

change in the **pop** dimension, relative to PCA. This suggests that **pop** is more highly correlated with the response than is **ad**. The PLS direction does not fit the predictors as closely as does PCA, but it does a better job explaining the response.

To identify the second PLS direction we first *adjust* each of the variables for  $Z_1$ , by regressing each variable on  $Z_1$  and taking *residuals*. These residuals can be interpreted as the remaining information that has not been explained by the first PLS direction. We then compute  $Z_2$  using this *orthogonalized* data in exactly the same fashion as  $Z_1$  was computed based on the original data. This iterative approach can be repeated  $M$  times to identify multiple PLS components  $Z_1, \dots, Z_M$ . Finally, at the end of this procedure, we use least squares to fit a linear model to predict  $Y$  using  $Z_1, \dots, Z_M$  in exactly the same fashion as for PCR.

As with PCR, the number  $M$  of partial least squares directions used in PLS is a tuning parameter that is typically chosen by cross-validation. We generally standardize the predictors and response before performing PLS.

PLS is popular in the field of chemometrics, where many variables arise from digitized spectrometry signals. In practice it often performs no better than ridge regression or PCR. While the supervised dimension reduction of PLS can reduce bias, it also has the potential to increase variance, so that the overall benefit of PLS relative to PCR is a wash.

## 6.4 Considerations in High Dimensions

### 6.4.1 High-Dimensional Data

Most traditional statistical techniques for regression and classification are intended for the *low-dimensional* setting in which  $n$ , the number of observations, is much greater than  $p$ , the number of features. This is due in part to the fact that throughout most of the field's history, the bulk of scientific problems requiring the use of statistics have been low-dimensional. For instance, consider the task of developing a model to predict a patient's blood pressure on the basis of his or her age, gender, and body mass index (BMI). There are three predictors, or four if an intercept is included in the model, and perhaps several thousand patients for whom blood pressure and age, gender, and BMI are available. Hence  $n \gg p$ , and so the problem is low-dimensional. (By dimension here we are referring to the size of  $p$ .)

In the past 20 years, new technologies have changed the way that data are collected in fields as diverse as finance, marketing, and medicine. It is now commonplace to collect an almost unlimited number of feature measurements ( $p$  very large). While  $p$  can be extremely large, the number of observations  $n$  is often limited due to cost, sample availability, or other considerations. Two examples are as follows:

1. Rather than predicting blood pressure on the basis of just age, gender, and BMI, one might also collect measurements for half a million

*single nucleotide polymorphisms* (SNPs; these are individual DNA mutations that are relatively common in the population) for inclusion in the predictive model. Then  $n \approx 200$  and  $p \approx 500,000$ .

2. A marketing analyst interested in understanding people's online shopping patterns could treat as features all of the search terms entered by users of a search engine. This is sometimes known as the "bag-of-words" model. The same researcher might have access to the search histories of only a few hundred or a few thousand search engine users who have consented to share their information with the researcher. For a given user, each of the  $p$  search terms is scored present (0) or absent (1), creating a large binary feature vector. Then  $n \approx 1,000$  and  $p$  is much larger.

Data sets containing more features than observations are often referred to as *high-dimensional*. Classical approaches such as least squares linear regression are not appropriate in this setting.

high-dimensional

Many of the issues that arise in the analysis of high-dimensional data were discussed earlier in this book, since they apply also when  $n > p$ : these include the role of the bias-variance trade-off and the danger of overfitting. Though these issues are always relevant, they can become particularly important when the number of features is very large relative to the number of observations.

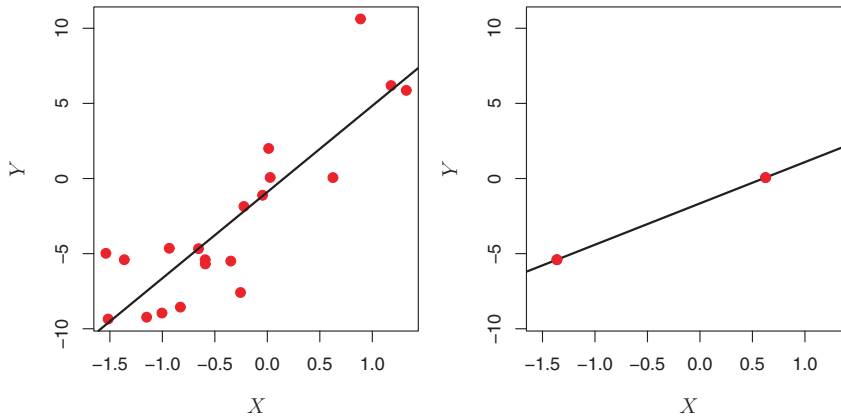
We have defined the *high-dimensional setting* as the case where the number of features  $p$  is larger than the number of observations  $n$ . But the considerations that we will now discuss certainly also apply if  $p$  is slightly smaller than  $n$ , and are best always kept in mind when performing supervised learning.

### 6.4.2 What Goes Wrong in High Dimensions?

In order to illustrate the need for extra care and specialized techniques for regression and classification when  $p > n$ , we begin by examining what can go wrong if we apply a statistical technique not intended for the high-dimensional setting. For this purpose, we examine least squares regression. But the same concepts apply to logistic regression, linear discriminant analysis, and other classical statistical approaches.

When the number of features  $p$  is as large as, or larger than, the number of observations  $n$ , least squares as described in Chapter 3 cannot (or rather, *should not*) be performed. The reason is simple: regardless of whether or not there truly is a relationship between the features and the response, least squares will yield a set of coefficient estimates that result in a perfect fit to the data, such that the residuals are zero.

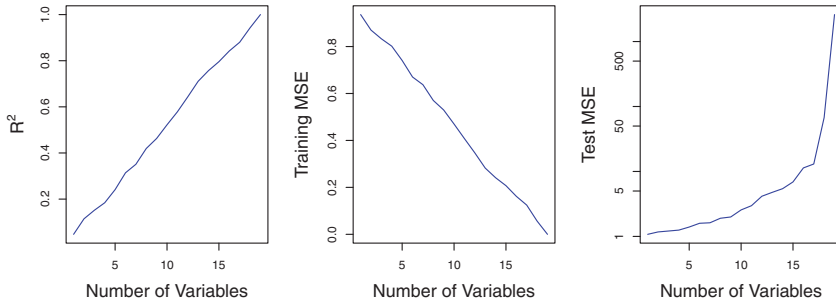
An example is shown in Figure 6.22 with  $p = 1$  feature (plus an intercept) in two cases: when there are 20 observations, and when there are only two observations. When there are 20 observations,  $n > p$  and the least



**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).*

squares regression line does not perfectly fit the data; instead, the regression line seeks to approximate the 20 observations as well as possible. On the other hand, when there are only two observations, then regardless of the values of those observations, the regression line will fit the data exactly. This is problematic because this perfect fit will almost certainly lead to overfitting of the data. In other words, though it is possible to perfectly fit the training data in the high-dimensional setting, the resulting linear model will perform extremely poorly on an independent test set, and therefore does not constitute a useful model. In fact, we can see that this happened in Figure 6.22: the least squares line obtained in the right-hand panel will perform very poorly on a test set comprised of the observations in the left-hand panel. The problem is simple: when  $p > n$  or  $p \approx n$ , a simple least squares regression line is too flexible and hence overfits the data.

Figure 6.23 further illustrates the risk of carelessly applying least squares when the number of features  $p$  is large. Data were simulated with  $n = 20$  observations, and regression was performed with between 1 and 20 features, each of which was completely unrelated to the response. As shown in the figure, the model  $R^2$  increases to 1 as the number of features included in the model increases, and correspondingly the training set MSE decreases to 0 as the number of features increases, even though the features are completely unrelated to the response. On the other hand, the MSE on an independent test set becomes extremely large as the number of features included in the model increases, because including the additional predictors leads to a vast increase in the variance of the coefficient estimates. Looking at the test set MSE, it is clear that the best model contains at most a few variables. However, someone who carelessly examines only the  $R^2$  or the training set MSE might erroneously conclude that the model with the greatest number of variables is best. This indicates the importance of applying extra care



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

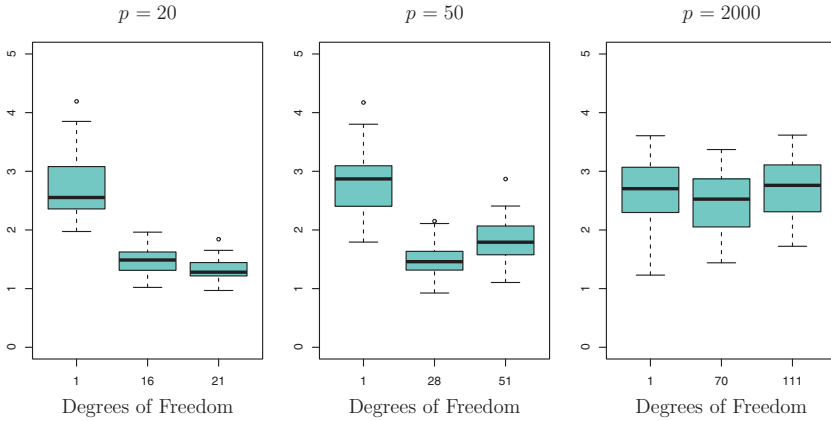
when analyzing data sets with a large number of variables, and of always evaluating model performance on an independent test set.

In Section 6.1.3, we saw a number of approaches for adjusting the training set RSS or  $R^2$  in order to account for the number of variables used to fit a least squares model. Unfortunately, the  $C_p$ , AIC, and BIC approaches are not appropriate in the high-dimensional setting, because estimating  $\hat{\sigma}^2$  is problematic. (For instance, the formula for  $\hat{\sigma}^2$  from Chapter 3 yields an estimate  $\hat{\sigma}^2 = 0$  in this setting.) Similarly, problems arise in the application of adjusted  $R^2$  in the high-dimensional setting, since one can easily obtain a model with an adjusted  $R^2$  value of 1. Clearly, alternative approaches that are better-suited to the high-dimensional setting are required.

### 6.4.3 Regression in High Dimensions

It turns out that many of the methods seen in this chapter for fitting less flexible least squares models, such as forward stepwise selection, ridge regression, the lasso, and principal components regression, are particularly useful for performing regression in the high-dimensional setting. Essentially, these approaches avoid overfitting by using a less flexible fitting approach than least squares.

Figure 6.24 illustrates the performance of the lasso in a simple simulated example. There are  $p = 20, 50$ , or  $2,000$  features, of which 20 are truly associated with the outcome. The lasso was performed on  $n = 100$  training observations, and the mean squared error was evaluated on an independent test set. As the number of features increases, the test set error increases. When  $p = 20$ , the lowest validation set error was achieved when  $\lambda$  in (6.7) was small; however, when  $p$  was larger then the lowest validation set error was achieved using a larger value of  $\lambda$ . In each boxplot, rather than reporting the values of  $\lambda$  used, the *degrees of freedom* of the resulting



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

lasso solution is displayed; this is simply the number of non-zero coefficient estimates in the lasso solution, and is a measure of the flexibility of the lasso fit. Figure 6.24 highlights three important points: (1) regularization or shrinkage plays a key role in high-dimensional problems, (2) appropriate tuning parameter selection is crucial for good predictive performance, and (3) the test error tends to increase as the dimensionality of the problem (i.e. the number of features or predictors) increases, unless the additional features are truly associated with the response.

The third point above is in fact a key principle in the analysis of high-dimensional data, which is known as the *curse of dimensionality*. One might think that as the number of features used to fit a model increases, the quality of the fitted model will increase as well. However, comparing the left-hand and right-hand panels in Figure 6.24, we see that this is not necessarily the case: in this example, the test set MSE almost doubles as  $p$  increases from 20 to 2,000. In general, adding additional signal features that are truly associated with the response will improve the fitted model, in the sense of leading to a reduction in test set error. However, adding noise features that are not truly associated with the response will lead to a deterioration in the fitted model, and consequently an increased test set error. This is because noise features increase the dimensionality of the

curse of dimensionality

problem, exacerbating the risk of overfitting (since noise features may be assigned nonzero coefficients due to chance associations with the response on the training set) without any potential upside in terms of improved test set error. Thus, we see that new technologies that allow for the collection of measurements for thousands or millions of features are a double-edged sword: they can lead to improved predictive models if these features are in fact relevant to the problem at hand, but will lead to worse results if the features are not relevant. Even if they are relevant, the variance incurred in fitting their coefficients may outweigh the reduction in bias that they bring.

#### 6.4.4 *Interpreting Results in High Dimensions*

When we perform the lasso, ridge regression, or other regression procedures in the high-dimensional setting, we must be quite cautious in the way that we report the results obtained. In Chapter 3, we learned about multicollinearity, the concept that the variables in a regression might be correlated with each other. In the high-dimensional setting, the multicollinearity problem is extreme: any variable in the model can be written as a linear combination of all of the other variables in the model. Essentially, this means that we can never know exactly which variables (if any) truly are predictive of the outcome, and we can never identify the *best* coefficients for use in the regression. At most, we can hope to assign large regression coefficients to variables that are correlated with the variables that truly are predictive of the outcome.

For instance, suppose that we are trying to predict blood pressure on the basis of half a million SNPs, and that forward stepwise selection indicates that 17 of those SNPs lead to a good predictive model on the training data. It would be incorrect to conclude that these 17 SNPs predict blood pressure more effectively than the other SNPs not included in the model. There are likely to be many sets of 17 SNPs that would predict blood pressure just as well as the selected model. If we were to obtain an independent data set and perform forward stepwise selection on that data set, we would likely obtain a model containing a different, and perhaps even non-overlapping, set of SNPs. This does not detract from the value of the model obtained—for instance, the model might turn out to be very effective in predicting blood pressure on an independent set of patients, and might be clinically useful for physicians. But we must be careful not to overstate the results obtained, and to make it clear that what we have identified is simply *one of many possible models* for predicting blood pressure, and that it must be further validated on independent data sets.

It is also important to be particularly careful in reporting errors and measures of model fit in the high-dimensional setting. We have seen that when  $p > n$ , it is easy to obtain a useless model that has zero residuals. Therefore, one should *never* use sum of squared errors, p-values,  $R^2$

statistics, or other traditional measures of model fit on the training data as evidence of a good model fit in the high-dimensional setting. For instance, as we saw in Figure 6.23, one can easily obtain a model with  $R^2 = 1$  when  $p > n$ . Reporting this fact might mislead others into thinking that a statistically valid and useful model has been obtained, whereas in fact this provides absolutely no evidence of a compelling model. It is important to instead report results on an independent test set, or cross-validation errors. For instance, the MSE or  $R^2$  on an independent test set is a valid measure of model fit, but the MSE on the training set certainly is not.

## 6.5 Lab 1: Subset Selection Methods

### 6.5.1 Best Subset Selection

Here we apply the best subset selection approach to the `Hitters` data. We wish to predict a baseball player's `Salary` on the basis of various statistics associated with performance in the previous year.

First of all, we note that the `Salary` variable is missing for some of the players. The `is.na()` function can be used to identify the missing observations. It returns a vector of the same length as the input vector, with a `TRUE` for any elements that are missing, and a `FALSE` for non-missing elements. The `sum()` function can then be used to count all of the missing elements.

`is.na()`

`sum()`

```
> library(ISLR)
> fix(Hitters)
> names(Hitters)
[1] "AtBat" "Hits" "HmRun" "Runs" "RBI"
[6] "Walks" "Years" "CAtBat" "CHits" "CHmRun"
[11] "CRuns" "CRBI" "CWalks" "League" "Division"
[16] "PutOuts" "Assists" "Errors" "Salary" "NewLeague"
> dim(Hitters)
[1] 322 20
> sum(is.na(Hitters$Salary))
[1] 59
```

Hence we see that `Salary` is missing for 59 players. The `na.omit()` function removes all of the rows that have missing values in any variable.

```
> Hitters=na.omit(Hitters)
> dim(Hitters)
[1] 263 20
> sum(is.na(Hitters))
[1] 0
```

The `regsubsets()` function (part of the `leaps` library) performs best subset selection by identifying the best model that contains a given number of predictors, where *best* is quantified using RSS. The syntax is the same as for `lm()`. The `summary()` command outputs the best set of variables for each model size.

`regsubsets()`