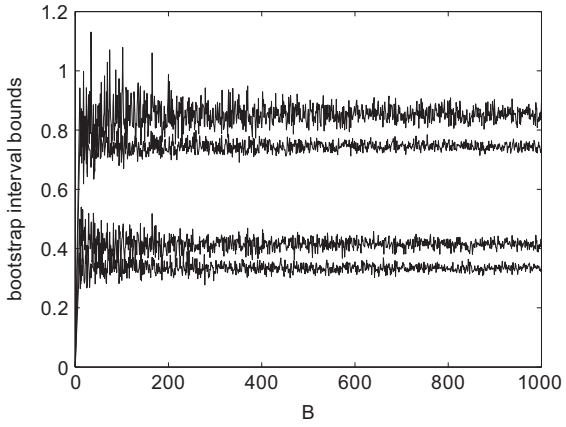

Figure 4.7 Dependence of percentile and BC_a bootstrap intervals (68% CL) on the number of bootstrap samples. The top pair of curves are the estimated upper limit of the interval, and the bottom pair the lower limit. In each pair, the upper curve is from the BC_a method, and the lower is from the percentile method. All bootstrap replicas are independent, so the local scatter of the points indicates the statistical variation from the resampling process. The sample is size 20 from N(0, 1). The sample variance is 0.568 for this sample.

Table 4.1 Comparing coverage of estimated 68% confidence intervals from the bootstrap percentile and BC_a methods.

	Percentile	**BC_a**
Target tail probability	0.1587	
Low tail probability	0.0826	0.2093
High tail probability	0.3035	0.1889
Target coverage	0.6827	
Coverage	0.6139	0.6018

4.5 Cross-Validation

In later chapters we develop a variety of classification algorithms. We can view the typical situation as one where we have a set of understood data that we wish to learn from, and thence make predictions about data with some unknown characteristic. It is important to know how good our prediction is. Another situation occurs when we wish to make a prediction in the familiar regression setting. Again, we wish to know how good our prediction is. We may attempt to provide a measure for this with the *expected prediction error* (EPE), defined below. The EPE requires knowing the sampling distribution, which is in general not available. Thus, we must find a means to estimate EPE. A technique for doing this is the resampling method known as cross-validation.

To define the expected prediction error, consider the regression problem (it could also be a classification problem, as we shall discuss in Section 9.4.1) where we wish to predict a value for random variable Y, depending on the value for random variable X. In general, X is a vector of random variables, but we will treat it as one-dimensional for the moment. The joint probability distribution for X and Y is $F(x, y)$, with pdf $f(x, y)$. We wish to find the "best" prediction for Y, given any $X = x$. That is, we look for a function $r(x)$ providing our prediction for Y. What do we mean by "best"? Well, that is up to us to decide; there are many possibilities. However, the most common choice is an estimator that minimizes the expected squared deviation, and this is how we define the expected (squared) prediction error:

$$\mathrm{EPE}(r) = E\{[Y - r(X)]^2\} = \int [y - r(x)]^2 f(x, y) dx dy \,. \tag{4.38}$$

Note that the expectation is over both X and Y; it is the expected error (squared) over the joint distribution. The function r that minimizes EPE is the regression function

$$r(x) = E(Y)_{X=x} = \int y f(x, y) dy \,, \tag{4.39}$$

that is, the conditional expectation for Y, given $X = x$. Fortunately, this is a nicely intuitive result. We remark again that x and y may be multivariate.

How might we estimate the EPE? For a simple case, consider a bivariate dataset $\mathcal{D} \equiv \{(X_n, Y_n), n = 1, \ldots, N\}$. Suppose we are interested in finding the best straight line fit. In this case, our regression function is $\ell(x) = ax + b$. We estimate parameters a and b by finding $\hat{a}$ and $\hat{b}$ that minimize (assuming equal weights for simplicity)

$$\sum_{n=1}^{N} \left(Y_n - \hat{a} X_n - \hat{b} \right)^2 \,. \tag{4.40}$$

The value of a new sampling is predicted given X_{N+1}:

$$\hat{Y}_{N+1} = \hat{a} X_{N+1} + \hat{b} \,. \tag{4.41}$$

We wish to estimate the EPE for Y_{N+1}.

A simple approach is to divide our $\{(X_i, Y_i), i = 1, \ldots, N\}$ dataset into two pieces, perhaps two halves. Then one piece (the training set) could be used to determine the regression function, and the other piece (the testing set) could be used to estimate the EPE. However, this seems a bit wasteful, since we are only using half of the available data to obtain our regression function, and we could do a better job with all of the data. The next thing that occurs to us is to reverse the roles of the two pieces and somehow average the results, and this is a pretty good idea. But let us take this to an extreme, known as *leave-one-out cross-validation*.

The algorithm for leave-one-out cross-validation is as follows:

1. Form N subsets of the dataset $\mathcal{D}$, each one leaving out a different datum, say (X_k, Y_k). We will use the subscript $-k$ to denote quantities obtained omitting datum (X_k, Y_k). Likewise, we let $\mathcal{D}_{-k}$ be the dataset leaving out (X_k, Y_k).
2. Do the regression on dataset $\mathcal{D}_{-k}$, obtaining regression function r_{-k}.
3. Using this regression predict the value for the missing point:

$$\hat{Y}_k = r_{-k}(X_k)\,. \tag{4.42}$$

4. Repeat this process for $k = 1, \ldots, N$. Estimate the EPE according to

$$\frac{1}{N}\sum_{k=1}^{N}(\hat{Y}_k - Y_k)^2\,. \tag{4.43}$$

Let us try an example application. Suppose we wish to investigate polynomial regression models for a dataset (perhaps an angular distribution or a background distribution). For example, we have a dataset and wish to consider whether to use the straight line relation

$$Y = aX + b\,, \tag{4.44}$$

or the quadratic relation

$$Y = aX^2 + bX + c\,. \tag{4.45}$$

We know that the fitted errors for the quadratic model will always be smaller than for the linear model. A common approach is to compute the sum of squared fitted errors, and apply some criterion on the difference in this quantity between the two models, often resorting to an approximation with a χ^2 distribution. That is, we use the Snedecor F distribution to compare the χ^2 values from fits for the two models. However, we may not wish to rely on the accuracy of this approximation. In this case, cross-validation may be applied.

The predictive error is not necessarily smaller with the additional adjustable parameters. We may thus use our estimated prediction errors as a means to decide between models. Suppose, for example, that our data is actually sampled from the linear model, as in the filled circles in Figure 4.8a. We do cross-validation estimates of the prediction error for both the linear and quadratic fit models, and take the difference (linear EPE minus quadratic EPE). The distribution of this difference, for 100 such "experiments", is shown in Figure 4.8b. Choosing the linear model when the difference is larger than zero gets it right in 84 out of 100 cases. The MATLAB function `crossval` is used to perform the estimates, with calls of the form:

```
crossval('mse',x,y,'Predfun',@linereg,'leaveout',1);
```

Alternatively, suppose that our data is sampled from a quadratic model (with a rather small quadratic term), as in the plus symbols in Figure 4.8a. We do cross-

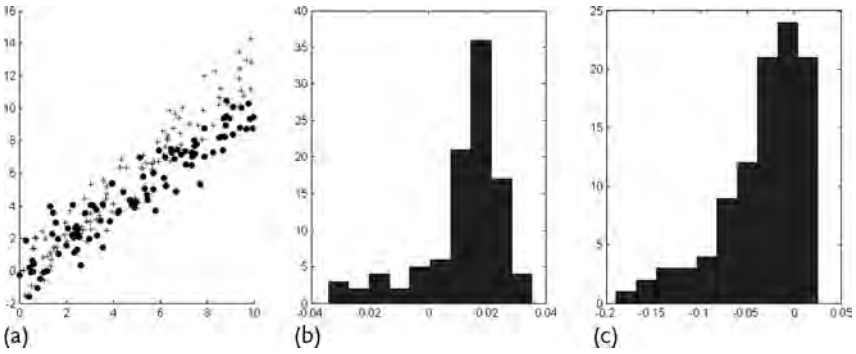

Figure 4.8 Leave-one-out cross-validation example. (a) Data samples, each of size 100, generated according to a linear model $Y = X$ (filled circles) or a quadratic model $Y = X + 0.03X^2$ (plus symbols). (b) Distribution of linear model minus quadratic model EPE for data generated according to a linear model. (c) Distribution of linear model minus quadratic model EPE for data generated according to a quadratic model.

validation estimates of the prediction error for both the linear and quadratic fit models, and again take the difference. The distribution of this difference, for 100 such experiments, is shown in Figure 4.8c. Choosing the quadratic model when the difference is less than zero gets it right in 79 out of 100 cases.

Leave-one-out cross-validation is particularly suitable if the size of our dataset is not large. However, as N becomes large, the required computer resources may be prohibitive. We may back off from the extreme represented by leave-one-out cross-validation and obtain *$\mathcal{K}$-fold cross-validation*. In this case, we divide the dataset into $\mathcal{K}$ disjointed subsets, of essentially equal size $m \approx N/\mathcal{K}$. Leave-one-out cross-validation corresponds to $\mathcal{K} = N$.

In $\mathcal{K}$-fold cross-validation, the model is trained on the dataset consisting of everything except the "held-out" sample of size m. Then the prediction error is obtained by applying the model to the held-out validation sample. This procedure is applied in turn for each of the $\mathcal{K}$ held-out samples, and the results for the squared prediction errors averaged.

We have a choice for $\mathcal{K}$; how can we decide? To get some insight, we note that there are three relevant issues: computer time, bias, and variance in our estimated prediction error. The choice of $\mathcal{K}$ may require compromises among these. In particular, for large datasets, we may be driven to small $\mathcal{K}$ by limited computing resources.

To understand the bias consideration, we introduce the *Learning Curve*, illustrated in Figure 4.9. This curve shows how the expected prediction error decreases as the training set size is increased.[1)] We may use it to understand how different choices of folding can lead to different biases in estimating the EPE. Except for very small dataset sizes, if we use N-fold cross-validation, we use essentially the whole dataset

1) It is conventional in the classification problem to show one minus the error rate as the learning curve.

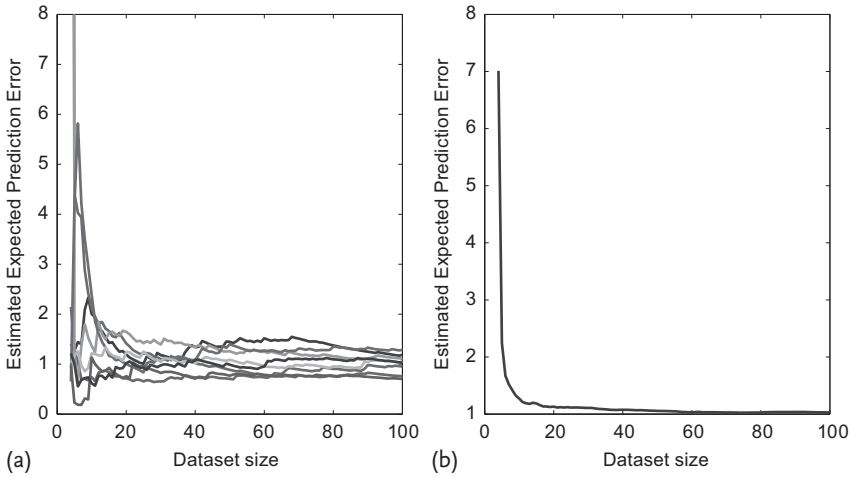

Figure 4.9 Computing the learning curve for cross-validation. (a) The dependence of estimated predicted error for leave-one-out cross-validation on sample size for several data samples. The data is generated according to the linear model as in Figure 4.8. (b) The learning curve, estimated by averaging together the results from 100 data samples. That is, 100 curves of the form illustrated in the left plot are averaged together.

for each evaluation of the EPE, and our estimator is close to unbiased (though as we see in Figure 4.9a, it may have a large variance). However, consider what happens if we go to smaller $\mathcal{K}$ values. To make the point, consider a dataset of size 20, and $\mathcal{K} = 2$. In this case, our estimated EPE will be based on a training sample of size 10. Looking at the learning curve, we see that yields an overestimate of the EPE for $N = 20$.

On the other hand, the larger $\mathcal{K}$ is, the more computation is required, so the lowest acceptable $\mathcal{K}$ is preferred. A much more subtle issue is the variance of the estimator (Breiman and Spector, 1992). As a rule of thumb, it is proposed that $\mathcal{K}$ values between 5 and 10 are typically approximately optimal. Performance on our regression example above degrades somewhat from leave-one-out to $\mathcal{K} = 5$ or 10.

We will see in the next chapter another application of cross-validation, to the problem of optimizing smoothing in density estimation.

4.6 ✂ Resampling Weighted Observations

It is not uncommon to deal with datasets in which the samplings are accompanied by a nonnegative "weight", w. For example, we may have samples corresponding to different luminosities that we wish to incorporate into an analysis. Hence, we describe our dataset as a set of pairs, $(x_n, w_n; n = 1, \ldots, N)$, where w_n is the weight associated with the sampling x_n (x_n could also be a vector). The question arises: How can we apply resampling methods to such a dataset?