Cross Validation

Quantifying the Quality of a Fit

How do you measure the performance of your fitted model? That is, what mathematical standard should you use to measure how well the model's predictions compare to the data? There are many such measures, but the one most commonly-used is the *mean-squared error* (MSE), defined by:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2,$$

where $\hat{y}_i$ is the predicated value of the ith observation. One way to remember the MSE formula is to think about it backwards: Find the "error" (E) or distance between the observed value, y_i and its predicted value, $\hat{y}_i$; "square" (S) this error; then take its "mean" (M) over all n observations.

Training and Test Data

When you fit a model to data, how confident are you that the model will generalize? For example, if you gather new data and fit the same model to the new data, will your results vary significantly? If the predictive power of your worsens as you fit the model to a different set of data, then you might want to consider another model. Cross validation is a powerful technique to investigate, among other things, the degree to which fitting a model to different data differs and is a necessary component of any workflow where the objective is predictive performance. When you use cross-validation techniques you're in the stage of model assessment, or perhaps model comparison.

New data are usually not available when you're at the stage of model fitting. Unless the dataset you're working with is very small, however, you can divide the data into two non-overlapping groups: A training set and a test set. A model is fit with the *training set*. Observations in the training set are used to *train*, or teach, the estimation method

An obvious question you might be thinking of is: How should you partition the data into training and testing sets? Shouldn't there be a some sort of decision rule to guide how the data are divided? There are a few ways to determine the split:

1. *The Validation Set Approach*:
    - Step 1: Randomly divide the data into two (usually equal) parts to form a training and test set.
    - Step 2: Fit the model with the training set to obtain results.
2. *k-Fold Cross-Validation*:
    - Step 1: Choose some integer k with $1 < k < n$, and partition the data into k approximately equal-sized groups called "folds."
    - Step 2: Hold out one fold and use the remaining $k - 1$ folds as training data to fit the model.
    - Step 3: Repeat Step 2 $k - 1$ more times, holding out a different fold each time you fit a new training set.
3. *Leave-One-Out Cross-Validation* (LOOCV):

- Step 1: Select the first observation, (x_1, y_1), to form the test set $\{(x_1, y_1)\}$, the remaining observations forming the training set $\{(x_2, y_2), \ldots, (x_n, y_n)\}$.
- Step 2: Fit the model with the training set.
- Step 3: Repeat Steps 1 and 2, each time holding out a single observation (x_i, y_i) for $i = 2, \ldots, n$. Note that implementing LOOCV means you'll have n training sets, one for each observation left out, and thus fit the model n times.

We'll use LOOCV in the following examples as it's currently the most widely adopted approach to cross validation. See the following schematic display of the LOOCV approach that illustrates how the data are partitioned.

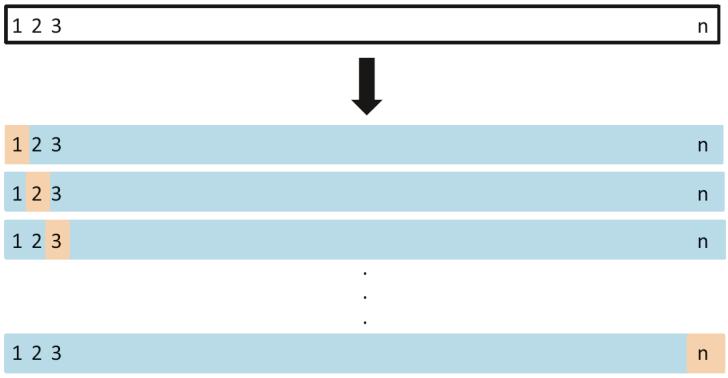

FIGURE 5.3. *A schematic display of LOOCV. A set of n data points is repeatedly split into a training set (shown in blue) containing all but one observation, and a validation set that contains only that observation (shown in beige). The test error is then estimated by averaging the n resulting MSE's. The first training set contains all but observation 1, the second training set contains all but observation 2, and so forth.*

Now, recalling that the MSE is our choice to measure the quality of a fit, the n estimations generate n squared errors for the test set, $MSE_1, MSE_2, \ldots, MSE_n$. Specifically, $MSE_1 = (y_1 - \hat{y}_1)^2$, $MSE_2 = (y_2 - \hat{y}_2)^2$, ..., $MSE_n = (y_n - \hat{y}_n)^2$. Looking back at the definition of the MSE given above, note that there's not need to sum observations because there's only 1 observation in each of the n test sets.

We can derive the LOOCV estimate for the test MSE by averaging these n test error estimates:

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

Do CV for purposes of model assessment;

Cross-validation techniques can be implemented with the **modelr** package. The **modelr** package isn't part of **tidyverse**, so you'll need to install and load it into R in the usual way.