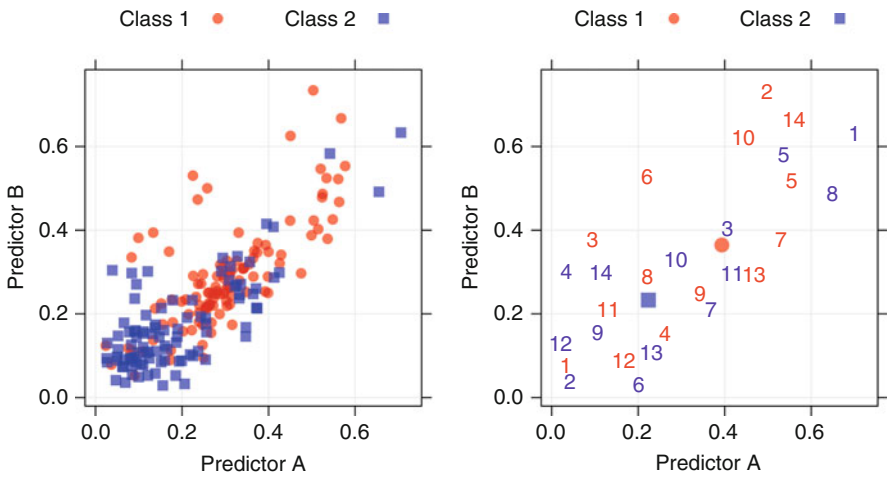

Fig. 4.5: An example of maximum dissimilarity sampling to create a test set. After choosing an initial sample within a class, 14 more samples were added

for the first class, the chosen point was far in the Northeast of the predictor space. As the sampling proceeds, samples were selected on the periphery of the data then work inward.

Martin et al. (2012) compares different methods of splitting data, including random sampling, dissimilarity sampling, and other methods.

4.4 Resampling Techniques

Generally, resampling techniques for estimating model performance operate similarly: a subset of samples are used to fit a model and the remaining samples are used to estimate the efficacy of the model. This process is repeated multiple times and the results are aggregated and summarized. The differences in techniques usually center around the method in which subsamples are chosen. We will consider the main flavors of resampling in the next few subsections.

k-Fold Cross-Validation

The samples are randomly partitioned into k sets of roughly equal size. A model is fit using the all samples except the first subset (called the first *fold*). The held-out samples are predicted by this model and used to estimate performance measures. The first subset is returned to the training set and

procedure repeats with the second subset held out, and so on. The k resampled estimates of performance are summarized (usually with the mean and standard error) and used to understand the relationship between the tuning parameter(s) and model utility. The cross-validation process with $k = 3$ is depicted in Fig. 4.6.

A slight variant of this method is to select the k partitions in a way that makes the folds balanced with respect to the outcome (Kohavi 1995). Stratified random sampling, previously discussed in Sect. 4.3, creates balance with respect to the outcome.

Another version, leave-one-out cross-validation (LOOCV), is the special case where k is the number of samples. In this case, since only one sample is held-out at a time, the final performance is calculated from the k individual held-out predictions. Additionally, repeated k-fold cross-validation replicates the procedure in Fig. 4.6 multiple times. For example, if 10-fold cross-validation was repeated five times, 50 different held-out sets would be used to estimate model efficacy.

The choice of k is usually 5 or 10, but there is no formal rule. As k gets larger, the difference in size between the training set and the resampling subsets gets smaller. As this difference decreases, the *bias* of the technique becomes smaller (i.e., the bias is smaller for $k = 10$ than $k = 5$). In this context, the bias is the difference between the estimated and true values of performance.

Another important aspect of a resampling technique is the uncertainty (i.e., variance or noise). An unbiased method may be estimating the correct value (e.g., the true theoretical performance) but may pay a high price in uncertainty. This means that repeating the resampling procedure may produce a very different value (but done enough times, it will estimate the true value). k-fold cross-validation generally has high variance compared to other methods and, for this reason, might not be attractive. It should be said that for large training sets, the potential issues with variance and bias become negligible.

From a practical viewpoint, larger values of k are more computationally burdensome. In the extreme, LOOCV is most computationally taxing because it requires as many model fits as data points and each model fit uses a subset that is nearly the same size of the training set. Molinaro (2005) found that leave-one-out and k =10-fold cross-validation yielded similar results, indicating that $k = 10$ is more attractive from the perspective of computational efficiency. Also, small values of k, say 2 or 3, have high bias but are very computationally efficient. However, the bias that comes with small values of k is about the same as the bias produced by the bootstrap (see below), but with much larger variance.

Research (Molinaro 2005; Kim 2009) indicates that repeating k-fold cross-validation can be used to effectively increase the precision of the estimates while still maintaining a small bias.

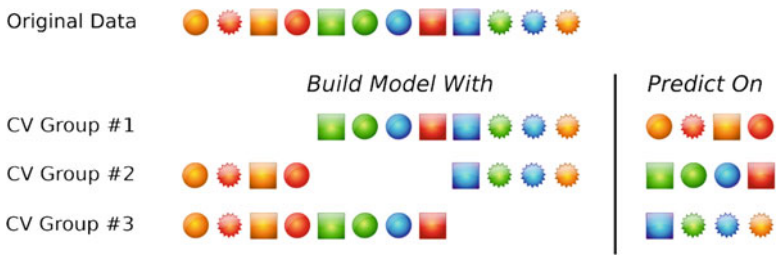

Fig. 4.6: A schematic of threefold cross-validation. Twelve training set samples are represented as symbols and are allocated to three groups. These groups are left out in turn as models are fit. Performance estimates, such as the error rate or R^2 are calculated from each set of held-out samples. The average of the three performance estimates would be the cross-validation estimate of model performance. In practice, the number of samples in the held-out subsets can vary but are roughly equal size

Generalized Cross-Validation

For linear regression models, there is a formula for approximating the leave-one-out error rate. The generalized cross-validation (GCV) statistic (Golub et al. 1979) does not require iterative refitting of the model to different data subsets. The formula for this statistic is the i_{th} training set outcome

$$\text{GCV} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - df/n}\right)^2,$$

where y_i is the i^{th} in the training set set outcome, $\hat{y}_i$ is the model prediction of that outcome, and df is the degrees of freedom of the model. The degrees of freedom are an accounting of how many parameters are estimated by the model and, by extension, a measure of complexity for linear regression models. Based on this equation, two models with the same sums of square errors (the numerator) would have different GCV values if the complexities of the models were different.

Repeated Training/Test Splits

Repeated training/test splits is also known as "leave-group-out cross-validation" or "Monte Carlo cross-validation." This technique simply creates multiple splits of the data into modeling and prediction sets (see Fig. 4.7). The proportion of the data going into each subset is controlled by the practitioner as is the number of repetitions. As previously discussed, the bias

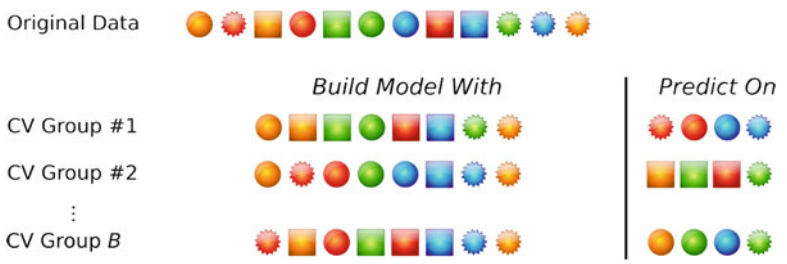

Fig. 4.7: A schematic of B repeated training and test set partitions. Twelve training set samples are represented as symbols and are allocated to B subsets that are 2/3 of the original training set. One difference between this procedure and k-fold cross-validation are that samples can be represented in multiple held-out subsets. Also, the number of repetitions is usually larger than in k-fold cross-validation

of the resampling technique decreases as the amount of data in the subset approaches the amount in the modeling set. A good rule of thumb is about 75–80 %. Higher proportions are a good idea if the number of repetitions is large.

The number of repetitions is important. Increasing the number of subsets has the effect of decreasing the uncertainty of the performance estimates. For example, to get a gross estimate of model performance, 25 repetitions will be adequate if the user is willing to accept some instability in the resulting values. However, to get stable estimates of performance, it is suggested to choose a larger number of repetitions (say 50–200). This is also a function of the proportion of samples being randomly allocated to the prediction set; the larger the percentage, the more repetitions are needed to reduce the uncertainty in the performance estimates.

The Bootstrap

A bootstrap sample is a random sample of the data taken *with replacement* (Efron and Tibshirani 1986). This means that, after a data point is selected for the subset, it is still available for further selection. The bootstrap sample is the same size as the original data set. As a result, some samples will be represented multiple times in the bootstrap sample while others will not be selected at all. The samples not selected are usually referred to as the "out-of-bag" samples. For a given iteration of bootstrap resampling, a model is built on the selected samples and is used to predict the out-of-bag samples (Fig. 4.8).

In general, bootstrap error rates tend to have less uncertainty than k-fold cross-validation (Efron 1983). However, on average, 63.2 % of the data points the bootstrap sample are represented at least once, so this technique has bias

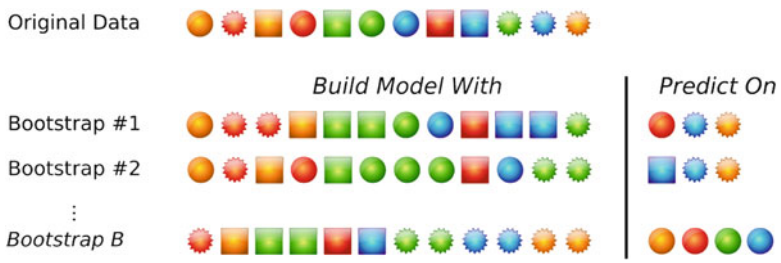

Fig. 4.8: A schematic of bootstrap resampling. Twelve training set samples are represented as symbols and are allocated to B subsets. Each subset is the same size as the original and can contain multiple instances of the same data point. Samples not selected by the bootstrap are predicted and used to estimate model performance

similar to k-fold cross-validation when $k \approx 2$. If the training set size is small, this bias may be problematic, but will decrease as the training set sample size becomes larger.

A few modifications of the simple bootstrap procedure have been devised to eliminate this bias. The "632 method" (Efron 1983) addresses this issue by creating a performance estimate that is a combination of the simple bootstrap estimate and the estimate from re-predicting the training set (e.g., the apparent error rate). For example, if a classification model was characterized by its error rate, the 632 method would use

$$(0.632 \times \text{simple bootstrap estimate}) + (0.368 \times \text{apparent error rate}).$$

The modified bootstrap estimate reduces the bias, but can be unstable with small samples sizes. This estimate can also result in unduly optimistic results when the model severely over-fits the data, since the apparent error rate will be close to zero. Efron and Tibshirani (1997) discuss another technique, called the "632+ method," for adjusting the bootstrap estimates.

4.5 Case Study: Credit Scoring

A straightforward application of predictive models is credit scoring. Existing data can be used to create a model to predict the probability that applicants have good credit. This information can be used to quantify the risk to the lender.

The German credit data set is a popular tool for benchmarking machine learning algorithms. It contains 1,000 samples that have been given labels of good and bad credit. In the data set, 70 % were rated as having good