# DSO530 Statistical Learning Methods

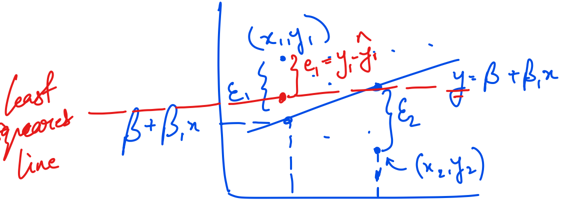
## Lecture 5: Cross-Validation and Bootstrap

Dr. Xin Tong
Department of Data Sciences and Operations
Marshall School of Business
University of Southern California
xint@marshall.usc.edu

Regression : $y = \beta_0 + \beta_1 x + \varepsilon$ , $\varepsilon$ is $N(0, \sigma^2)$

# Resampling methods

∴ residual can be different from error, but as sample size improves, red & blue lines tend to converge.



- *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 of cross-validation

Test set should be large enough, however sometimes it is not big enough.

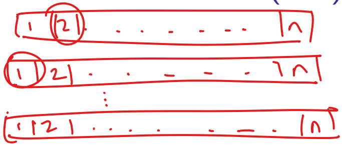
train
validation

You want both parts to be large

→ leave one out cross validation (LOOCV)

# Cross-validation (CV)

each model will have MSE which will be averaged out.

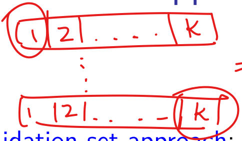


- In most problems, there is no designated "test dataset" that is huge in size and set aside a priori.
- Cross-validation (CV) can be used to estimate the test error associated with a given statistical learning method
  - to evaluate its performance (*model assessment*)
  - or to select the appropriate 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

k-fold CV

k = 5,10.. etc



⇒ performance metric aggregation

- 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. Fit a model 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.5 in HW1) e.g. writing new test & scoring higher than old ones.
  - 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? Random shuffling does not change $CV_{(n)}$ at all.

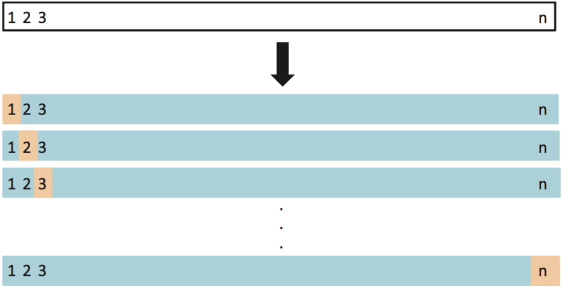


**FIGURE 5.3.** *A schematic display of LOOCV. A set of $n$ data points is repeatedly split into a training set (shown in blue) containing all but one observation, and a validation set that contains only that observation (shown in beige). The test error is then estimated by averaging the $n$ resulting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

# k-fold CV

- Computationally, LOOCV has the potential to be expensive to implement, since the model has to be fit $n$ times
- 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 hold-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,

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

average of performance metric

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

Q. Is the validation set approach the same as 2 fold CV? No!!

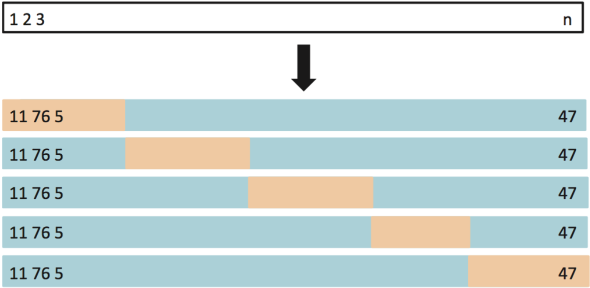


**FIGURE 5.5.** *A schematic display of 5-fold CV. A set of $n$ observations is randomly split into five non-overlapping groups. Each of these fifths acts as a validation set (shown in beige), and the remainder as a training set (shown in blue). The test error is estimated by averaging the five resulting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

Figure 2: 5-fold CV

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

# About k-fold CV

- The usual choices in practice for $k$ in k-fold CV are $k = 5$ and $k = 10$.
- Sometimes, when people have enough computation power, they do 10-fold CV multiple times and then take the average performance
- The cross-validation cretirion should be consistent with prediction evaluation criterion (group discussion on the following questions):
  - think about the MSE in k-fold CV for regression we just introduced. Are they (other than LOOCV) consistent with the out-of-sample $R^2$?
  - For classification, if we care about classification error, what CV criterion should we adopt? How about the CV criterion if we care about ROC AUC?

Ans 1]: How are MSE & $R^2$ similar: (i) $MSE_{(k)} = 1/k \sum_{i=1}^{k} MSE_i$, $R^2_{(k)} = \frac{1}{k} \sum_{i=1}^{k} R_i^2$ where

$MSE_i = \frac{1}{I_i} \sum_{j \in I_i} (y_j - \hat{y}_j^2)^2$ $\quad R^2 = 1 - \frac{RSS_i}{TSS_i} = 1 - \sum_{j \in I_i} (y_j - \hat{y}_j)^2 / \sum_{j \in I_i} (y_j - \bar{y}_i)^2$

then which one to use: people use both, stat → MSE CS → $R^2$.

For LOOCV, use MSE

# k-fold CV for model selection

- Suppose in an Auto dataset, we are deciding between two linear (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 better $CV_{(5)}$.
- 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 model?

# k-fold CV for model selection

- Suppose in an Auto dataset, we are deciding between two linear (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 better $CV_{(5)}$.
- 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 model?

# k-fold CV for model selection

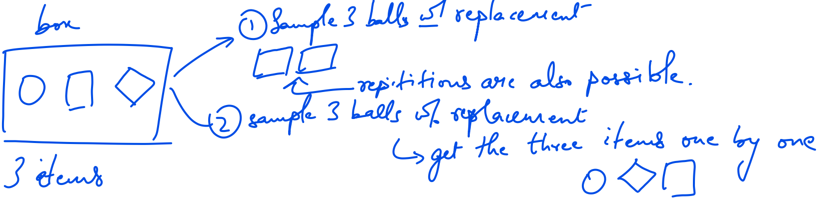
- Suppose in an Auto dataset, we are deciding between two linear (prediction) models
- model 1: use displacement to predict mpg
- model 2: use displacement and horsepower to predict mpg
- To choose between these two models using 5-fold CV, we calculate $CV_{(5)}$ for both models 1 and 2, and choose the model with the better $CV_{(5)}$.
- 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 model?

↳ Since everything is used as training, there is nothing left for evaluation.

# Bootstrap

Figure 3: Picture from Wikipedia

# Bootstrap in Statistics



① Sample 3 balls w/ replacement

repetitions are also possible.

② sample 3 balls w/o replacement

↳ get the three items one by one

- Bootstrap: in statistics, *bootstrap* is any procedure that relies on random sampling with replacement (from the original sample)
- "random sampling with replacement" vs. "random sampling without replacement"

# Bootstrap in Statistics

- 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

# Execute “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?

replace = False (for w/o replacement)

# 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 $\alpha$ fraction in $X$ and $1-\alpha$ in $Y$
- Want to choose $\alpha$ to minimize the total risk, i.e.,

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

- It can be shown that the $\alpha$ 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 $\alpha$ can be estimated by

$$\begin{aligned}
\text{sample } 1 &\longrightarrow \hat{\alpha}_1 \\
\text{sample } 2 &\longrightarrow \hat{\alpha}_2 \\
\vdots \quad & \qquad \vdots \\
\text{sample } n &\longrightarrow \hat{\alpha}_n
\end{aligned}$$

$$\hat{\alpha} = \frac{\hat{\sigma}_Y^2 - \hat{\sigma}_{XY}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\hat{\sigma}_{XY}} \longrightarrow \text{minimizing total risk}$$

# A toy investment example (cont')

sample 1

↑

pretend that this is the population.

- Wish to quantify the accuracy of our estimate $\hat{\alpha}$ of $\alpha$, 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 . . . ..

# A toy investment example (cont')

- 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 $\{\hat{\alpha}^{*1}, \hat{\alpha}^{*2}, \cdots, \hat{\alpha}^{*B}\}$:

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

Bootstrap will appear in an important algorithm that we will introduce soon.

# A toy investment example (cont')

- 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 $\{\hat{\alpha}^{*1}, \hat{\alpha}^{*2}, \cdots, \hat{\alpha}^{*B}\}$:

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

Bootstrap will appear in an important algorithm that we will introduce soon.