i:x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2 \tag{55}$$

After this first binary split, another variable (that can be the same as above) and another cutpoint is chosen so as to minimize RSS from the split of one of the previously defined regions R_1 or R_2. Now, there are three terminal nodes, and another pair of variable and cutpoint is chosen from the minimization of RSS by splitting one of those three created regions. This process continues up to some condition is satisfied, such as at most a given number of observations belong to each terminal node. Figure 7.1 presents a decision tree for which the process described has continued only for one additional step.

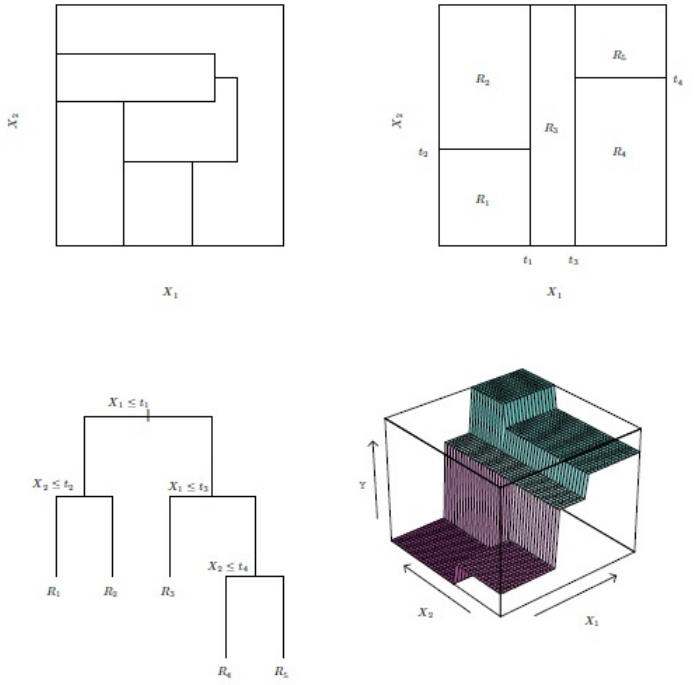

FIGURE 8.3. Top Left: *A partition of two-dimensional feature space that could not result from recursive binary splitting.* Top Right: *The output of recursive binary splitting on a two-dimensional example.* Bottom Left: *A tree corresponding to the partition in the top right panel.* Bottom Right: *A perspective plot of the prediction surface corresponding to that tree.*

Figure 7.2: Regression tree with 5 terminal nodes

- The binary recursive splitting may imply in overfitting, given that no restriction is set to the size of the resulting tree, which can become too complex, fitting pretty well the training data, but having bad performance on test observations. **Tree pruning** is oriented to stop the tree construction when the additional reduction in RSS is no larger than some predefined value. Since this may stop the tree building process too soon, given that a small reduction in RSS can be succeeded by a large reduction later, the **cost complexity pruning** creates a large and complex tree T_0 from the binary recursive splitting procedure, which is followed by the selection of a subtree with the lowest possible test error.

  - For each value of a tunning parameter α, where $\alpha \geq 0$, there is a subtree $T \subset T_0$ that minimizes the following expression:

$$\sum_{m=1}^{|T|} \sum_{i:x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha|T| \tag{56}$$

    Where $|T|$ refers to the number of terminal nodes of the subtree T and $\hat{y}_{R_m}$ is the mean of Y for the training observations in R_m. The tunning parameter α penalizes the complexity of the tree T, since a large value of α requires that T has few terminal nodes ($|T|$ small) in order to (56) be minimized. Alternatively, if $\alpha = 0$, then no penalty is defined to the size of the tree, so that $T = T_0$.

  - **Procedure to build a regression tree:** k-fold CV is used to choose a value for α (see page 309 of AISL for an alternative presentation):

    1. Divide the data set into K different folds of data. For each $k \in \{1, 2, ..., K\}$:

       1.1 Using all folds but k, create a large reference tree T_{k0} through binary recursive splitting, where no terminal node should have more than some predefined number of observations.

       1.2 Also for all folds but k, apply cost complexity pruning to T_{k0}, obtaining a subtree for each value of α, T_α.

       1.3 Use each subtree T_α to predict the response for observations in the fold k. Then, calculate test set MSE.

       1.4 For each α, calculate the K-fold CV estimate of MSE by averaging all K test set MSE estimates.

       1.5 Define the best α as being that the minimizes the K-fold CV estimate of MSE.

2. Apply binary recursive splitting to the entire data set, obtaining T_0. Then, use cost complexity pruning with the chosen value of α to estimate the best subtree.

– As was discussed by the end of subsection 5.5, either K-fold CV can be applied to the entire dataset (option 1), or a prior split can be made, separating a prior-train set from a prior-test set (option 2). In the later case, the prior-train set is then subject to K-fold CV, revealing the best tunning parameter α. Fitting the best subtree associated with the chosen α using all observations in the prior-train set and predicting the response for the observations in the prior-test set can lead to an estimation of the test set MSE. In fact, such an estimation can be produced not only for the best α, but also for all the others, from which is obtained a curve similar to that of (prior-train set) K-fold CV, though usually with a higher level.

- **Classification trees:** in the case where Y is qualitative, instead of the mean, the decision tree assigns to a test observation the most occurring class (among training observations) inside the region in which that observation is located. Moreover, class probabilities can be estimated from the relative frequency of each class inside a given region.

  – Similar to regression trees, classification trees also use binary recursive splitting to construct a large tree T_0. However, RSS can not be used to guide the binary splits. The **classification error rate** is a natural candidate to serve as reference in the process of constructing a classification tree. If $\hat{p}_{mk}$ is the relative frequency of class k in region m, then to all observations in this region it is assigned the class k such that $\hat{p}_{mk}$ is maximum. Therefore, the classification error rate is given by:

$$E = 1 - \max_k(\hat{p}_{mk}) \tag{57}$$

  However, other measures are more sensitive to **node purity**, such as the **Gini index**:

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}) \tag{58}$$

  And the **cross-entropy**:

$$D = -\sum_{k=1}^{K} \hat{p}_{mk} \log(\hat{p}_{mk}) \tag{59}$$

  The two later expressions (58) and (59) take on values close to zero if most $\hat{p}_{mk}$ are either close to zero or one, justifying their larger sensitivity to node purity.

- It is worth notice that if node purity is considered relevant, then using Gini index or cross-entropy should be preferred when constructing classification trees through binary recursive splitting. However, if the accuracy of the final pruned tree is the main goal, then classification error rate may be a better option.
- Qualitative predictors can also be used to construct either for regression or classification trees. In this case, binary splits from qualitative predictors allocates some categories to one branch and the remaining to the other branch.
- Node purity is relevant for increasing the reliability of classifications when final regions have an extremely large share of a given class.

- The linear regression model, and other methods that follow from this classical approach, define a specific functional form for the relationship between X and Y:

$$f(X) = \beta_0 + \sum_{j=1}^{p} \beta_j X_j \tag{60}$$

While regression trees are more generic when relating X to Y:

$$f(X) = \sum_{m=1}^{M} c_m I(X \in R_m) \tag{61}$$

If the true relationship between X and Y can be well approximated by a linear expression such as (60), then linear regression model should outperform decision trees. However, if the systematic relationship between response and predictors is highly complex and non-linear, tree-based methods are a good alternative to classical approaches. Besides, decision trees have another advantage over linear models: they are very easy to present and interpret.

- Summarizing pros and cons of decision trees, first there is the above-mentioned facility of interpretation of the prediction procedure and of the relationship between X and Y, specially when they are graphically represented. Related to this, decision trees may capture the way how individuals decide about a given response considering their observed features. The main disadvantage of decision trees consists on their poor expected accuracy of prediction, which is aggravated by the fact that the final constructed tree is highly dependent on the training data used.

7.2 Bagging, random forests, boosting

- These methods artificially introduce more variability in tree construction, seeking to improve the predictive accuracy of tree-based models.

- Decision tree is a method with high variance in its results, since changing the underlying training data may imply in very different trees and consequently may give predictions with high variance. **Bagging** is a technique based on bootstrap, which are usually applied to estimate parameters, such as standard deviation, from the resample of a given data set. It happens that resample also tends to reduce the variance of estimates from any statistical learning method, because averaging a set of different estimations leads to an estimate whose variance is lower than that of each of its components.

  - Ideally, B different training sets should be achieved, so that B different fitted models $\hat{f}^b$ could be produced. Then, a final prediction would follow from the average of all predictions $\hat{f}^b(x)$:

    $$\hat{f}_{avg}(x) = \sum_{b=1}^{B} \hat{f}^b(x) \tag{62}$$

    Resampling shows up as an alternative to (62) given the fact that obtaining different training sets may be unfeasible. Instead of using B different datasets, bootstrapped subsets from the original dataset are used to fit B models $\hat{f}^{b^*}$, whose predictions are averaged to result in a final bagging prediction:

    $$\hat{f}_{bag}(x) = \sum_{b=1}^{B} \hat{f}^{b^*}(x) \tag{63}$$

    In the context of decision trees, each bootstrapped individual tree $\hat{f}^{b^*}$ is constructed from binary recursive splitting without any further pruning, so that each model $\hat{f}^{b^*}$ has low bias, but high variance, which is reduced precisely by the averaging procedure.

  - When bagging is applied to classification problems, not the average of individual predictions is taken, as indicated in (63), but the most occurring class from all individual predictions is considered the final prediction of bagging.

  - Differently from most statistical learning methods, the parameter B can be as large as possible so as to minimize test error estimates, without the concern about overfitting.

  - The estimation of test set error in the context of bagging has an alternative to validation set approach and to cross-validation approaches. Each time a bootstrapped tree is constructed,

1/3 of the observations tend to not be used in the calculations. Therefore, for each training data observation i, there are $B/3$ different predictions that are made without using the observation i in the fitting procedure. Averaging these **out-of-bag (OOB)** predictions for each observation $i \in \{1, ..., n\}$ to calculate test error measures implies in a valid estimate of the test set error. Besides, as B increases indefinitely, this OOB error estimation is equivalent to LOOCV estimation.

  - Bagging improves prediction accuracy at the cost of reducing interpretability, since it does not rely on a single decision tree that can be graphically represented. Even so, there is a way to present *variables importance* from the average of all B trees of the reduction in RSS, or classification error measures (overall error, Gini index, cross-entropy), that is provided by a split in a given predictor X_j.

- Bagging reduces the variance of predictions by averaging different predictions from estimated models based on bootstrapped sub-samples of the entire dataset. However, if the individual estimates are correlated, then the reduction in the variance is lower than what would be obtained from averaging independent estimates. **Random forests** try to reduce prediction variance even further by decorrelating the individual trees.

  - In the case where there is a single predictor X_j strongly related with the response Y, binary recursive splitting will produce bootstrapped trees $\hat{f}^{b^*}$ that likely have X_j as the top split, leading to very similar trees as well as to very similar predictions.

  - For each split on each tree produced for the bootstrapped sub-samples, random forests only consider a random subset of m predictors from the whole set of p predictors, where usually $m = \sqrt{p}$. By doing so, random forests force the trees to be different from each other, creating an additional source of variability in the fitting procedure so as to reduce the prediction variance.

  - Random forest is, then, a more general version of bagging, since this is a special case of random forest when $m = p$. Besides, as with bagging, also for random forests the number B of trees that compose the aggregate model can be set as large as possible in order to minimize test set error estimates without the risk of incurring in overfitting.

- **Boosting** is another general approach to improve predictive accuracy of statistical learning methods. Similar to bagging and random forests, boosting also estimates B different trees, but now these trees

are constructed sequentially, using for the construction of a tree information on the previous one. The process of *slow learning* uses the residuals from the estimation of the previous tree as the response when fitting the current tree. Then, once a new tree is fitted, this is aggregated into the previous. It is useful to define each of this individual trees to be very small, so the learning process is slow and extracts most information from the data.

– **Procedure to implement boosting:**

1. Define $\hat{f}(x) = 0$ to all training observations x, so that each residual equals the response $r_i = y_i$.
2. For $b \in \{1, 2, ..., B\}$:
   2.1 Fit a tree $\hat{f}^b$ with d splits, or $d+1$ terminal nodes using X as predictors and r as the response.
   2.2 Update $\hat{f}$ by adding a shrunken version of $\hat{f}^b$: $\hat{f} + \lambda\hat{f}^b$.
   2.3 Calculate the residuals r from the updated $\hat{f}$.
3. Finally, define the output boosted model:

$$\hat{f}(x) = \sum_{b=1}^{B} \lambda\hat{f}^b(x) \tag{64}$$

– From the procedure above, follow three main parameters to be set when boosting is used. First, the number of trees B now can imply in overfitting, so cross-validation is an option to define the best value for B. The shrinkage parameter λ, a small positive number (usually 0.01 or 0.001), controls the learning speed, and the smaller λ, the larger B should be in order to guarantee good performance. The last parameter is the number of splits in each individual tree, d. In most cases, $d = 1$, and each tree $\hat{f}^b$ has only two terminal nodes and is based in only one predictor, thus leading to an additive model. Consequently, parameter d controls the interaction depth of the individual trees, because a tree with d splits can not have more than d different predictors.

8 Support vector machines

8.1 SVM methods

- This set of classification methods relies on separating hyperplanes that create perfect (*maximal margin classifier*) and non perfect (*support vector classifier* and *support vector machine*) splits of the feature space into regions concerning to each class for the qualitative response variable. Besides, the boundaries defined by those splits can be either linear (*maximal margin classifier* and *support vector classifier*) or non-linear (*support vector machine*). All these methods are developed to the two-class setting, but may also be extended to cases where there are more than two distinct classes.

8.2 Maximal margin classifier

- SVM methods are based on the notion of separating hyperplanes. First of all, a hyperplane is a generalization of the two-dimensional object that is a *line* and of the three-dimensional object defined as a *plane*. Therefore, considering the p-dimensional space, the following expression defines a **hyperplane**:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p = 0 \tag{65}$$

  The expression in (65) creates an affine subspace of dimension $p - 1$ in the p-dimensional space. Any p-dimensional vector X that satisfies (65) is located precisely on the hyperplane.

- Given n training observations located in the p-dimensional space created by the set of predictors X, if there exists a hyperplane that perfectly separates all those data points into the two classes of the binary response variable Y, then that geometrical object constitutes a **separating hyperplane** figure 8.2 that algebraically satisfies the following properties for any observation i:

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip} > 0 \text{ if } y_i = 1 \tag{66}$$

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip} < 0 \text{ if } y_i = -1 \tag{67}$$

  Where $y_i = 1$ refers to a given class of Y, and $y_i = -1$ refers to the remaining class. Alternatively, a separating hyperplane defines for all observations:

$$y_i.(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip}) > 0 \tag{68}$$

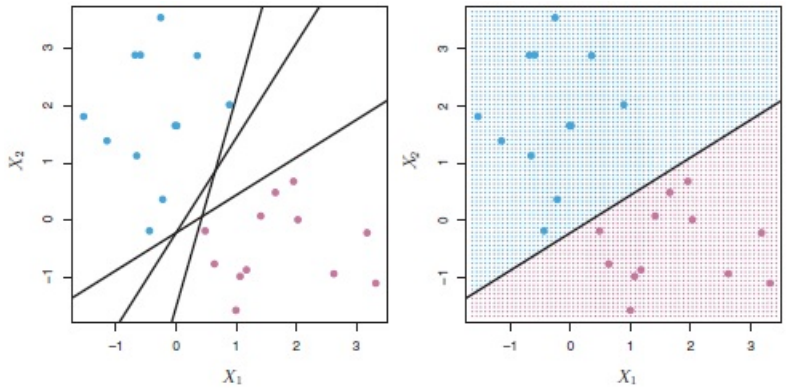

FIGURE 9.2. Left: *There are two classes of observations, shown in blue and in purple, each of which has measurements on two variables. Three separating hyperplanes, out of many possible, are shown in black.* Right: *A separating hyperplane is shown in black. The blue and purple grid indicates the decision rule made by a classifier based on this separating hyperplane: a test observation that falls in the blue portion of the grid will be assigned to the blue class, and a test observation that falls into the purple portion of the grid will be assigned to the purple class.*

Figure 8.1: Two-dimensional separating hyperplanes

- A test observation x^* is classified according with the sign of the function $f(x^*) = \beta_0 + \beta_1 x_1^* + \beta_2 x_2^* + ... + \beta_p x_p^*$. If $f(x^*) > 0$, then x^* is assigned to $Y = 1$, while if $f(x^*) < 0$, then x^* is defined to have $Y = -1$. The magnitude of $f(x^*)$ suggests how far an observation is from the group of data points belonging to the other class of Y.

- For analytical reasons, if there exists a separating hyperplane for a given data set, then an indefinite number of different separating hyperplanes also can be used to perfectly separates the observations. The **maximal margin classifier** consists on a separating hyperplane whose distance to the closest data point is maximized.

- Formally, from all perpendicular distances of data points to the hyperplane, there is a minimal one which is called *margin*. The optimal separating hyperplane is that whose distance to the hyperplane is the largest, justifying the definition of maximal margin classifier. Given that β_0, β_1, ..., β_p are the coefficients of the *maximal margin hyperplane*, then the sign of $f(x^*) = \beta_0 + \beta_1 x_1^* + ... + \beta_p x_p^*$ defines the side of the hyperplane from where x^* belongs, and thus defines the class of x^*.

- Another geometrical interpretation of separating hyperplanes that follows from its definitions points to the fact that a separating hyperplane lies on the midpoint of the space defined by two parallel

hyperplanes who tangency the closest points from the two classes (figure 8.2). The data points that lies in those parallel hyperplanes and that are equidistant from the separating hyperplane are defined as **support vectors**, since they are data points (vectors) whose location are crucial for defining the maximal margin hyperplane. Changing the location of any other data point does not alter the optimal separating hyperplane, given that it does not cross the separating boundary.

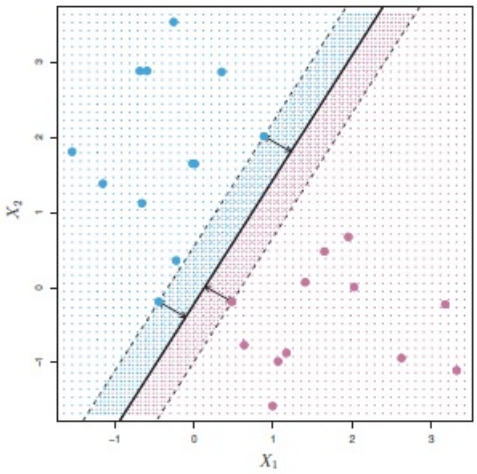

FIGURE 9.3. *There are two classes of observations, shown in blue and in purple. The maximal margin hyperplane is shown as a solid line. The margin is the distance from the solid line to either of the dashed lines. The two blue points and the purple point that lie on the dashed lines are the support vectors, and the distance from those points to the hyperplane is indicated by arrows. The purple and blue grid indicates the decision rule made by a classifier based on this separating hyperplane.*

Figure 8.2: Two-dimensional separating hyperplane

- Formally, considering n training observations x_1, x_2, ..., $x_n \in \mathbb{R}^p$ with its associated classes y_1, y_2, ..., y_n, constructing a maximal margin hyperplane is equivalent to finding those values of β_0, β_1, ..., β_p that are solution for:

$$\max_{\beta_0,\beta_1,...,\beta_p} M \tag{69}$$

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1 \tag{70}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip}) \geq M \text{ for all i} \tag{71}$$

The restriction in (71) with $M > 0$ requires that each data point is located in the correct side of the hyperplane, while (70) implies that the distance of any point to the hyperplane is equal to the expression in the left-hand side of (71). Therefore, both (70) and (71) requires that the parameters chosen lead to observations correctly located and with a distance from the hyperplane of at least M, which thus represents the margin and that should be maximized in (69).

- The optimization problem given by (69)-(71) not necessarily have solution, if the data points are disposed as in figure 8.2. Consequently, a generalization of maximal margin classifier is need for when perfect separation is not possible.

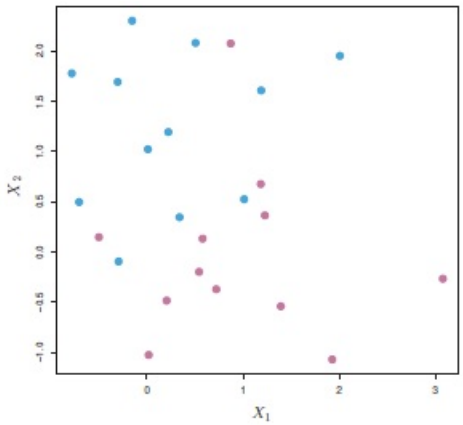

FIGURE 9.4. *There are two classes of observations, shown in blue and in purple. In this case, the two classes are not separable by a hyperplane, and so the maximal margin classifier cannot be used.*

Figure 8.3: Data points not perfectly separated

8.3 Support vector classifier

- The impossibility of a perfect separation is just one of the reasons to develop a classifier who allows for some wrong classification. Maximal margin classifier, for example, fits perfectly the training data, but is highly sensible to the observations used to construct the decision boundary and may not account for the precise location of test observations in the predictor space, i.e., maximal margin classifier is likely to imply in overfitting.

- The **support vector classifier**, or soft margin classifier, allows that some training data points to be located in the incorrect side of the margin, or even in the incorrect side of the hyperplane. So,

the classifier is more robust to individual observations, having higher bias, but lower variance in its estimations.

- The definition of a hyperplane that separates correctly most of the training observations, but that misclassify some of them leads to the following optimization problem:

$$\max_{\beta_0,\beta_1,...,\beta_p,\epsilon_1,...,\epsilon_n} M \tag{72}$$

$$\sum_{j=1}^{p} \beta_j^2 = 1 \tag{73}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip}) \geq M(1 - \epsilon_i) \text{ for all i} \tag{74}$$

$$\epsilon_i \geq 0, \sum_{i=1}^{n} \epsilon_i \leq C \tag{75}$$

  Where $C \geq 0$ and M is still positive and represents the margin. As before, given a test observation x^*, it is assigned to one of the two classes of Y depending on the signal of $f(x^*) = \beta_0 + \beta_1 x_1^* + ... + \beta_p x_p^*$.

  - Considering the restrictions (74) and (75), ϵ_i is a *slack variable* that allows a training observation i to be in the incorrect side of the margin ($\epsilon_i > 0$) or in the incorrect side of the hyperplane ($\epsilon_i > 1$).
  - The non-negative tunning parameter C sets a limit to the number of misclassified training observations, since (75) implies that no more than C observations can be in the correct side of the hyperplane. In fact, if $C = 0$, then $\epsilon_i = 0 \ \forall \ i$, and (72)-(75) is equivalent to (69)-(71). The larger C, the wider the boundary around the hyperplane will be, and the more training observations can be misclassified (figure 8.3).
  - Parameter C is usually defined through cross-validation, since it controls the bias-variance trade-off. A small value for C leads to a classifier with good performance on training data, given that few misclassification is allowed, thus representing a classifier with low bias. At the same time, the hyperplane becomes more dependent on a few support vectors, making the classifier to have high variance. Analogously, if C is large, then more errors are made on classifying training observations and more support vectors give robustness for the fitted decision boundary, so a high bias, but low variance classifier emerges.

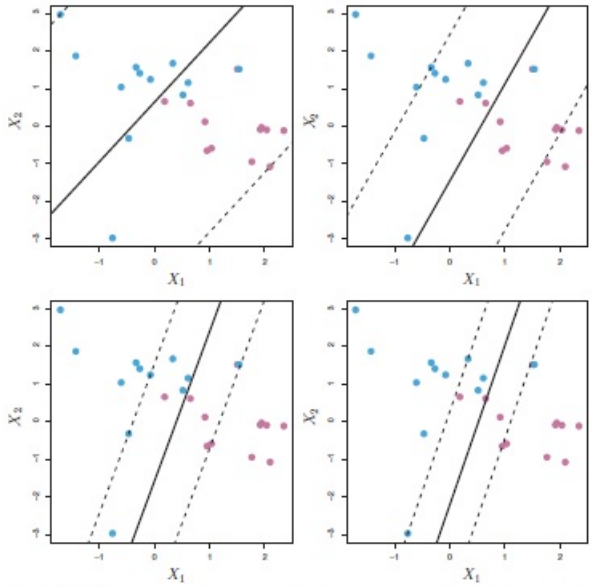

FIGURE 9.7. *A support vector classifier was fit using four different values of the tuning parameter C in (9.12)–(9.15). The largest value of C was used in the top left panel, and smaller values were used in the top right, bottom left, and bottom right panels. When C is large, then there is a high tolerance for observations being on the wrong side of the margin, and so the margin will be large. As C decreases, the tolerance for observations being on the wrong side of the margin decreases, and the margin narrows.*

Figure 8.4: Hyperplanes for different values of C

– The existence of more support vectors for support vector classifier in comparison to the number of support vectors for maximal margin classifier follows from the fact that the hyperplane is sensitive not only to training observations that is located on the margin hyperplane, but also to observations that are in the incorrect side of the margin. Therefore, parameter C controls the flexibility of the fitted support vector classifier and this is equivalent to controlling the number of support vectors.

– Maximal margin and support vector classifiers rely essentially on the training data points very close to the decision boundary, having the farthest observations few importance in the fitted classifier. Methods with this property, also shared by logistic regression, are more robust than methods sensitive to observations easily separated from data of the other class, as occurs with LDA.

8.4 Support vector machines

- Support vector classifier consists on an extension from maximal margin classifier since the first allows some misclassification of training observations so a more robust hyperplane can be constructed. Now, since both previously discussed classifiers imply in linear decision boundaries, a more general approach also considers the possibility of non-linearities in the relationship between predictors and response. In such a case, the distribution of data points does not make hyperplanes a good option when separating classes (8.4).

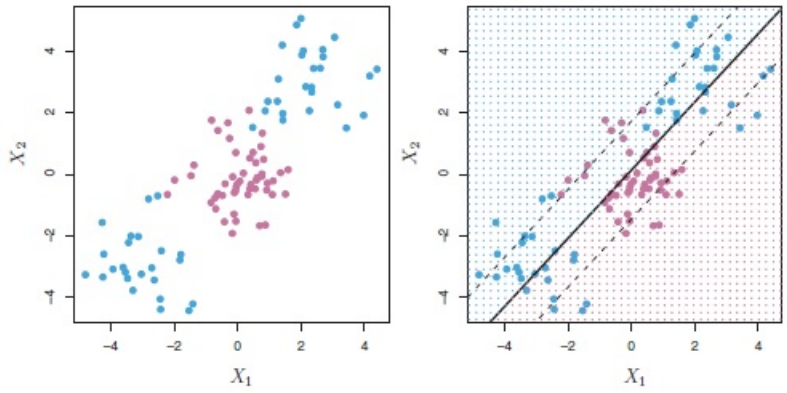

FIGURE 9.8. Left: *The observations fall into two classes, with a non-linear boundary between them.* Right: *The support vector classifier seeks a linear boundary, and consequently performs very poorly.*

Figure 8.5: Non-linear data

- As applies with linear regression models, expanding the predictor space by incorporating polynomial functions of X_j is a natural alternative to capture non-linear relationships. Thus, the optimization problem of support vector classifier (72)-(75) changes to the following when quadratic functions of the original predictors are included:

$$\max_{\beta_0,\beta_{11},\beta_{12}\ldots,\beta_{p1},\beta_{p2},\epsilon_1,\ldots,\epsilon_n} M \tag{76}$$

$$\sum_{j=1}^{p}\sum_{k=1}^{2}\beta_{jk}^2 = 1 \tag{77}$$

$$y_i(\beta_0 + \sum_{j=1}^{p}\beta_{j1}x_{ij} + \sum_{j=1}^{p}\beta_{j2}x_{ij}^2) \geq M(1-\epsilon_i) \text{ for all i} \tag{78}$$

$$\epsilon_i \geq 0, \sum_{i=1}^{n} \epsilon_i \leq C \tag{79}$$

Polynomial functions of higher order and that also includes interaction terms add even more flexibility to the non-linear decision boundary.

- The **support vector machine (SVM)** is an extension of support vector classifier that incorporates non-linearities and that is computationally efficient.

  – The equation of the hyperplane of support vector classifiers can be expressed in terms of inner products of the observations:

  $$f(x) = \beta_0 + \sum_{i=1}^{n} \alpha_i < x, x_i > \tag{80}$$

  Where $< x, x_i >= \sum_{j=1}^{p} x_j x_{ij}$. Given that α_i is non-zero only for support vectors x_i, then (80) is expressed by:

  $$f(x) = \beta_0 + \sum_{i \in S} \alpha_i < x, x_i > \tag{81}$$

  Where S is the set of support vectors.

  – The inner product $< x, x_i >$ is a measure of similarity between vectors x and x_i, and can be generalized through the use of **kernels**, $K(x_i, x_{i'})$. In fact, $K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij} x_{i'j}$ is a *linear kernel*, having as an alternative the *polynomial kernel*:

  $$K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p} x_{ij} x_{i'j})^d \tag{82}$$

  Another non-linear alternative to $K(x_i, x_{i'})$ is the **radial kernel**:

  $$K(x_i, x_{i'}) = \exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2) \tag{83}$$

  Where $\gamma > 0$. Using non-linear kernels such as (82) and (83) in the support vector classifier (81) leads to the support vector machine (figure 8.4):

  $$f(x) = \beta_0 + \sum_{i \in S} \alpha_i K(x, x_i) \tag{84}$$

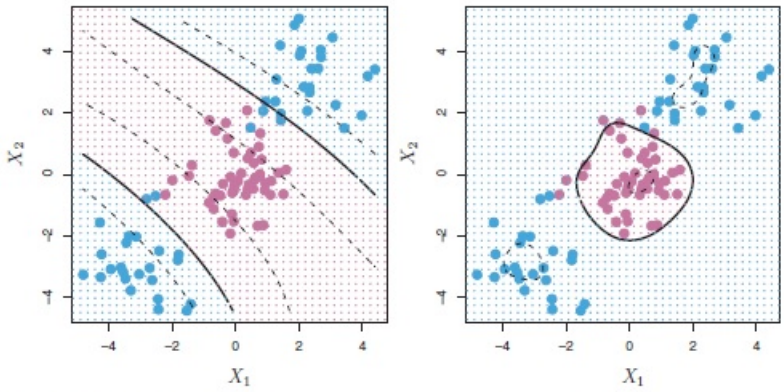

FIGURE 9.9. Left: *An SVM with a polynomial kernel of degree 3 is applied to the non-linear data from Figure 9.8, resulting in a far more appropriate decision rule.* Right: *An SVM with a radial kernel is applied. In this example, either kernel is capable of capturing the decision boundary.*

Figure 8.6: Support vector machines

- Considering the radial kernel SVM, for a given test observation x^*, the farther a given training observation x_i is from x^*, the smaller (83) will be. Thus, the training observation x_i will have a small relevance in $f(x^*)$ for distant test observations. Consequently, the radial kernel SVM has a local behavior, and the parameter γ regulates this sensitivity, given that the larger γ the more local and flexible the SVM will be.

8.5 SVMs with more than two classes

- Maximal margin classifier, support vector classifier and support vector machine are all developed considering the qualitative response variable Y as being binary. There are two main approaches to make predictions from SVMs methods for response variables with more than two classes.
  - The **one-versus-one** classification, or all pairs approach, estimates all $K(K-1)/2$ different SVMs from all pairs of categories. Then, a given test observation is assigned to the category that most frequently receives the observation in the binary classifications performed.
  - The **one-versus-all** classification estimates K different SVMs opposing each class k to a joint category that aggregates all remaining classes. A test observation is assigned to the class k for which $f(x^*) = \beta_{0k} + \beta_{1k}x_1^* + ... + \beta_{pk}x_p^*$ is the largest, where β_{0k}, β_{1k}, ..., β_{pk} are the coefficients of the SVM classifier that opposes k to all other classes.

9 Unsupervised learning

9.1 The challenge of unsupervised learning

- Supervised learning problems present two kinds of measures for each of n available observations: features X_1, X_2, ..., X_p, and a response Y. Consequently, this kind of statistical learning problem usually has the goal of predicting Y from the features X_1, X_2, ..., X_p. In **unsupervised learning** problems, there is only a set of features X_1, X_2, ..., X_p, with no distinguishable response to be predicted. So, the dataset in this context should be explored in order to find patterns in the data, helping to further understand the relationship among the set of features.

- The subjective character of unsupervised learning problems leads to the absence of performance metrics for assessing the quality of fit. Thus, the goals of finding patterns in data make the domain knowledge fundamental when analyzing the results.

9.2 Principal components analysis

- PCA was already discussed in dimension reduction methods when the information contained in a set of features is summarized as much as possible using only a few elements, so a smaller number of predictors could be used when performing linear regression.

- Besides principal components regression, exploratory data analysis can also make use of PCA, since when there are a large number of features, data visualization may become cumbersome. As PCA reduces data dimensionality by grouping most information in only a few elements, data visualization can be based on the first M principal components from p predictors with $p \gg M$.

- As introduced in subsection 5.4, the **first principal component** from a set of features X_1, X_2, ..., X_p is given by:

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + ... + \phi_{p1}X_p \tag{85}$$

  Where ϕ_{11}, ϕ_{21}, ..., ϕ_{p1} is defined so as to maximize the variance of (85) subject to $\sum_j \phi_{j1}^2 = 1$. The objective of maximizing the variance of linear combinations of features reflects the fact that they should capture the most variability in data. Geometrically, the **loadings vector** $\phi_1 = (\phi_{11}, \phi_{21}, ..., \phi_{p1})$ configures a direction in feature space along which the data vary the most.

- Given the sample version of (85), the optimization problem that results in estimates for the loadings is expressed by:

$$\max_{\phi_{11},\phi_{21},...,\phi_{p1}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \Big(\sum_{j=1}^{p} \phi_{j1} x_{ij} \Big)^2 \right\} \text{ subject to } \sum_{j=1}^{p} \phi_{j1}^2 = 1 \tag{86}$$

  Once estimated the loadings vector ϕ_1, the observations can be projected into the line that follows the direction defined by ϕ_1, and these projections are given by the **scores vector** $z_1 = (z_{11}, z_{21}, ..., z_{n1})$, which contains the first principal component for each observation. It is worth notice that the projection of each observation into the line with direction given by ϕ_1 implies in the smallest possible overall distance between the data points and a line that passes through them.

- After calculated the first principal component z_{i1}, the second principal component z_{i2} derives from an optimization problem similar to (86) with an additional restriction that the loadings vectors ϕ_1 and ϕ_2 must be orthogonal to each other, i.e., z_{i1} and z_{i2} must be uncorrelated. From this, follow both a loadings vector ϕ_2 and a scores vector z_2 associated with the second principal component. This can be extended further up to the M-th principal component, where $M \leq \min((n-1), p)$.

- Plotting the scores vectors z_1 and z_2 together with the loadings ϕ_{j1} and ϕ_{j2} in a *biplot* helps to understand the relationship between the original features. The size and direction of vectors (ϕ_{j1}, ϕ_{j2}) indicate the relative importance of the predictor j to the construction of each principal components. Then, patterns can be understood by checking the position of scores for each data point (figure 9.2).

- Each loadings vector ϕ_m, with $m \in \{1, 2, ..., M\}$ captures the direction in which the data vary the most, and geometrically defines a line in the p-dimensional space whose distance to the observations is minimized (figure 5.4). The scores vector z_m presents for each observation the location of its projection into the line given by ϕ_m, considering the principal components space (figure 5.4).

- All directions ϕ_m are orthogonal by construction, and considering the loadings vectors of the first two principal components ϕ_1 and ϕ_2, for example, they define a plane in the p-dimensional space with minimal distance from the data points, similar to the case when only a single direction is considered. Also here the scores can be projected into the plane that follows from ϕ_1 and ϕ_2 when the principal components space is considered (figure 9.2).

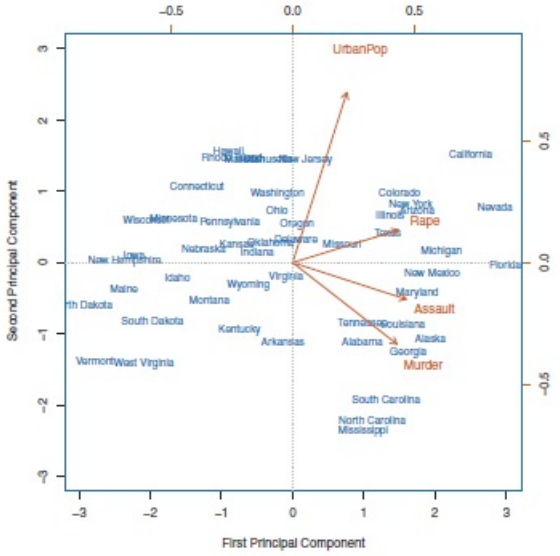

FIGURE 10.1. *The first two principal components for the* `USArrests` *data. The blue state names represent the scores for the first two principal components. The orange arrows indicate the first two principal component loading vectors (with axes on the top and right). For example, the loading for* `Rape` *on the first component is* 0.54*, and its loading on the second principal component* 0.17 *(the word* `Rape` *is centered at the point* $(0.54, 0.17)$*). This figure is known as a biplot, because it displays both the principal component scores and the principal component loadings.*

Figure 9.1: Example of a biplot

- More generally, the first M principal components scores and loadings lead to the best M-dimensional approximation of each x_{ij}:

$$x_{ij} \approx \sum_{m=1}^{M} \phi_{jm} z_{im} \tag{87}$$

- Standardizing the variables is crucial when performing PCA, given that the results vary widely when the scales are changed. Since all results follow from maximizing the variance of projections, those variables X_j with high variance will be given large values of loading ϕ_{jm}. If variables are all measured in a same unit, then data in its original scale can be used without any undesired consequence.

- Both loadings ϕ_{jm} and scores z_{im} are unique for a given dataset, except for their signals that can vary as a result of the fact that a direction is preserved when all the elements from the corresponding vector are multiplied by -1.

- **Proportion of variance explained (PVE):** the main goal of PCA is to reduce the dimensionality of data without loosing too much of information. Thus, it is relevant to assess how much of the

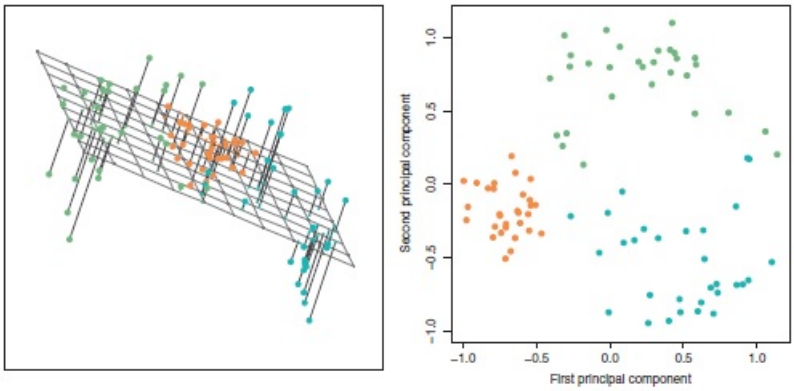

FIGURE 10.2. *Ninety observations simulated in three dimensions.* Left: *the first two principal component directions span the plane that best fits the data. It minimizes the sum of squared distances from each point to the plane.* Right: *the first two principal component score vectors give the coordinates of the projection of the 90 observations onto the plane. The variance in the plane is maximized.*

Figure 9.2: First two principal components directions and projections

variance in data that is captured by the calculated principal components. Given a mean-zero data, the PVE for the principal component m is defined as:

$$\frac{(1/n)\sum_{i=1}^{n} z_{im}^2}{\sum_{j=1}^{p}(1/n)\sum_{i=1}^{n} x_{ij}^2} = \frac{\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{jm}x_{ij}\right)^2}{\sum_{j=1}^{p}\sum_{i=1}^{n} x_{ij}^2} \tag{88}$$

The cumulative PVE for all first M principal components follows from the sum of (88) over all $m \in \{1, 2, ..., M\}$.

- Except for the use of PCA for supervised learning problems, when cross-validation is an alternative, the definition of the appropriate value for M is subjective in the context of unsupervised learning. The ideal is to select the smallest number of principal components that capture a sufficient amount of variability in data. The *scree plot* is an alternative to define M by plotting individual and cumulative PVE against M. A good choice for M follows from the identification of where an elbow is formed in the curve, indicating that no more significant gain in variability is achieved. Alternatively, the estimation of principal components continues up to the point where no more patterns are found in data.

9.3 Clustering methods

- Clustering defines subgroups, or clusters from data points that are similar within each of those clusters and different across alternative clusters. As with PCA, clustering is a tool for discovering patterns in data by reducing all dimensions into a small set of created elements. More generally, either observations can be gathered into clusters on the basis of features or features can be grouped into clusters on the basis of observations, even that the prior is more common in practice.

- **K-means clustering:** requires the prior definition of the number of clusters K in which the data will be classified. These non-overlapping clusters are defined as C_1, C_2, ..., C_K and receive indices of observations in such a way that $C_1 \cup C_2 \cup ... \cup C_K = \{1, 2, ..., n\}$ and $C_k \cap C_{k'} =$.

  - K-means clustering tries to allocate each observation so that the **within-cluster variation** $W(C_K)$ is as small as possible, indicating that all observations inside a cluster are similar to each other. Formally:

$$\min_{C_1, C_2, ..., C_K} \left\{ \sum_{k=1}^{K} \frac{1}{|C_K|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\} \tag{89}$$

  - In order to implement a feasible solution to (89), the following algorithm produces a local minimum for the aggregated within-cluster variation.

    * **K-means clustering algorithm:**

      1. Randomly assign a number from 1 to K for each observation, which serves as a initial cluster assignment.
      2. Iterates the following procedure until no more assignments are made:

      1.1 For each cluster, calculate its centroid, i.e., the vector with the within-cluster average of each feature.

      1.2 Assign each observation to the cluster whose centroid is closest.

  - Since step 1 is random, each repetition of K-means clustering produces a different result. Then, it is recommended to repeat the procedure several times and to select the clustering for which the objective function (89) is smallest.

- **Hierarchical clustering:** differently from K-means clustering, this approach does not need a prior definition of the number of subgroups in which the observations will be classified. The **agglomerative clustering**, or bottom-up, implies in a graphical representation similar to that of decision

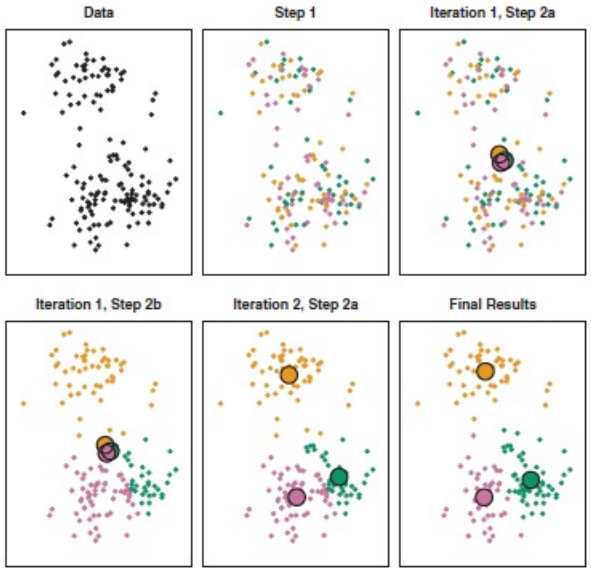

FIGURE 10.6. *The progress of the K-means algorithm on the example of Figure 10.5 with K=3.* Top left: *the observations are shown.* Top center: *in Step 1 of the algorithm, each observation is randomly assigned to a cluster.* Top right: *in Step 2(a), the cluster centroids are computed. These are shown as large colored disks. Initially the centroids are almost completely overlapping because the initial cluster assignments were chosen at random.* Bottom left: *in Step 2(b), each observation is assigned to the nearest centroid.* Bottom center: *Step 2(a) is once again performed, leading to new cluster centroids.* Bottom right: *the results obtained after ten iterations.*

Figure 9.3: K-means clustering algorithm

trees, so that it constitutes a more flexible and more visually appealing approach in comparison to K-means.

- A **dendrogram** (figure 9.3) is a tree whose leaves correspond to the observations. These terminal nodes are then fused when the corresponding observations are similar to each other. Graphically, this means that branches are grouped forming an internal node. This process of fusing continues to aggregate similar data points until all of them are united into a single initial node.
- The lower in the dendrogram the fusion occurs, the more similar the observations grouped are. Thus, the vertical distance between two observations indicates how similar they are, differently from the horizontal distance, which does not reveal anything about the data points (figure 9.3).

- The procedure to identifying clusters consider horizontal cuts to the dendrogram (figure 9.3). Once a horizontal line is defined within the figure, each of the first internal nodes after the

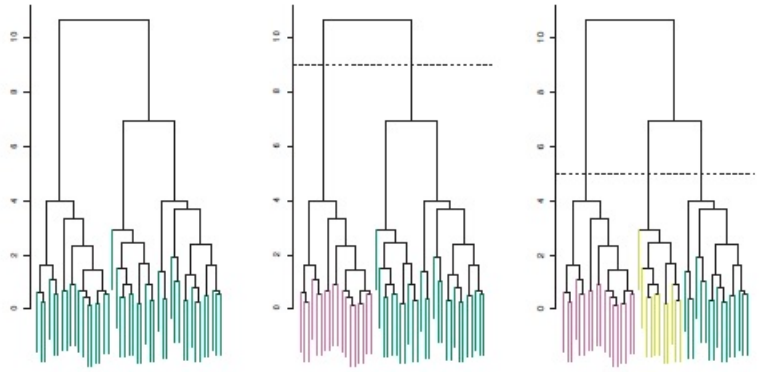

FIGURE 10.9. Left: *dendrogram obtained from hierarchically clustering the data from Figure 10.8 with complete linkage and Euclidean distance.* Center: *the dendrogram from the left-hand panel, cut at a height of nine (indicated by the dashed line). This cut results in two distinct clusters, shown in different colors.* Right: *the dendrogram from the left-hand panel, now cut at a height of five. This cut results in three distinct clusters, shown in different colors. Note that the colors were not used in clustering, but are simply used for display purposes in this figure.*

Figure 9.4: Dendrograms

cut defines clusters (two in the center and three in the right of figure 9.3). Thus, hierarchical clustering is more flexible then K-means clustering in the sense that, after fitting a dendrogram to the data, any number of clusters between 1 (no cut) and n (each observation being a cluster) can be defined. Even so, the height of the cut is similar to the parameter K in K-means.

– The choice of the cut into the dendrogram is mainly arbitrary, since visual inspection and domain knowledge should lead this definition. It is also worth notice that depending on the dataset at hand, hierarchical clustering may not be a good alternative for unsupervised learning. As the name says, *hierarchical* clustering relies on the assumption that some categories in which data can be classified is precedent to others.

– The algorithm to fit a dendrogram to a dataset first defines a measure for dissimilarity between each pair of observations, being the Euclidean distance usually the main choice. So, all n observations are considered a unique cluster, and for each one of the $n(n-1)/2$ pairs of observations available their within-distance is calculated, where that pair corresponding to the minimal distance is considered a single cluster, thus remaining $n-1$ clusters. From these, the distance among each of the clusters are again calculated, and those two closest to each other are now considered a new unified cluster in a set of $n-2$ clusters. This process continues until all observations belong to a same single cluster.

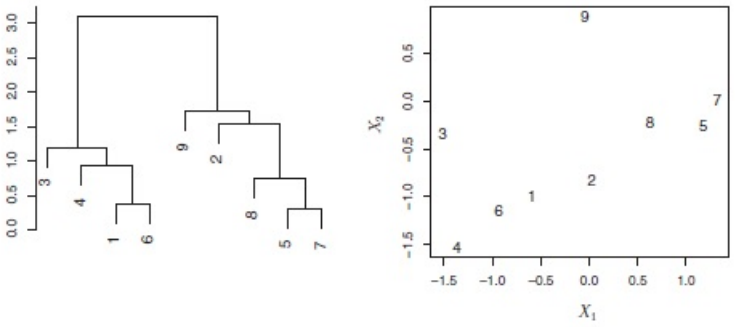

FIGURE 10.10. *An illustration of how to properly interpret a dendrogram with nine observations in two-dimensional space.* Left: *a dendrogram generated using Euclidean distance and complete linkage. Observations* 5 *and* 7 *are quite similar to each other, as are observations* 1 *and* 6. *However, observation* 9 *is* no more similar to *observation* 2 *than it is to observations* 8, 5, *and* 7, *even though observations* 9 *and* 2 *are close together in terms of horizontal distance. This is because observations* 2, 8, 5, *and* 7 *all fuse with observation* 9 *at the same height, approximately* 1.8. Right: *the raw data used to generate the dendrogram can be used to confirm that indeed, observation* 9 *is no more similar to observation* 2 *than it is to observations* 8. 5. *and* 7.

Figure 9.5: Dendrogram and vertical and horizontal distances

∗ **Hierarchical clustering algorithm:**

1. Given n data points and a measure of distance, each observation is considered a cluster, and $n(n-1)/2$ pairwise dissimilarities are calculated among all pairs of observations.
2. For $i \in \{n, n-1, ..., 2\}$:
   a) From all i clusters, the pair of clusters with minimal dissimilarity is fused into a single cluster.
   b) Calculate the new pairwise inter-cluster dissimilarities among the $i-1$ remaining clusters.

– If the dissimilarity measure for two clusters containing each only one observation is straightforward to calculate, alternative methods exist to define the distance between two clusters where one or both have multiple observations. This leads to the question of **linkage** between clusters, and the four main options for calculating their distances are displayed in table 9.3. In general, average and complete linkages are preferred over single linkage, and the centroid is less appealing given its more complex visualization.

– The resultant dendrogram is highly dependent on the dissimilarity measure and on the linkage type chosen to calculate the distances in step 2 b) of the above-presented algorithm.

Linkage	*Description*
Complete	Maximal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *largest* of these dissimilarities.
Single	Minimal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *smallest* of these dissimilarities. Single linkage can result in extended, trailing clusters in which single observations are fused one-at-a-time.
Average	Mean intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *average* of these dissimilarities.
Centroid	Dissimilarity between the centroid for cluster A (a mean vector of length p) and the centroid for cluster B. Centroid linkage can result in undesirable *inversions*.

TABLE 10.2. *A summary of the four most commonly-used types of linkage in hierarchical clustering.*

Figure 9.6: Linkage types

– Concerning dissimilarity measures, an alternative to Euclidean distance is the *correlation-based distance*, which calculates the correlation among features vectors of two different observations. This may be relevant for cases where the patterns of distribution in data (positive versus non-positive values) are more important than the magnitudes assumed by features. In general, the type of data available and the scientific question to be answered may guide the choice between Euclidean and correlation-based distances.

– Given that distances are calculated in both presented clustering methods, the scale of the variables is crucial for the results obtained. When variables have different units of measurements, and more generally, when they have very different scales, the standardization of the variables in order to they have variance equals to one should be considered.

– Transforming the variables to have mean zero and standard deviation one applies both to K-means and hierarchical clustering. Considering the first solely, K is the key parameter when implementing that approach. Considering the later, the dissimilarity measure, the type of linkage and the height of the cut into the dendrogram are the main decisions to be made.

– A good practice is to define a grid of such parameters and to try different specifications, so alternative results can be compared and the most interesting clustering should be chosen.

– There are ways to calculate p-values so the clustering obtained can be assessed concerning the validity of the classification achieved.

– Clustering methods are very sensitive to the dataset used, and also to the presence of outliers (observations very different from the others, even that they do not constitute a separate sub-group). In the later case, *mixture models* are available so those outliers are accommodated without the construction of an additional cluster.

– Finally, the robustness of the clustering obtained can be assess through the classification of data using subsets of the entire dataset. Besides, irrespective of how much robust a clustering can be, it never shows an absolute true concerning the data structure.