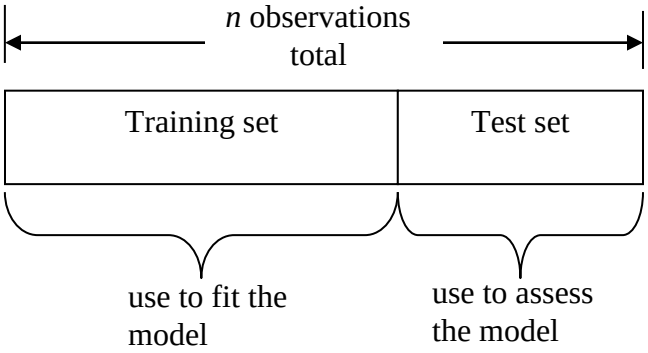
Cross-Validation

Cross-validation (CV) is a very general tool for:

- Assessing how well a fitted model will predict new data, and/or
- Selecting the best model, based on how well we think the model will predict a new set of "test" data, when we do not have the luxury of setting aside a real set of test data.

CV serves the same purpose as Mallows' C_p statistic (or Akaike's generalization the *AIC* statistic) but is completely general and can be used to assess the quality of virtually any type of fitted model in any scenario. In contrast, theoretically derived criteria like C_p and *AIC* only apply to the specific types of models on which their derivations were based, such as linear least squares regression (for C_p) or maximum likelihood estimation (for *AIC*). In addition, *AIC* and (to a lesser extent) C_p are based on asymptotic approximations that render them strictly valid only for large sample sizes. Whereas there are many assumptions and approximations in the derivation of theoretical criteria like C_p and *AIC*, CV is an entirely empirical measure that involves no assumptions and is, therefore, almost universally applicable.

Suppose we have a data set consisting of n observations of a single response variable y and k predictor variables $\{x_1, x_2, \ldots, x_k\}$ and we want to assess the quality of some model of a particular structure (e.g., a linear regression model containing a subset of the predictors) that we intend to fit to the data. Here, "quality" means that we would like to assess how well the fitted model would do at predicting a new set of data **that was not used to fit the model**. It is pointless to use the prediction over the training set as a measure of how well the model would predict over a new test set of data. Indeed, a model having more predictors will always fit the training data better than a model with fewer predictors. If we have enough data to set aside a "test" set, while still leaving enough data in the "training" set to fit the model, then we can fit the models to the training data and choose the best model as the one that does the best at predicting the test data. For example, we might set aside 1/3 of the data for test purposes and use the remaining 2/3 for training. The roughly $n/3$ observations that are put in the test set must be chosen randomly. Conceptually, it looks like the following figure:



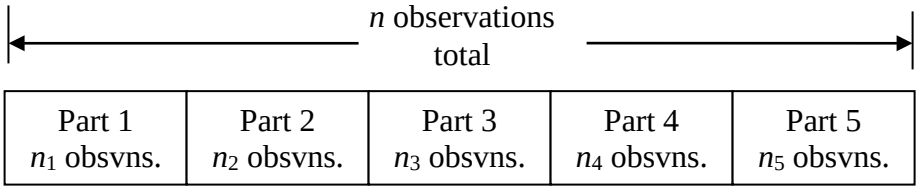
If we do this, however, it means that some of our data are not available for fitting purposes, which is generally undesirable. An alternative that does not involve this tradeoff is CV.

We will illustrate the basic idea behind *K*-fold CV for the (linear or nonlinear) regression situation in which the squared prediction error on "test" observations is our preferred measure of how good the model is.

First, *randomly* split the *n* observations into *K* parts of roughly equal size (as equal as we can get them). Let n_j denote the exact number of observations in Part *j* ($n_j = n/K$ roughly), so that

$$n = \sum_{j=1}^{K} n_j$$

The following figure is an illustration for $K = 5$:



Then, for $j = 1, 2, \ldots, K$, we:

1) Set aside Part *j* as a temporary test set.

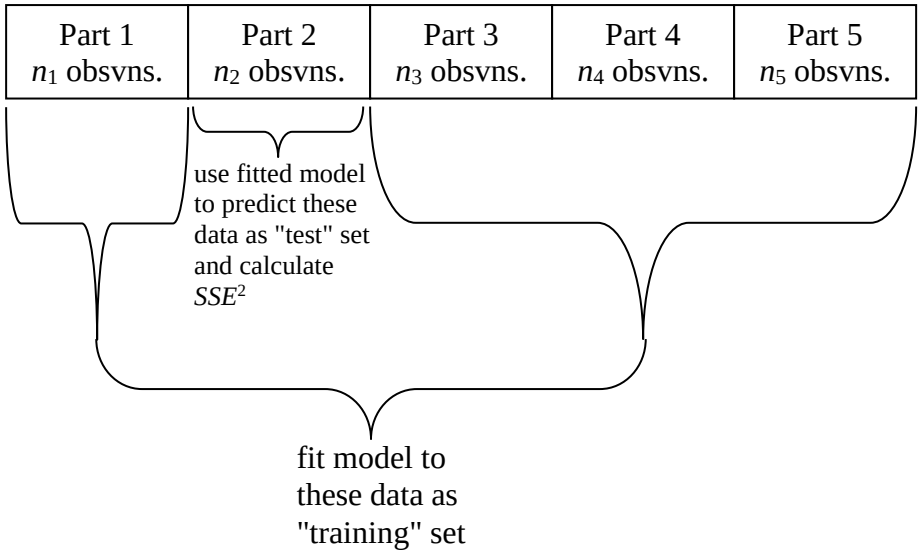
2) For the particular model that you are trying to assess (e.g., a neural network model with a specified set of predictors and number of hidden layers and nodes; a linear regression model with a specified set of predictors; a ridge regression model with a specified set of predictors and a specified shrinkage parameter; a tree with a specified set of predictors and a specified complexity parameter, etc.) fit this model to the

remaining K–1 parts, taken as a single training set of roughly $[(K-1)/K]n$ observations.

3) Use the fitted model from Step 2 to predict the n_j response values in Part j (the temporary test set) and calculate the resulting test error sum of squares for Part j:

$$SSE^j = \sum_{\substack{\text{all } i \text{ in} \\ \text{Part } j}} \left(y_i - \hat{y}_i^{\,j}\right)^2,$$

where $\hat{y}_i^{\,j}$ denotes the prediction of y_i (from Part j) using the model fitted in Step 2 (which excluded Part j). The following figure illustrates for $j = 2$



After repeating Steps 1—3 for $j = 1, 2, \ldots, K$ (i.e., for each Part set aside as a test part), we finally calculate the cross-validation SSE:

$$SSE_{CV} = \sum_{j=1}^{K} SSE^j$$

We can use SSE_{CV} as a direct measure of the quality of our model, in terms of how well it would do predicting a new set of data, independent from the training set.

Comments:

1) SSE_{CV}/n serves as a measure of the prediction error variance that we can expect when using the model in question to predict a new test case that is independent of the training data (but whose predictor values are consistent with those in the training data

set). Similarly, you can take $r^2_{CV} = 1 - \frac{SSE_{CV}}{SST}$, where *SST* is the total sum of squares for the response (n–1 times the sample variance of the response).

2) Note that we have actually fit K different models (one for each $j = 1, 2, \ldots, K$), each having different fitted parameters (but otherwise the same). The final, single model that we would use for any future purposes should be fit to the *entire* set of n observations. It does not matter that this model will have slightly different parameters than the five models fit in the above procedure; we can still use SSE_{CV}/n as a measure of how well we expect the final single model to perform when predicting a new test case.

3) CV is useful for model comparison and selection. Suppose we want to compare a number of different fitted models (e.g., linear regression models with different sets of predictors, ridge regression models with different shrinkage parameter, a neural network models, other nonlinear regression models, etc.) and select the "best" one. For <u>*each*</u> model we can calculate an SSE_{CV} as described above. Then, to select the "best" model, we simply take the one that has the smallest SSE_{CV}. This is analogous to selecting the model with the smallest C_p or *AIC*, except CV applies to virtually all types of models. You should **use the same CV partition for all models**.

4) SSE_{CV} measures directly and empirically what C_p and *AIC* are intended to measure analytically. Namely, it measures how well the fitted model can predict a new set of test data. Because CV involves fewer approximations and assumptions than the analytical methods, it is much more broadly applicable and generally more accurate. The only drawback is that CV is more computationally expensive.

5) Common choices for K are $3 \le K \le 10$. The main drawback of larger K within this range is higher computational expense. A good rule-of-thumb is to always use $K = 10$, unless you need to use a lower K for computational reasons. Sometimes n-fold CV is used for certain simple-structured models like linear regression, for which there is a very clever computational trick that applies only for n-fold CV. Note that SSE_{CV} for n-fold CV in linear regression is precisely the $PRESS_p$ statistic that is used for comparing linear regression models. The $PRESS_p$ computational trick allows the n-fold SSE_{CV} to be calculated without having to fit n separate models ($j = 1, 2, \ldots, n$).

6) The CV measure of how good the model is does not necessarily have to be the SSE. You can choose whatever measure you think best reflects the model performance. For example, in Step 3 of the CV procedure, you could instead calculate the sum of the absolute error (SAE):

$$SAE^j = \sum_{\substack{\text{all } i \text{ in} \\ \text{Part } j}} \left| y_i - \hat{y}_i^j \right|,$$

Or, for a classification problem, you could use the number of misclassified observations in the *j*th Part. It is VERY general. In contrast, AIC only applies when the performance measure is the expected log-likelihood.

7) Some important issues to keep in mind when using CV:

- Always randomly assign the *n* observations to the *K* parts, but under the constraint that the sizes of the parts are as equal as possible.
- When comparing multiple models, always uses the same partition for each of the models.
- Unless you are using *n*-fold CV, you will get a different result each time you run CV (with a different random partition). Hence, it is important to conduct a number of replicates of CV, each time using a different random partition. You can then use the average SSE_{CV} values across all of the replicates to compare multiple models.