DSO530 Statistical Learning Methods

Lecture 5: Cross-Validation and Bootstrap

Dr. Xin Tong
Department of Data Sciences and Operations
Marshall School of Business
University of Southern California

Resampling methods

- *Resampling methods*: repeatedly draw **samples** from a training set and refit a model of interest (or compute certain estimates) on each sample in order to obtain additional information about the fitted model (or those estimates)
- Common resampling methods include: cross-validation and bootstrap
- This set of slides communicate the main ideas of these methods
- Python tutorial covers the impelmentation
- Some people tend to treat a large test dataset as the population and not worry about the difference between them. We do not endorse this review though.

Cross-validation (CV)

- In most problems, there is no designated “test dataset” that is huge in size and set aside a priori.
- Cross-validation (CV) can be used to estimate the test error associated with a given statistical learning method
  - to evaluate its performance (*model assessment*) This is actually a subtle point. We will come back and answer the why question.
  - or to select the appropriate level of flexibility (*model selection*)
- When to use CV to estimate test error?
  - when you don’t have a designated test set
- CV can be used for both classification and regression
- CV has a few variants; we only discuss the canonical version
- A precursor of CV is the validation set approach

The validation set approach

- Validation set approach: randomly divide the available set of observations into two (equal) parts, a training set and a validation set or 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. We have practiced something similar multiple times already in our tutorials and homework.
- Drawbacks of this approach:
  - the validation estimate of the test error rate can be highly variable (Recall the p.4 and p.5 in HW1)
  - only about a half of the observations are used to train the model (inefficient use of data).
- Cross-validation: a refinement of the validation set approach that addresses these two issues.

Leave-One-Out Cross-Validation (LOOCV)

- We first illustrate in the context of regression
- Suppose the training data contains $\{(x_1, y_1), \cdots, (x_n, y_n)\}$
- First, use $(n-1)$ observations $\{(x_2, y_2), \cdots, (x_n, y_n)\}$ to train and use the remaining observation (x_1, y_1) to evaluate the performance: $MSE_1 = (y_1 - \hat{y}_1)^2$
- Repeat this procedure by using (x_2, y_2) for the validation data, training on the $n-1$ observations $(x_1, y_1), (x_3, y_3), \cdots (x_n, y_n)$, and compute $MSE_2 = (y_2 - \hat{y}_2)^2$
- Repeat this approach n times produces n squared errors $MSE_1, \cdots, MSE_n$
- The LOOCV estimate for the test MSE is the average of these n estimates:

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

- There is no need to randomly shuffle the training data before implementing LOOCV. Why?

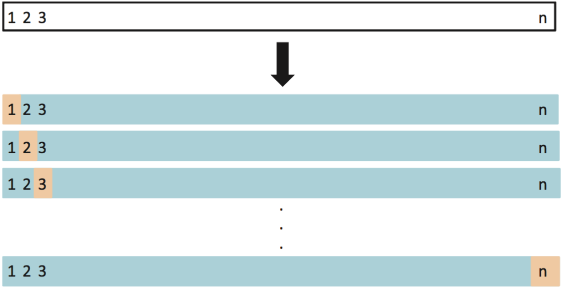

FIGURE 5.3. *A schematic display of LOOCV. A set of n data points is repeatedly split into a training set (shown in blue) containing all but one observation, and a validation set that contains only that observation (shown in beige). The test error is then estimated by averaging the n resulting MSE's.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

Figure 1: LOOCV

Potential problems for LOOCV

- Computationally, LOOCV has the potential to be expensive to implement, since the model has to be fit n times
    - (Optional) However, with least squares linear or polynomial regression, the following formula holds:

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

    - (Optional) where h_i is the leverage of the ith observation defined in Chap 3 of `ISLR`
- The n fitted models are each trained on an almost identical set of observations, and so averaging does not reduce variance much
    - Think about letting $X_i \sim N(0,1)$ but insist $X_1 = \cdots = X_n$
    - What is the variance of $\bar{X}$?

k-fold CV

- k-fold CV: randomly divide the set of observations into k groups, or folds, of approximately equal size. The first fold is treated as a validation set, and the method is fit on the remaining $k-1$ folds. The mean squared error, MSE_1, is then computed on the observations in the held-out fold
- This procedure is repeated k times; each time, a different group of observations is treated as a validation set
- This process results in k estimates of the test error, $MSE_1, MSE_2, \cdots, MSE_k$
- The k-fold CV estimate is computed by averaging these values,

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

- We commonly use $CV_{(k)}$ to estimate the test error (model evaluation)

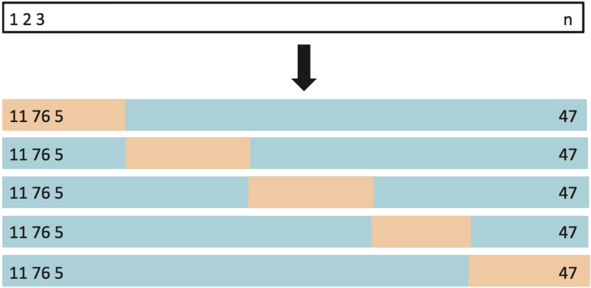

FIGURE 5.5. *A schematic display of 5-fold CV. A set of n observations is randomly split into five non-overlapping groups. Each of these fifths acts as a validation set (shown in beige), and the remainder as a training set (shown in blue). The test error is estimated by averaging the five resulting MSE estimates.*

Figure 2: 5-fold CV

- LOOCV is a special case of k-fold CV in which $k = n$

Bias-variance trade-off for k-fold CV

- The larger the k, the smaller "bias"
- However, large k (e.g., $k = n - 1$) means higher "variance". Two reasons of higher variance. We mentioned one, and there is another one. What is it?
- So for prediction accuracy, the better choices are usually not the extreme choices for k.
- The usual choices in practice for k in k-fold CV are $k = 5$ and $k = 10$.
- Why do we care about bias and variance here? Think about our goal: model assessment and model selection

k-fold CV for model selection

- Suppose in an Auto dataset, we are deciding between two (prediction) models
- model 1: use displacement to predict mpg
- model 2: use displacement and horsepower to predict mpg
- To choose between these two models using 5-fold CV, we calculate $CV_{(5)}$ for both models 1 and 2, and choose the model with the smaller $CV_{(5)}$.
- The same idea extends to the case where we need to choose among many models.
- After you have chosen the model, which of the 5 linear prediction equations do you actually use?
- Answer: none. We refit the model with all available data.
- But then how can we know the performance of this final fitted model?

k-fold CV for model selection

- Suppose in an `Auto` dataset, we are deciding between two (prediction) models
- model 1: use `displacement` to predict `mpg`
- model 2: use `displacement` and `horsepower` to predict `mpg`
- To choose between these two models using 5-fold CV, we calculate $CV_{(5)}$ for both models 1 and 2, and choose the model with the smaller $CV_{(5)}$.
- The same idea extends to the case where we need to choose among many models.
- After you have chosen the model, which of the 5 linear prediction equations do you actually use?
- Answer: none. We refit the model with all available data.
- But then how can we know the performance of this final fitted model?

k-fold CV for model selection

- Suppose in an `Auto` dataset, we are deciding between two (prediction) models
- model 1: use `displacement` to predict `mpg`
- model 2: use `displacement` and `horsepower` to predict `mpg`
- To choose between these two models using 5-fold CV, we calculate $CV_{(5)}$ for both models 1 and 2, and choose the model with the smaller $CV_{(5)}$.
- The same idea extends to the case where we need to choose among many models.
- After you have chosen the model, which of the 5 linear prediction equations do you actually use?
- Answer: none. We refit the model with all available data.
- But then how can we know the performance of this final fitted model?

Use k-fold CV for classification

- The basic idea of CV for classification is the same as that for regression, except the way to calculate $CV_{(k)}$
  - For example, the LOOCV error rate is

$$CV_{(n)} = \frac{1}{n} \sum_{i=1}^{n} Err_i \,,$$

  - where $Err_i = I(y_i \neq \hat{y}_i)$

Bootstrap

Figure 3: Picture from Wikipedia

Bootstrap in Statistics

- Bootstrap: in statistics, *bootstrap* is any procedure that relies on random sampling with replacement (from the original sample)
- Bootstrap allows assigning *measures of accuracy* (defined in terms of standard error, variance, confidence intervals, prediction error or some other such measure) to *sample estimates*.
- This is a rather strange idea when we first look at it
- On the other hand, it is one of the most influential ideas in Statistics invented in the the second half of the 20th century
- As a verb, bootstrap means "get oneself out of a situation using existing resources (without extra help)" (quote: pull oneself over a fence by one's bootstraps)
- Bootstrap is computationally heavy
- It is a class of widely used procedures. But its theory is beyond the scope of DSO 530
- Bootstrap has different versions (e.g., moving block bootstrap for time series data); we only discuss the simplest kind
- We illustrate the basic bootstrap idea with two toy examples

What is “sample with replacement” (the first example)?

- Suppose there is a box that contains a black ball and a red ball
- Draw one ball from the box, put it back, and then draw another ball from the box; repeat this process multiple times (see different outcomes?)

```
import numpy as np
np.random.seed(2); color = ['red', 'black']
np.random.choice(color, size = 2, p=[0.5, 0.5]) # p is optional
```

```
## array(['red', 'red'], dtype='<U5')
```

```
np.random.choice(color, size = 2, p=[0.5, 0.5], replace = True)
```

```
## array(['black', 'red'], dtype='<U5')
```

```
np.random.choice(color, size = 2, p=[0.5, 0.5], replace = True)
```

```
## array(['red', 'red'], dtype='<U5')
```

- When we do sample with replacement, it is possible to get one element twice? What is the default value for ‘replace’?
- What if we sample two elements without replacement from the above box?

A toy investment example (the second example)

- Now you understand “sample with replacement”. How to use it?
- Suppose we want to invest a fixed sum of money in two financial assets that yield returns of X and Y
- We will invest α fraction in X and $1-\alpha$ in Y
- Want to choose α to minimize the total risk, i.e.,

$$Var(\alpha X + (1-\alpha)Y)$$

- It can be shown that the α value giving the minimum risk is

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}},$$

- where $\sigma_X^2 = Var(X)$, $\sigma_Y^2 = Var(Y)$ and $\sigma_{XY} = Cov(X, Y)$
- Based on data (past measurements of X and Y), we can get estimates of the above quantities: $\hat{\sigma}_X^2$, $\hat{\sigma}_Y^2$ and $\hat{\sigma}_{XY}$.
- Then the optimal α can be estimated by

$$\hat{\alpha} = \frac{\hat{\sigma}_Y^2 - \hat{\sigma}_{XY}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\hat{\sigma}_{XY}}$$

A toy investment example (cont')

- Wish to quantify the accuracy of our estimate $\hat{\alpha}$ of α, such as $SE(\hat{\alpha})$
- A quote: "one should not use an estimate (a procedure) unless one is able to quantify its accuracy"
- Think about $\bar{X} = (X_1 + \cdots + X_n)/n$, where $X_i \sim N(\mu, \sigma^2)$
- What is the formula for standard deviation of $\bar{X}$?
- Do NOT know a formula for $SE(\hat{\alpha})$ (a common situation)
- If you knew the population, you can do repetitive sampling from the population, and . . . ..
- We resort to bootstrap to approximate $SE(\hat{\alpha})$
- Here is what we do:
  - sample with replacement B (e.g., $1,000$) times from the original example
  - resulting in B different bootstrap datasets $Z^{*1}, Z^{*2}, \cdots, Z^{*B}$
  - Apply the same $\hat{\alpha}$ formula to these datasets and get $\hat{\alpha}^{*1}, \hat{\alpha}^{*2}, \cdots, \hat{\alpha}^{*B}$
  - Approximate $SE(\hat{\alpha})$ by the sample standard deviation of $\{\hat{\alpha}^{*1}, \hat{\alpha}^{*2}, \cdots, \hat{\alpha}^{*B}\}$:

$$SE_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1}\sum_{r=1}^{B}\left(\hat{\alpha}^{*r} - \frac{1}{B}\sum_{r'=1}^{B}\hat{\alpha}^{*r'}\right)^2}$$

A toy investment example (cont')

- Wish to quantify the accuracy of our estimate $\hat{\alpha}$ of α, such as $SE(\hat{\alpha})$
- A quote: "one should not use an estimate (a procedure) unless one is able to quantify its accuracy"
- Think about $\bar{X} = (X_1 + \cdots + X_n)/n$, where $X_i \sim N(\mu, \sigma^2)$
- What is the formula for standard deviation of $\bar{X}$?
- Do NOT know a formula for $SE(\hat{\alpha})$ (a common situation)
- If you knew the population, you can do repetitive sampling from the population, and .....
- We resort to bootstrap to approximate $SE(\hat{\alpha})$
- Here is what we do:
  - sample with replacement B (e.g., $1,000$) times from the original example
  - resulting in B different bootstrap datasets $Z^{*1}, Z^{*2}, \cdots, Z^{*B}$
  - Apply the same $\hat{\alpha}$ formula to these datasets and get $\hat{\alpha}^{*1}, \hat{\alpha}^{*2}, \cdots, \hat{\alpha}^{*B}$
  - Approximate $SE(\hat{\alpha})$ by the sample standard deviation of $\{\hat{\alpha}^{*1}, \hat{\alpha}^{*2}, \cdots, \hat{\alpha}^{*B}\}$:

$$SE_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1}\sum_{r=1}^{B}\left(\hat{\alpha}^{*r} - \frac{1}{B}\sum_{r'=1}^{B}\hat{\alpha}^{*r'}\right)^2}$$