Advanced Topics in Statistics

∼ Resampling Methods ∼

4

INTRODUCTION.

Resampling methods involve repeatedly drawing samples from a training set and refitting a model of interest on each sample in order to obtain more information about the fitted model.

They are computationally expensive, because they involve fitting the same statistical method multiple times using different subsets of the training data. (But we have powerful computers).

Two of the most commonly used resampling methods are cross validation and bootstrapping.

Cross-validation has the following main applications.

- Model Assessment: estimate test error rate for a given statistical learning method in order to evaluate its performance.
- Model Selection: select the appropriate level of model flexibility.

Bootstrapping can be used to provide a measure of accuracy of a parameter estimate or of a given statistical learning method.

Here we will focus on cross-validation.

ESTIMATING THE TEST ERROR RATE.

The use of a particular learning method is warranted if it results in a low test error.

Recall that the test error rate is the average error that results from using the statistical learning method to predict the response on a new observation (that was not used in the training method).

The test error rate is easily calculated if a designated test set is available. Unfortunately this is usually not the case.

One approach for estimating the test error rate in the absence of a test data set is holding out a subset of the training observations from the fitting process, and then applying the fitted statistical learning method to those held out observations.

There are a number of strategies for leaving a subset of the observations out. We will consider the following:

- The validation set approach.
- Leave one out cross validation
- *k*-fold cross validation.

These methods can be used in both the regression and classification setting.

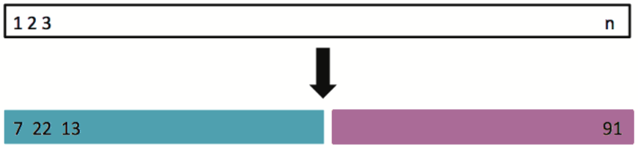
THE VALIDATION SET APPROACH.

Suppose we would like to estimate the test error associated when fitting a particular statistical learning method on a set of observations.

If we have a large data set, we can achieve this goal by randomly splitting the data into two parts, a training and a validation (hold-out) set.

The model is fit on the training set and the fitted model is used to predict the responses for the observations in the validation set. The validation set error rate provides an estimate of the test error rate.

The method can also be used to compare models. We use the training set to build each model, and choose the model that gives the lowest error rate when applied to the validation data.



Validation set approach - Auto example.

The Auto dataset contains information of 398 different automobile models. There are 9 different variables in the dataset, but here we will focus on the fuel consumption (in miles per gallon), and horsepower. In particular, suppose that we want to predict mpg from horsepower.

Some exploratory data analysis reveals a non-linear relationship between `mpg` and `horsepower`. Thus, we consider the following models

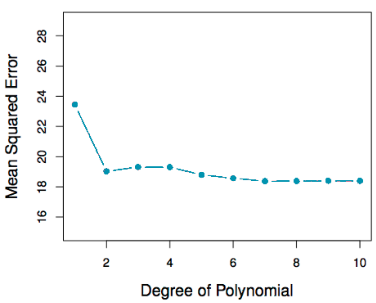
- `mpg` $\sim$ `horsepower`,
- `mpg` $\sim$ `horsepower + horsepower`^{2},
- ...

and use the validation set approach to decide on the order of polynomial that gives the best fit. That is

- We first randomly split the Auto data set into training (196 obs.) and validation data (196 obs.).
- Then we fit both models using the training data set, and evaluate both models using the validation data set.
- The model with the lowest validation (testing) MSE is the winner. (Note that since this is a regression problem we use the mean squared error to assess the accuracy).

AUTO DATA RESULTS.

We can plot the validation set MSE as a function of the degree of polynomial to see how the model fit changes as higher and higher order terms are added to the model.



The plot shows that the validation set MSE for the quadratic fit is considerably lower than for the linear fit.

However the validation set MSE for the cubic fit is slightly larger than for the quadratic fit, thus including the cubic term does not lead to better prediction than simply using the quadratic term.

Furthermore we can't see much improvement later on either.

Overall we can conclude that a linear model is not adequate for this data, using a quadratic function of the `horsepower` variable however provides a dramatic improvement in model accuracy.

We usually try to find the balance between model accuracy and model complexity. Thus we should always weigh the benefit of including extra terms against the introduced complexity.

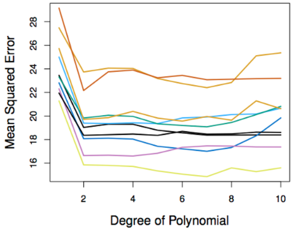
Drawbacks of the validation set approach.

Recall that the validation set approach randomly divides the data into two parts. Each time we repeat the process of random splitting we will get a somewhat different estimate for the test MSE.

Upon repeating the validation set method is10 times, we can observe the following:

Each curve shows that including the second order terms will lead to a dramatically smaller valideation set MSE than the linear one.

However each of the ten curves results in a different test MSE estimate for each of the ten regression models considered. And there is no consensus among the curves as to which model results in the smallest validation set MSE.



Even though the validation set approach is simple and easy to implement, it has two potential drawbacks:

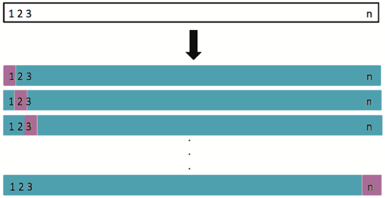
- The validation MSE can be highly variable.
- Only a subset of observations are used to fit the model (training data). We are decreasing our sample size.

LEAVE-ONE-OUT CROSS VALIDATION.

Leave-one-out cross-validation (LOOCV) is similar to the Validation Set Approach, but it tries to address the latter's disadvantages.

For each suggested model we do the following steps.

1. Split the dataset of size n into a training data set of size $n-1$ and a validation data set of size 1.
2. Fit the model using the training data.
3. Validate model using the validation data, and compute the corresponding error rate.
4. Repeat this process n times.
5. The test error rate is approximated by the average of the n test error estimates.



LOOCV vs Validation set approach.

The LOOCV and the validation set approach both split the data into two parts. But instead of creating two subsets of comparable size, the LOOCV approach uses a single observation from the training data for the validation set. And it repeats this process for each observation of the training data. As a result:

- LOOCV has less bias.

  We repeatedly fit the statistical learning method using training data that contains $n - 1$ observation, i.e. almost all the data set is used.

- LOOCV produces a less variable error rate.

  The validation approach produces different error rates when applied repeatedly due to randomness in the splitting process, while performing LOOCV multiple times will always yield the same results, because we split based on 1 observation each time (there is no randomness).

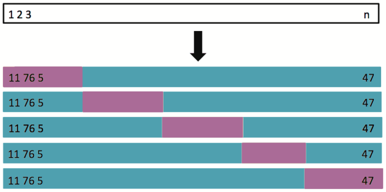
- LOOCV has the potential to be computationally intensive.

  This is the drawback compared to the validation set approach. LOOCV fits each model n times. Although in certain cases shortcuts exist that reduces the cost of LOOCV.

k-FOLD CROSS VALIDATION.

An alternative to LOOCV is *k*-fold cross validation which aims to address the drawback of the LOOCV.

1. *k*-fold CV first randomly divides the set of observations into k groups, or folds, of approximately equal size.
2. The first fold is treated as a validation set, and the method is fitted on the remaining $k-1$ folds.



3. The error rate is then computed on the observations in the held-out fold.
4. This procedure is repeated k times. Each time a different group of observations is treated as a validation set.
5. The process results in k estimates of the test error. The *k*-fold CV estimate is computed by averaging these values.

It is not hard to see that LOOCV is a special case of the *k*-fold CV in which k is set to equal n.

Validation methods - Regression example.

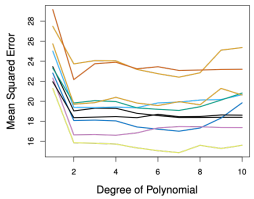
To compare the three validation methods, we use the previously introduced Auto dataset.

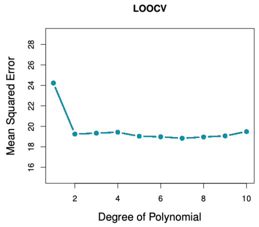
First, the validation set approach is used ten times, then LOOCV, and finally a 10-fold cross validation is repeated nine times.

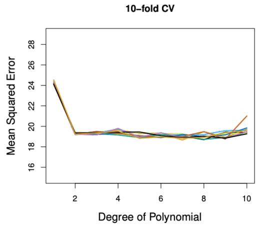
We can see that when using 10-fold CV we do get some variability in the test error estimates as a result of randomly dividing the observations into ten folds.

But this variability is much lower that the variability in the test error estimates that results from the validation set approach (on the left).

Overall we can see that both LOOCV and 10-fold CV are stable methods.







Bias-Variance Trade-Off for k-fold CV.

When we perform cross-validation, usually our goal is to determine how well a given statistical learning procedure can be expected to perform on independent data.

Thus our interest lies in estimating the test error rate.

We have seen that both LOOCV and k-fold CV are stable methods, but LOOCV is computationally expensive.

But putting aside the computational cost, which is better, LOOCV or k-fold CV?

- LOOCV has less bias than k-fold CV (when $k < n$).
- But, LOOCV has higher variance than k-fold CV (when $k < n$).

Thus, there is a trade-off associated to the choice of k in k-fold CV.

Given these considerations we tend to use k-fold CV with $k = 5$ and $k = 10$. It has been empirically shown that they both yield test error rate estimates that suffer neither from excessively high bias, nor from very high variance.

A consequence of the bias-variance trade-off is that the k-fold CV often results in more accurate estimates of the test error rate than LOOCV.

10-fold cross validation - Classification example.

As we mentioned previously validation methods work similarly for regression and classification problems. But while in regression problems we are trying to estimate the test MSE, in classification problems we use the number of misclassified observations.

In the classification setting the CV-error for a k-fold CV is

$$CV_{(k)} = \frac{1}{k} \sum_{j=1}^{k} \mathrm{Err}_j,$$

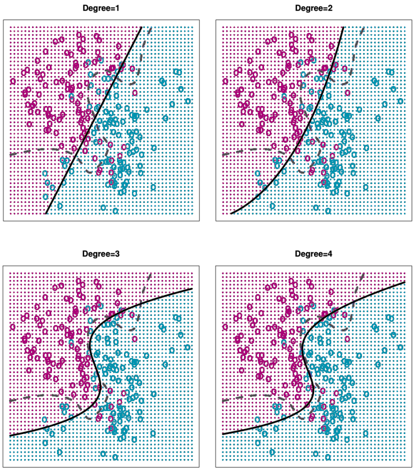
where Err_j is the estimated test error when the jth fold is used as a validation data set.

As an example we will take some simulated data (so that the Bayes decision boundary and test error rates are known), and use 10-fold CV to estimate the test error rates for some polynomial logistic regression fits.

These include using linear, quadratic, cubic and quartic logistic regressions on simulated data. E.g. the quadratic logistic regression model with two predictors is given by

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_2 + \beta_4 X_2^2.$$

CLASSIFICATION EXAMPLE OUTCOME.



The figures show the estimated decision boundaries (solid lines) for the linear, quadratic, cubic and quartic logistic regression.

We can see that the linear logistic regression is not able to fit the Bayes' decision boundary (given by the dashed line. It gives a test error rate of 0.201, while the error rate of the Bayes' classifier is 0.133.

Quadratic Logistic regression does better than linear with test error rate 0.197.

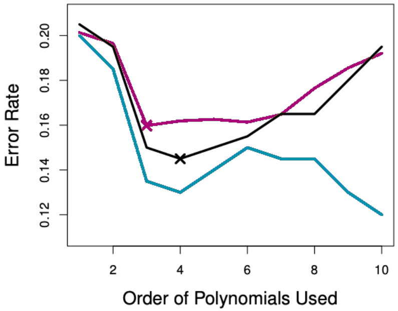
While by using cubic and quartic predictors, the accuracy of the model improves even further (test error rates are 0.160 and 0.162 respectively).

Here, since the data is simulated, we could just compute the test error rates. But how should we decide on the degree of the polynomial when we don't have access to the true test error rate? This is where *k*-fold CV comes into the picture.

Classification example 10-fold CV outcome.

We use 10-fold CV to estimate the test error rate for ten different logistic regression models, where we use polynomial functions of the predictors up to the tenth order.

The true test error is shown in **pink**, and the training error is shown in **blue**, while the 10-fold CV error is in **black**.



As we have seen previously the training error tends to decrease as the flexibility of the model increases

The figure indicate that though the training error rate doesn't quite decrease monotonically, it tends to decrease on the whole as the model complexity increases.

In contrast, the test error rate displays a characteristic U-shape.

The 10-fold CV error rate provides a pretty good approximation to the test error rate.

While the CV error rate slightly underestimates the test error rate, it reaches its minimum when forth-order polynomials are used, which is very close to the minimum of the test curve, which occurs when third-order polynomials are used.