1 Resampling Methods

Sampling methods are a class of methods that serve a twofold purpose

- Provide a subset of data to be used for evaluating the test error rate
- Use different samples of data to assess the variability in the model parameters to choose the level of flexibility

1.1 Cross Validation

Validation set approach forms the bedrock for cross validation. We randomly split our training data into two sets, a training set and a validation set. We fit our models on the training set, and the validation set will serve as the unseen data. We can evaluate different models on the validation test to judge which of them performs the best.
A small problem with this approach is that the validation error will depend on the split of the data, i.e., we can expect slighlty different validation errors based on which subset of data we train. Hence, it is better to do this sampling multiple times in order to confidently select models and report test errors.
Furthermore, by preparing a validation set, we are reducing the size of our training set. Larger training data will usually result in better models. Hence, we might be overestimating the test errors in this case. A simple way to avoid this problem is to choose small sizes of the validation set and do the tests multiple times.

1.1.1 Leave One Out CV

An extension of the validation approach, here we train the data on all but on example. This way, if the data has n examples, we build n models (each trained on $n-1$ examples) and the error is

$$\text{test error } = \frac{1}{n}\sum_{i=1}^{n} error_i$$

where $error_i$ is the error on i^{th} observation from the model trained on remaining $n-1$ observations

Computing this can be extremely expensive when n is large. For linear regression, there exists a trick by which the time taken to get *test error* is exactly the same as the time to fit a single model on entire data set !

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$
$$\text{Hat matrix } H = X(X^T X)^{-1} X^T$$

Let X_i denote the matrix X but with the i^{th} row removed, and similarly Y_i. Let x_i^T denote the i^{th} row of X and h_i be the diagonal entry of H. Then we have the following

$$X_i^T X_i = X^T X - x_i x_i^T$$
$$X_i^T Y = X^T Y - x_i y_i$$
$$h_i = x_i^T (X^T X)^{-1} x$$
$$\hat{\beta}_i = (X_i^T X_i)^{-1} X_i^T Y_i$$
$$e_i = y_i - x_i^T \hat{\beta}_i$$

Also, we have the Sherman–Morrison formula for calculating the inverse of a perturbated matrix using the original matrix. Let A be the original invertible square matrix and u, v be column vectors. Then,

$$(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^TA^{-1}}{1 + v^TA^{-1}u}$$

The formula can be verified by evaluating $LHS * RHS = RHS * LHS = I$.

Substituiting $A = X^TX$ and $-u = v = x_i$,

$$\begin{aligned}
(X^TX - x_ix_i^T)^{-1} &= (X^TX)^{-1} + \frac{(X^TX)^{-1}x_ix_i^T(X^TX)^{-1}}{1 - x_i^T(X^TX)^{-1}x_i} \\
(X_i^TX_i)^{-1} &= (X^TX)^{-1} + \frac{(X^TX)^{-1}x_ix_i^T(X^TX)^{-1}}{1 - x_i^T(X^TX)^{-1}x_i} \\
\hat{\beta}_i(X_i^TY_i)^{-1} &= (X^TX)^{-1} + \frac{(X^TX)^{-1}x_ix_i^T(X^TX)^{-1}}{1 - x_i^T(X^TX)^{-1}x_i} \\
\hat{\beta}_i &= [(X^TX)^{-1} + \frac{(X^TX)^{-1}x_ix_i^T(X^TX)^{-1}}{1 - x_i^T(X^TX)^{-1}x_i}](X_i^TY_i) \\
\hat{\beta}_i &= [(X^TX)^{-1} + \frac{(X^TX)^{-1}x_ix_i^T(X^TX)^{-1}}{1 - h_i}](X^TY - x_iy_i) \\
\hat{\beta}_i &= \hat{\beta} - [\frac{(X^TX)^{-1}x_i}{1 - h_i}](y_i(1 - h_i) - x_i^T\beta_i + \hat{} h_iy_i) \\
&= \hat{\beta} - [\frac{(X^TX)^{-1}x_i(y_i - x_i^T\hat{\beta})}{1 - h_i}] \\
\text{Subsequently, } e_i &= y_i - x_i^T\hat{\beta}_i \\
&= y_i - x_i^T(\hat{\beta} - [\frac{(X^TX)^{-1}x_i(y_i - x_i^T\hat{\beta})}{1 - h_i}]) \\
&= (y_i - x_i^T\hat{\beta}) + \frac{h_i(y_i - x_i^T\hat{\beta})}{1 - h_i} \\
\text{or, } e_i &= \frac{y_i - x_i^T\hat{\beta}}{1 - h_i} \\
&= \frac{y_i - \hat{y}_i}{1 - h_i}
\end{aligned}$$

Applying this formula for all the errors across the n models,

$$\begin{aligned}
\text{test error } &= \frac{1}{n}\sum_{i=1}^{n} error_i \\
&= \frac{1}{n}\sum_{i=1}^{n}(\frac{y_i - \hat{y}_i}{1 - h_i})^2
\end{aligned}$$

which can be computed by simply building a single model on all the n data points.

1.1.2 k-Fold Cross Validation

Here, we divide the data set into k groups of roughly the equal size, and build k models. In each model, one of the folds is chosen as the validation set while the remaining $k - 1$ folds are

used for training. Let $MSE_1, MSE_2, \ldots, MSE_k$ denote the individual errors on the k subsets (when they are used for validation), then the estimate of test MSE is

$$MSE = \frac{1}{k}\sum_{i=1}^{k} MSE_i$$

Typically, $k = 5, 10$. Leave one out CV is a special case of k-fold CV. The MSE calculated above can help determine the best flexibility to chose for a given model (using the minima of MSE vs fliexibility) or help compare different types of models using the value of MSE.

1.1.3 k-Fold CV bias-variance

k-fold CV is preferable over LOOCV not just because of computational issues, but also due to a bias variance tradeoff. Note that the more data we use, it is expected that we will have less bias. By this logic, we expect LOOCV to have a low bias compared to k-fld CV. However, variance is also of interest in a statistical model. The n models that we fit in LOOCV are correlated with one another, since we have almost identical data sets being used for training. Hence, the average of the MSE's of such correlated models will tend to have a higher variance. On the other hand, k-fold CV effectively creates less correlated models which have quite lower variance in comparison. Hence k-fold CV is preferable in most of the cases.

$$Var(X + Y) = Var(X) + Var(Y) + 2Cov(X, Y)$$

Hence, higher the correlation, higher the variance of sum of two random variables.

1.2 Bootstrap

A widely applicable method that helps quantify uncertainty in the estimates of a statistical model. This is especially useful when formulae don't exist to quantify this uncertainty.
To estimate the uncertainty in estimates, we repeatedly create new data sets called bootstrap samples. Here, we simply select n observations from the original data set, but with replacement. This creates a slightly different version of the original data on which we train our model and get the estimates. From several bootsrapped data sets, we obtain a distribution of our estimate (a collection of estimates) whose mean and variance can help quantify the uncertainty.

1.2.1 Probability of selection

Since we are sampling with replacement, notice that the chance that the i^{th} observation is in the sampled data set is given by

$$P(not\ selection) = (1 - \frac{1}{n})^n$$
$$p(selection) = 1 - (1 - \frac{1}{n})^n$$

This comes using the product rule. The chance that any observation of the bootstrap sample is not equal to i^{th} observation is simply $1-\frac{1}{n}$ since we can choose any of the other $n-1$ examples.

$$
\begin{aligned}
\text{Note, } y &= \lim_{n\to\text{inf}}(1-\frac{1}{n})^n \\
\ln(y) &= \lim_{n\to\text{inf}}\frac{1}{n}\ln(1-\frac{1}{n}) \\
&= \lim_{n\to\text{inf}}\frac{\ln(1-\frac{1}{n})}{\frac{1}{n}} \\
&= \lim_{n\to\text{inf}}\frac{\frac{1}{1-\frac{1}{n}}\frac{1}{n^2}}{-\frac{1}{n^2}} && \text{Using L'Hospital's Rule} \\
&= \lim_{n\to\text{inf}}-\frac{1}{1-\frac{1}{n}} \\
&= -1 \\
y &= \frac{1}{e} \\
\lim_{n\to\text{inf}} P(\textit{selection}) &= 1-\frac{1}{e}\approx 0.632
\end{aligned}
$$

Hence, the probability that an observation from the original data set falls into a bootstrap sample is 63.2% which suggests that the sampled data sets are sufficiently variable !