Machine Learning for OR & FE

Resampling Methods

Martin Haugh

Department of Industrial Engineering and Operations Research
Columbia University
Email: `martin.b.haugh@gmail.com`

Some of the figures in this presentation are taken from "An Introduction to Statistical Learning, with applications in R" (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani (JWHT).

Additional References: Chapter 5 of ISLR.

Resampling Methods

- Resampling methods are a key tool in modern statistics and machine learning.
- Resampling methods involve:
  1. Repeatedly drawing a sample from the training data.
  2. Refitting the model of interest with each new sample.
  3. Examining all of the refitted models and then drawing appropriate conclusions.
- Resampling is computationally very expensive but with the advent of modern computing this is not a major drawback.
- There are two major resampling techniques:
  1. Cross-Validation: generally used to estimate the error (model assessment) associated with a given learning model and / or to select the appropriate learning model (model selection).
  2. The Bootstrap: most commonly used to provide a measure of accuracy of a parameter estimate or of a given learning method.

  Cross-validation and the bootstrap are easy to implement and very broadly applicable.

Problems in the Training, Validation and Test Set Approach

- Data is often scarce in which case we cannot afford to set aside separate validation (and / or test sets) when training a classifier or fitting a regression.
- There are also some drawbacks to using a validation set in the training procedure:
  1. Performance on the validation set is a (often highly variable) random variable depending on the data-split into training and validation sets.
  2. The error on the validation set tends to over-estimate the test-error rate of the model that is fitted to the entire data-set. Why?
- Instead we can perform K-fold cross-validation
  - closely related to the validation set approach but it does address the drawbacks highlighted above.

K-Fold Cross-Validation

K-Fold Cross-Validation proceeds as follows:

1. Partition the entire data-set into K equal-sized components.
2. For $k = 1, \ldots, K$, fit the model to the other $K - 1$ components and calculate the prediction error on the k^{th} component.
3. Combine the K estimates of the prediction error to obtain an average prediction error:

$$\mathsf{CV}_K\left(\hat{f}\right) = \frac{\sum_{i=1}^{N} L\left(y_i, \hat{f}^{-\kappa(i)}(\mathbf{x}_i)\right)}{N}$$

where:

- $\kappa(i) \in \{1, \ldots, K\}$ denotes the partition component to which the i^{th} observation was allocated
- $\hat{f}^{-\kappa(i)}(\cdot)$ denotes the fitted classifier / value in step (2) above when $k = \kappa(i)$.

K-Fold Cross-Validation

- Typical choices of K are 5 or 10.
- When $K = N$ the procedure is called leave-one-out cross-validation
  - very little bias then (why?) but also computationally very expensive.
- If we have a set of models indexed by α then we can use K-fold cross-validation to estimate $\mathrm{CV}_K\left(\hat{f}, \alpha\right)$ for each α
  - can then plot $\mathrm{CV}_K\left(\hat{f}, \alpha\right)$ against α to yield a test-error curve
  - choose $\alpha = \hat{\alpha}$ where $\hat{\alpha}$ minimizes this curve
  - then obtain $f(\mathbf{x}, \hat{\alpha})$ by fitting to the entire data-set with $\alpha = \hat{\alpha}$.
- Click on the image below for a brief but very nice tutorial by David Weiss on the use of cross-validation in the context of k-Nearest Neighbors.

The Wrong Way to Do Cross-Validation

This example is taken from Section 7.10.2 of HTF:

Consider a classification problem with a large number of predictors. One possible strategy for analysis is:

1. Screen the predictors to find a subset of "good" predictors that show strong univariate correlation with the class labels.
2. Use just this subset of predictors to build a multivariate classifier.
3. Use cross-validation to estimate the unknown tuning parameters (e.g. k in k-NN or a regularization parameter λ) and / or to estimate the prediction error of the final model.

Question: Is this a correct application of cross-validation?

The Wrong Way to Do Cross-Validation

Answer: No! Consider a scenario with $N = 50$ samples in two equal-size classes and $p = 5,000$ standard Gaussian predictors that are independent of the class labels.

Then the true test-error rate of any classifier will be (why?) 50%.

HTF carried out steps (1) to (3):

1. they chose the 100 predictors having the highest correlation with the class labels.
2. they built a 1-NN classifier using the 100 predictors from step (1).
3. they estimated the prediction error using CV.

They repeated this experiment 50 times and obtained an average CV error of 3%!

Question: What has happened?

Question: What is the correct way to do cross-validation here?

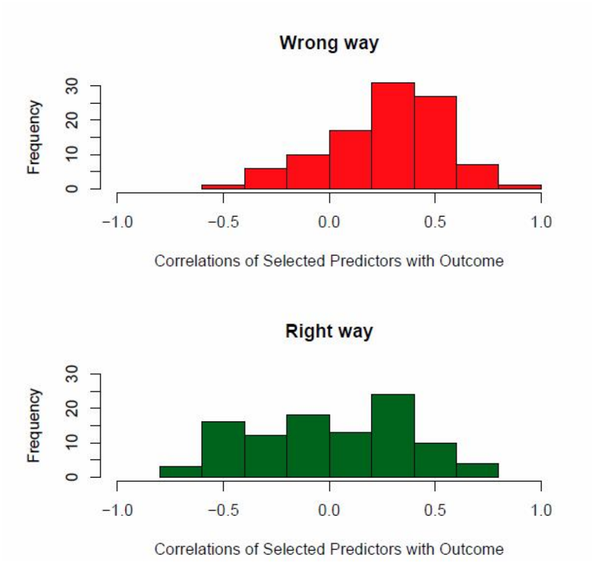

Figure 7.10 from HTF: Cross-validation the wrong and right way: histograms shows the correlation of class labels, in 10 randomly chosen samples, with the 100 predictors chosen using the incorrect (upper red) and correct (lower green) versions of cross-validation.

An Extreme Example with Leave-One-Out CV

- Consider a random data-set with $2N$ observations, 2 classes and M predictors.
- N of the observations have been assigned to the first class and the other N have been assigned to the second class.
- Suppose this class assignment has been made independently of the predictors.
- Suppose we use leave-one-out cross-validation to estimate the prediction error of a given model.
- Under leave-one-out CV the best model will predict the majority class.
- The true test error will then be 50%.
- But the predicted error will be 100%!

See Chapter 7 of HTF for further details on cross-validation and controlling over-fitting.

The Bootstrap

The bootstrap is a very powerful statistical tool that can be used to quantify:

1. The uncertainty associated with a given estimator
2. The uncertainty associated with a given statistical learning method.

The bootstrap is simple to use, broadly applicable and relies on resampling the data many times to compute some measure of variability

- given modern computing power, large-scale resampling is often easy to do.

The bootstrap approach is particularly useful for problems where analytic measures of variability are not available (or not automatically output by statistical software)

- even when analytic measures are available, they are often based on large-sample theory
  - and may therefore not be very accurate in finite-data settings.

The easiest way to explain the bootstrap approach is via an example

- we will follow the main example and development of ISLR.
- Ruppert's *Statistics and Data Analysis for FE* has further (finance) examples.

The Bootstrap via a Minimum Variance Portfolio Example

We wish to invest a fixed sum of money in two financial assets, X and Y say, that yield random returns of R_x and R_y, respectively.

We will invest a fraction α of our money in X, and will invest the remaining $1-\alpha$ in the Y.

We wish to choose α to minimize the total variance of our investment return.

Therefore want to minimize $\mathrm{Var}(\alpha R_x + (1-\alpha)R_y)$.

It is easy to see that the minimizing α is given by

$$\alpha = \frac{\sigma_y^2 - \sigma_{xy}}{\sigma_x^2 + \sigma_y^2 - 2\sigma_{xy}} \tag{1}$$

where $\sigma_x^2 = \mathrm{Var}(R_x)$, $\sigma_y^2 = \mathrm{Var}(R_y)$ and $\sigma_{xy} = \mathrm{Cov}(R_x, R_y)$.

In practice, we do not know these quantities and therefore have to estimate them from data and then use

$$\hat{\alpha} = \frac{\hat{\sigma}_y^2 - \hat{\sigma}_{xy}}{\hat{\sigma}_x^2 + \hat{\sigma}_y^2 - 2\hat{\sigma}_{xy}}. \tag{2}$$

Question: How good is $\hat{\alpha}$? Or more specifically, what sort of error do we expect in using $\hat{\alpha}$?

The Bootstrap via a Minimum Variance Portfolio Example

Suppose (for now) we know the true model: $\sigma_x^2 = 1$, $\sigma_y^2 = 1.25$ and $\sigma_{xy} = 0.5$

- so $\alpha = 0.6$ by (1).

Then it is easy(!) to answer to our question: use Monte-Carlo!

1. Simulate N samples from the true model. ($N = \#$ samples in the data-set.)
2. Use (2) to compute $\hat{\alpha}$.
3. Repeat steps 1 and 2 B times to obtain $\hat{\alpha}_1, \ldots, \hat{\alpha}_B$.

Can now compute the bias, standard deviation and mean-squared error (MSE):

$$\begin{aligned} \text{bias} &= \alpha - \bar{\alpha} \\ \text{st.dev.} &= \left(\frac{\sum_{i=1}^{B} \left(\hat{\alpha}_i - \bar{\alpha}\right)^2}{B-1} \right)^{1/2} \\ \text{MSE} &= \left(\frac{\sum_{i=1}^{B} \left(\hat{\alpha}_i - \alpha\right)^2}{B} \right)^{1/2} \end{aligned}$$

where $\bar{\alpha} := \left(\sum_{i=1}^{B} \hat{\alpha}_i\right)/B$.

The Bootstrap via a Minimum Variance Portfolio Example

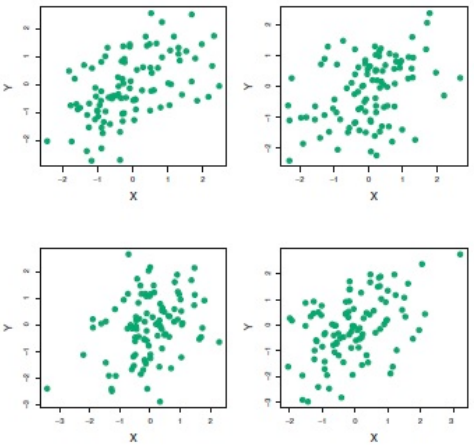

Figure 5.9 from ISLR: Each panel displays 100 simulated returns for investments in X and Y. From left to right and top to bottom, the resulting estimates for α are 0.576, 0.532, 0.657, and 0.651.

The Bootstrap via a Minimum Variance Portfolio Example

Figure 5.9 from ISLR displays four simulations of 100 return vectors from the true (known) distribution of joint returns.

For each such simulation we can estimate α.

Based on $B = 1,000$ such simulations, we obtained an average value of 0.6008 with a standard deviation of 0.079

- so the bias is extremely small
- and for a random sample (of size N) from the population we would expect $\hat{\alpha}$ to deviate from α by approximately 0.079 on average.

Question: How do you think the standard deviation would vary with N?

Unfortunately we do not know the true model!

- so what can we do?

This is where the bootstrap comes to the rescue!

The Bootstrap via a Minimum Variance Portfolio Example

Suppose we have available a data-set $(R_x^1, R_y^1),\ \ldots,\ (R_x^N, R_y^N)$.

The bootstrap works by taking the empirical distribution of the data to be the true distribution

- and proceeds as before by simulating B samples (each of size N) from this (empirical) distribution.
- Each of the B samples is called a bootstrap sample.

Very important that each bootstrap sample is obtained by sampling with replacement (why?) from the empirical distribution.

Returning to our minimum variance portfolio example, we can therefore estimate the variation in α:

1. We are given a data-set as above: $(R_x^1, R_y^1),\ \ldots,\ (R_x^N, R_y^N)$ with $N = 100$
    - which we simulated from the true distribution.
2. We then simulated $B = 1,000$ bootstrap samples of size N from this data-set.
3. For each bootstrap sample we estimated the variances and covariances and then used (2) to obtain $\hat{\alpha}_1^b, \ldots, \hat{\alpha}_B^b$

The Bootstrap via a Minimum Variance Portfolio Example

4. We found the st.dev of the $\hat{\alpha}_i^b$'s to be $.082$
   - which is very close to the value of $.079$ that we obtained when we simulated from the true distribution!

Question: Do you think the average of the $\hat{\alpha}_i^b$'s would be close to the true value of 0.6? Why or or why not?

Hint: See Figure 5.10 from ISLR on the next slide!

The bootstrap is an extremely simple yet powerful method for estimating the variability of a parameter estimate.

Question: How would you use the bootstrap to assess the variability of the predictions from a statistical learning method?

The Bootstrap via a Minimum Variance Portfolio Example

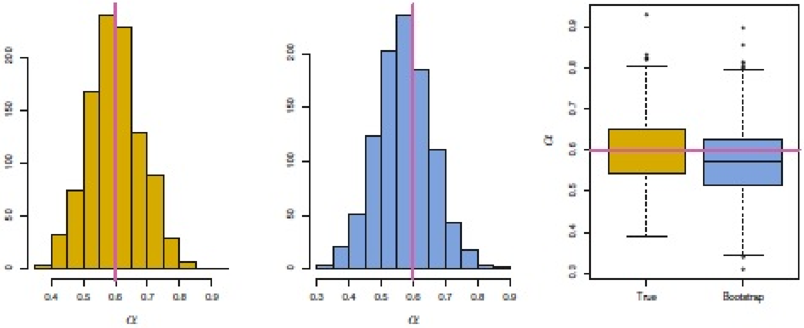

Figure 5.10 from ISLR: Left: A histogram of the estimates of α obtained by generating 1,000 simulated data sets from the true population. Center: A histogram of the estimates of α obtained from 1,000 bootstrap samples from a single data set. Right: The estimates of α displayed in the left and center panels are shown as boxplots. In each panel, the pink line indicates the true value of α.

Estimating Confidence Intervals Via the Bootstrap

The bootstrap is widely used to construct confidence intervals.

We will consider the so-called basic bootstrap interval

- but there are more sophisticated approaches – see for example Section 6.3 of Ruppert and Matteson's *Statistics and Data Analysis for Financial Engineering*.

Consider again the minimum variance portfolio example with bootstrap samples $\hat{\alpha}_1^b, \ldots, \hat{\alpha}_B^b$ and suppose we want a $1 - \alpha$ confidence interval for α.

Let q_l and q_u be the $\alpha/2$ lower- and upper-sample quantiles, respectively, of the bootstrap samples.

Then the fraction of bootstrap samples satisfying

$$q_l \leq \hat{\alpha}^b \leq q_u \tag{3}$$

is $1 - \alpha$.

Estimating Confidence Intervals Via the Bootstrap

But (3) is equivalent to

$$\hat{\alpha} - q_u \leq \hat{\alpha} - \hat{\alpha}^b \leq \hat{\alpha} - q_l \tag{4}$$

where $\hat{\alpha}$ is our estimate of α computed using the original data-set

- so $\hat{\alpha} - q_u$ and $\hat{\alpha} - q_l$ are the lower and upper quantiles for $\hat{\alpha} - \hat{\alpha}^b$.

The basic bootstrap assumes they are also the quantiles for $\alpha - \hat{\alpha}$

- makes sense since the empirical distribution of the original data-set is taken to be the "true" distribution by the bootstrap.

It therefore follows that

$$\hat{\alpha} - q_u \leq \alpha - \hat{\alpha} \leq \hat{\alpha} - q_l \tag{5}$$

will occur in approximately in a fraction $1 - \alpha$ of samples.

Adding $\hat{\alpha}$ across (5) yields an approximate $(1 - \alpha)\%$ CI for α of

$$(2\hat{\alpha} - q_u,\, 2\hat{\alpha} - q_l).$$

Other Variations of the Bootstrap

There are also other variations of the bootstrap:

1. Instead of simulating from the original data-set, we simulate from a distribution that was fitted using the original data-set
    - this is the parametric bootstrap.
2. We could also sample (with replacement) from the residuals of a fitted model.
3. There are also approaches for handling dependent data such as time-series data.

When Does the Bootstrap Fail?

The bootstrap approach is an extremely powerful and broadly applicable tool.

But it doesn't always work and can provide faulty inference in the following situations:

1. When there is too little data
   - e.g. suppose there is just one data-point!
2. When trying to approximate the sampling distribution of an **extreme order statistic**
   - e.g. $\theta := \max_{1 \le i \le N} X_i$.
3. When dealing with **heavy-tailed** distributions.
4. When the data is not IID
   - e.g. with **time-series** data a naive implementation of the bootstrap will be incorrect. Why?
   - but there are methods for bootstrapping from time-series data.