Ch 5.1.5-5.2: Cross-Validation for Classification and Bootstrap

Lecture 10 - CMSE 381

Michigan State University
::
Dept of Computational Mathematics, Science & Engineering

Mon, Feb 19, 2024

Announcements

Last time:

- CV for regression

This lecture:

- CV for classification
- bootstrap

Announcements:

- Homework #5 is Due Wed
- Grades

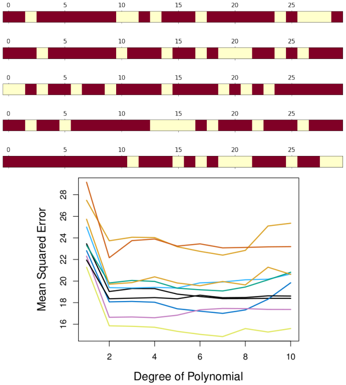
Percent	Convert
≥ 90%	4.0
≥ 85%	3.5
≥ 80%	3
≥ 75%	2.5
≥ 70%	2
≥ 65%	1.5
≥ 60%	1
< 60%	0

Section 1

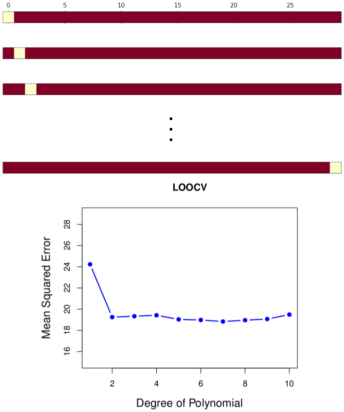
Last time

Approximations of Test Error

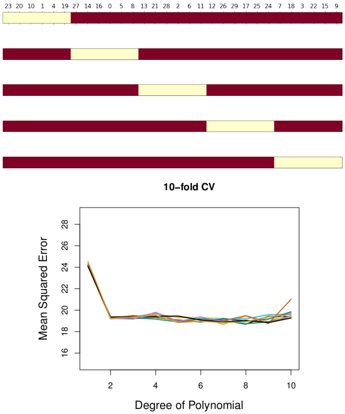
Validation Set



LOOCV

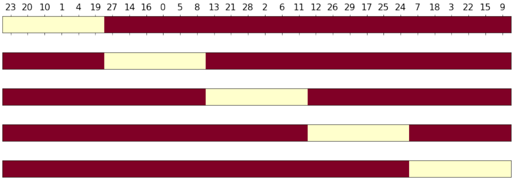


K-fold CV



Definition of k-fold CV

- Randomly split data into k-groups (folds)
- Approximately equal sized. For the sake of notation, say each set has ℓ points
- Remove ith fold U_i and reserve for testing.
- Train the model on remaining points
- Calculate $\mathrm{MSE}_i = \frac{1}{\ell} \sum_{(x_j, y_j) \in U_i} (y_j - \hat{y}_j)^2$
- Rinse and repeat



Return

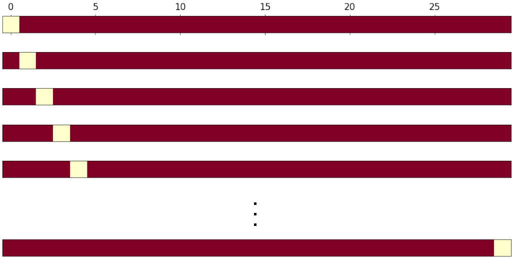
$$CV_{(k)} = \frac{1}{k} \sum_{i=1}^{k} \mathrm{MSE}_i$$

Section 2

CV for Classification

Setup: LOOCV

- Remove ith point (x_i, y_i) and reserve for testing.
- Train the model on remaining points
- Calculate $\mathrm{Err}_i = \mathrm{I}(y_j \neq \hat{y}_j)$
- Rinse and repeat

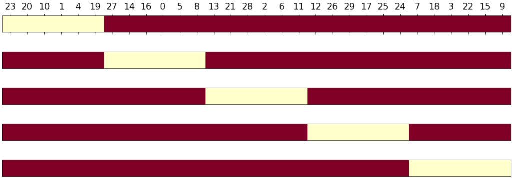


Return

$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{Err}_i$$

Setup: k-fold

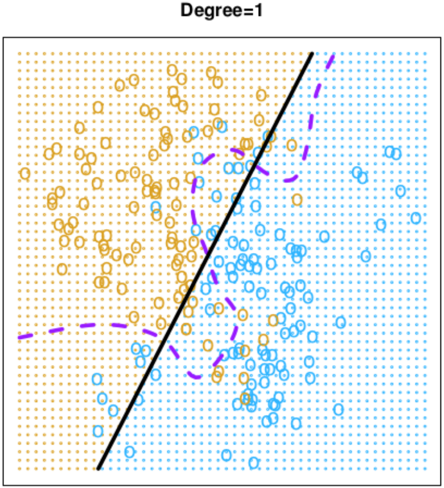
- Randomly split data into k-groups (folds)
- Approximately equal sized. For the sake of notation, say each set has ℓ points
- Remove ith fold U_i and reserve for testing.
- Train the model on remaining points
- Calculate $\mathrm{Err}_i = \frac{1}{\ell} \sum_{(x_j,y_j)\in U_i} \mathrm{I}(y_j \neq \hat{y}_j)$
- Rinse and repeat



Return

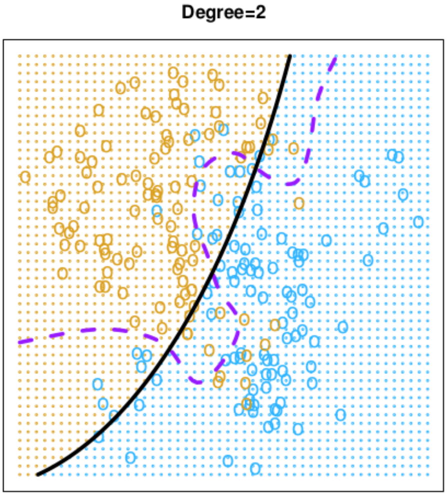
$$CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} \mathrm{Err}_i$$

Example on simulated data: Linear



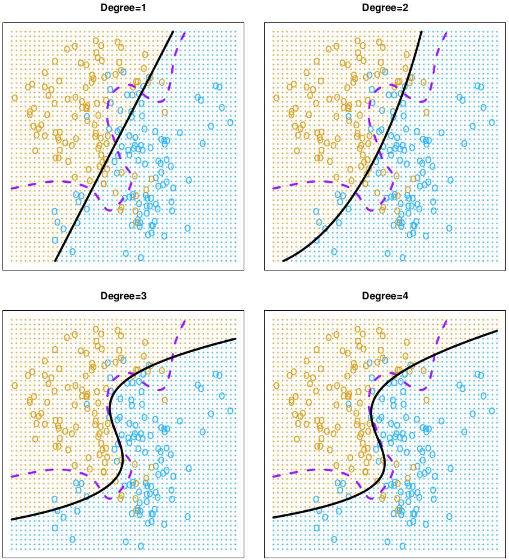
- Purple: Bayes decision boundary.
  - Error rate: 0.133
- Black: Logistic regression
  - $\log(p/(1-p)) = \beta_0 + \beta_1 X_1 + \beta_2 X_2$
  - Error rate: 0.201
- No k-fold yet
- Error rate for logistic still high, so not great job yet
- So, up the degree and see what happens

Example on simulated data: Quadratic logistic regression



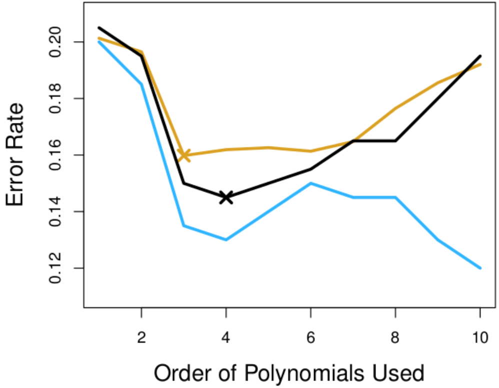
- Purple: Bayes decision boundary.
  - Error rate: 0.133
- Black: Logistic regression
  - $\log(p/(1-p)) =$ $\beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_2 + \beta_4 X_2^2$
  - Error rate: 0.197
- No k-fold yet
- Error rate for logistic slight improvement, but still not great
- So, up the degree again and see what happens

Example on simulated data: all the polynomials!



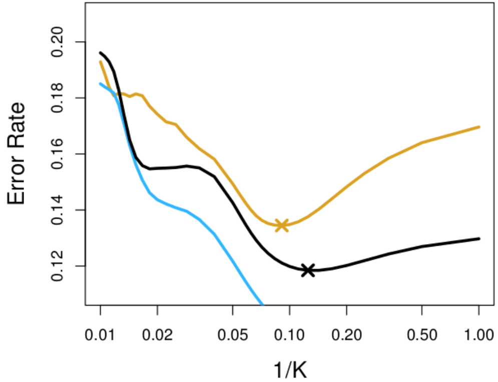
- Purple: Bayes decision boundary.
  - Error rate: 0.133
- Black: Logistic regression
  - Deg 1 Error rate: 0.201
  - Deg 2 Error rate: 0.197
  - Deg 3 Error rate: 0.160
  - Deg 4 Error rate: 0.162
- Normally, we don't have the Bayes decision boundary
- How do you pick between models?

Decide degree based on CV



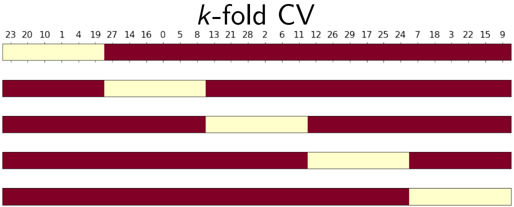
- Test error (brown)
- Training error (blue)
- 10-fold CV error (black)
- Training error tends to decrease as the flexibility of the fit increases.
- Note decrease in blue isn't monotonic, but still going down overall
- test error displays a characteristic U-shape.
- The 10-fold CV provides a pretty good (but underestimate) approximation to the test error rate.
- minimum at 4, close to minimum of test curve at 3

Similar game for KNN



- Test error (brown)
- Training error (blue)
- 10-fold CV error (black)
- Note that using a different changing parameter
- similar idea to figure out the right choice of K.
- Minimm in black (10 fold CV) has $1/K$ value close to where brown does, so good choice

k-folder CV for classification



$$CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} \mathrm{MSE}_i$$

Use $k = 5$ or 10 usually

k-fold CV for classification

$$\mathrm{Err}_i = \mathrm{I}(y_j \neq \hat{y}_j)$$

$$CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} \mathrm{Err}_i$$

Section 3

The Bootstrap

The Idea

The goal: quantify the uncertainty associated with a given estimator or statistical learning method.

The method: Bootstrap

- Can be used to estimate std error of linear regression coefficients, but that's boring since we have other tools
- the power of the bootstrap lies in the fact that it can be easily applied to a wide range of statistical learning methods,
- including some for which a measure of variability is otherwise difficult to obtain and is not automatically output by statistical software.

The boostrap idiom

"Pull yourself up by your bootstraps."

- Originally used as a saying meaning an impossible task, used sarcastically
- Now often used to imply that socioeconomic advancement is something everyone should be able to do
- Also source of the term "booting" a computer

Today's class: Bootstrap on a simple modeling problem

- We wish to invest a fixed sum of money in two financial assets that yield returns of X and Y, respectively, where X and Y are random quantities.
- We will invest a fraction α of our money in X, and will invest the remaining $1 - \alpha$ in Y.
- Since there is variability associated with the returns on these two assets, we wish to choose α to minimize the total risk, or variance, of our investment.

Really: want to minimize $\mathrm{Var}(\alpha X + (1 - \alpha)Y)$

One can show.....

...that $\mathrm{Var}(\alpha X + (1-\alpha)Y)$
is minimzed by

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

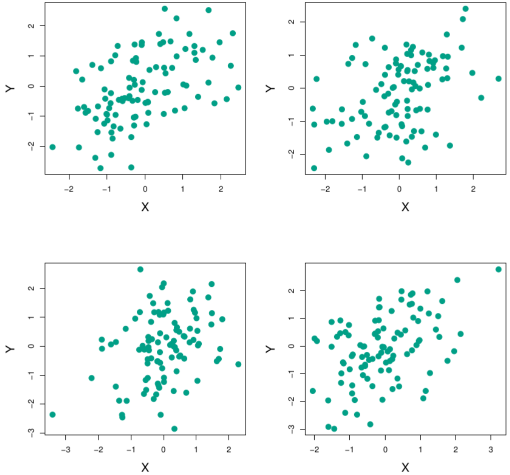
where

- $\sigma_X^2 = \mathrm{Var}(X)$
- $\sigma_Y^2 = \mathrm{Var}(Y)$
- $\sigma_{XY} = \mathrm{Cov}(X, Y)$

Get an estimate:

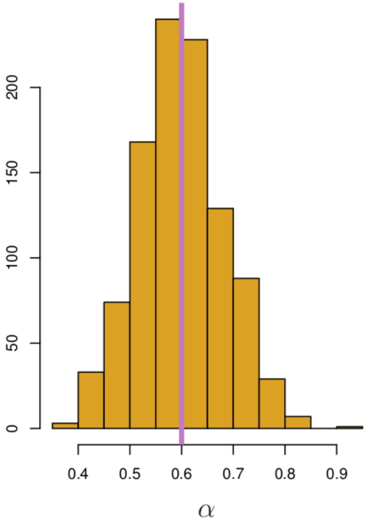
$$\hat{\alpha} = \frac{\hat{\sigma}_Y^2 - \hat{\sigma}_{XY}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\hat{\sigma}_{XY}}$$

Simulated data



- Simulate data using
  - $\sigma_X^2 = 1$
  - $\sigma_Y^2 = 1.25$
  - $\sigma_{XY} = 0.5$
  - Implies $\alpha = 0.6$
- In each panel: Simulate 100 pairs of returns for investments
- Predict σ_X^2, σ_Y^2, and σ_{XY}
- $\hat{\alpha}$ prediction by panel:
  0.576   0.543
  0.657   0.651

Resimulate: Rinse and repeat



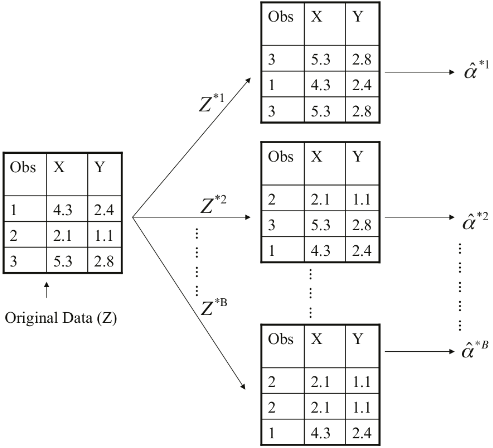
- Simulate 100 data points 1,000 times
- Left: Histogram of predictions for α
- Pink line: True value for α
- Mean over simulated values:
  $0.5996 = \bar{\alpha} = \frac{1}{1000} \sum \hat{\alpha}_r$
- St dev: $0.083 = \sqrt{\frac{1}{1000-1} \sum (\hat{\alpha}_r - \bar{\alpha})}$

So what's the problem?

I can't simulate my data!

So the bootstrap plan..... create simulations from the original data set

The solution: Sample the data with replacement



- Note that data points can show up repeatedly
- Note that the size of the sample is always n, but with repeats
- Different from the validation set approach since there we had no repeats; also cared more about error than about the models

Computation of error

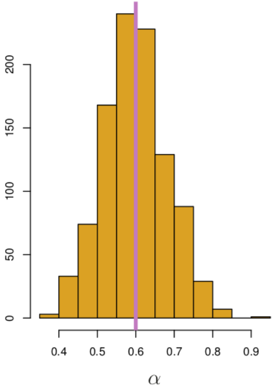
- Repeate procedure B times:
- Get B bootstrap data sets, $Z^{*1}, Z^{*2}, \ldots, Z^{*B}$
- Get B bootstrap estimates $\hat{\alpha}^{*1}, \hat{\alpha}^{*2}, \ldots, \hat{\alpha}^{*B}$

Get standard error estimate:

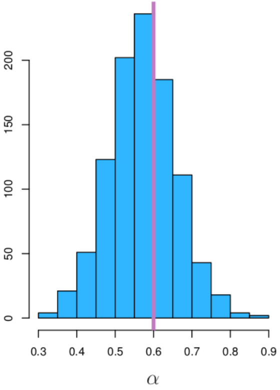
$$SE_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1} \sum_{r=1}^{B} \left(\hat{\alpha}^{*r} - \frac{1}{B} \sum_{r'=1}^{B} \hat{\alpha}^{*r'} \right)^2}$$

- This is the estimate of standard deviation of the alpha hats
- If you use 'np.std' you need to be careful because it doesn't do divde by N-1 without some extra flag

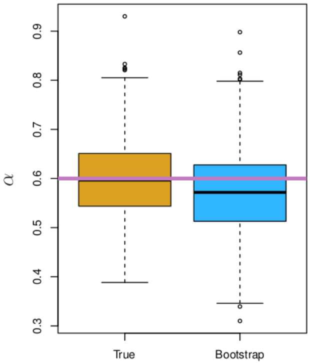
Back to the example



Resample version
Predicted $SE(\hat{\alpha}) = 0.083$

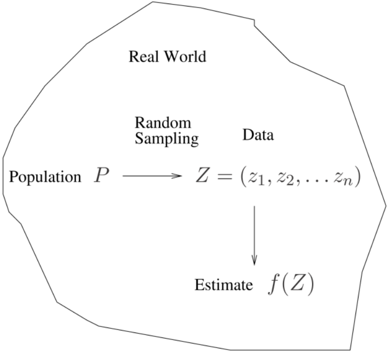


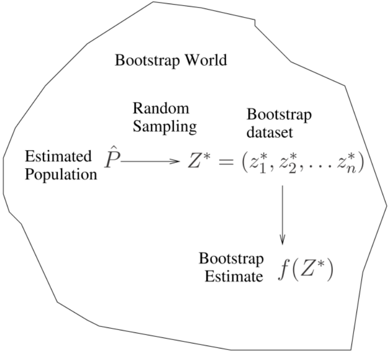
Bootstrap version
Predicted $SE(\hat{\alpha}) = 0.087$



Boxplots have similar spreads, meaning can both be used to estimate the variability of $\hat{\alpha}$

A general picture for the bootstrap





Summary of Bootstrap

- Start with data set of n points
- Sample n points with replacement to get data set Z^{*1}
- Use this to estimate whatever parameter we want $\hat{T}^{*1}$
- Repeat B times to get estimates $\hat{T}^{*1}, \cdots, \hat{T}^{*B}$
- Estimate standard error of our T estimate by

$$SE_B(\hat{T}) = \sqrt{\frac{1}{B-1}\sum_{r=1}^{B}\left(\hat{T}^{*r} - \frac{1}{B}\sum_{r'=1}^{B}\hat{T}^{*r'}\right)^2}$$

- Use for getting variance of a estimated quantity

Boostrap vs Cross-Validation

Bootstrap:

- Resamples with replacement
- Same number of data points per sample as original data set (n)
- Randomness from doing this B times
- Goal: establish empirical distribution functions for a widespread range of statistics

CV:

- Takes subset without replacement
- Subset, $< n$
- k-fold CV version: Randomness from sorting, but then subsets hit all points
- Goal: Measuring performance of a model