Ch 5.1.4-5: More Cross-Validation

Lecture 12 - CMSE 381

Prof. Elizabeth Munch

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

Fri, Oct 7, 2022

Announcements

Last time:

- k-fold CV

This lecture:

- More k-fold CV
- Bias-Variance Tradeoff
- CV for classification

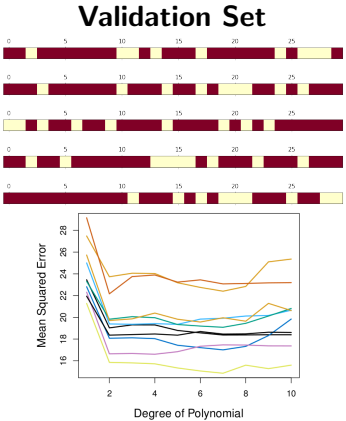
Announcements:

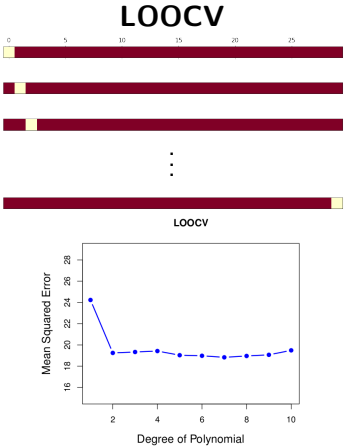
- Homework #4 is posted, Due Monday
-

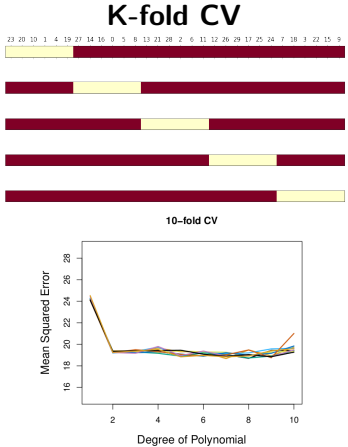
Section 1

k-fold CV

Approximations of Test Error





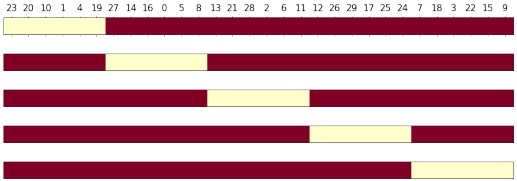


Definition of k-fold CV

- 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 $\mathrm{MSE}_i = \frac{1}{\ell} \sum_{(x_j, y_j) \in U_i} (y_j - \hat{y}_j)^2$

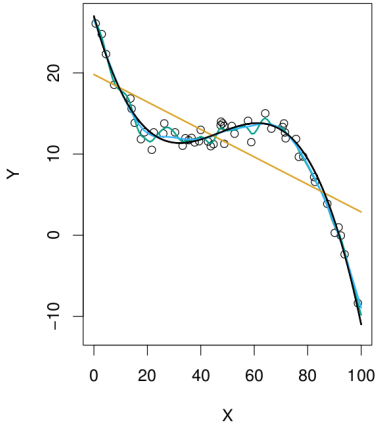
- Rinse and repeat

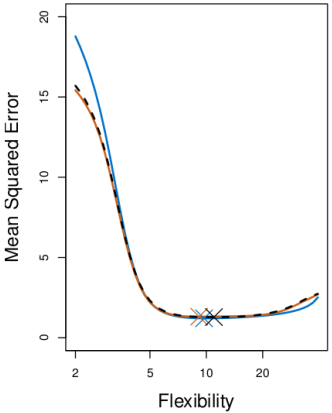


Return

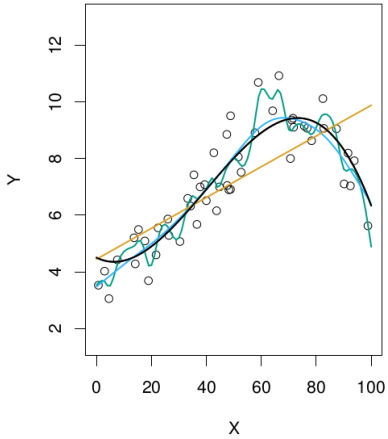
$$CV_{(k)} = \frac{1}{k} \sum_{i=1}^{k} \mathrm{MSE}_i$$

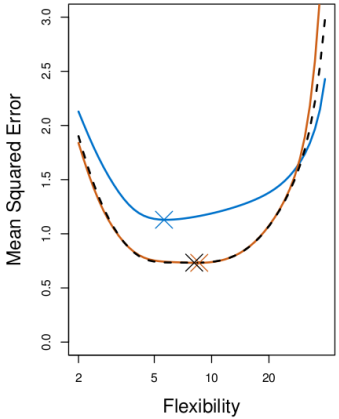
Comparison with simulated data: Ex 3



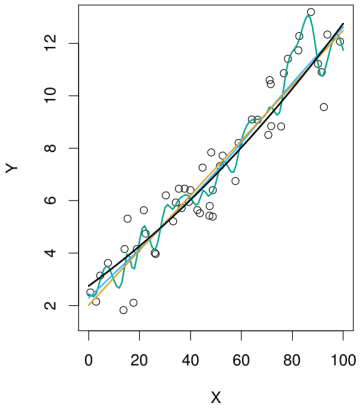


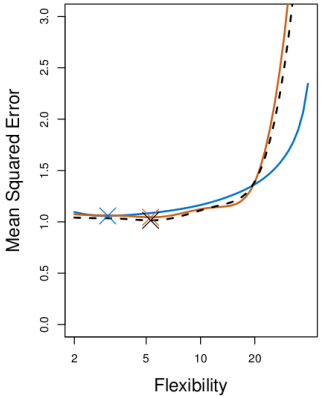
Comparison with simulated data: Ex 1





Comparison with simulated data: Ex 2





Takeaways from the examples

Bias-Variance Tradeoff: Bias

$$E(y_0 - \hat{f}(x_0))^2 = \text{Var}(\hat{f}(x_0)) + \left[\text{Bias}(\hat{f}(x_0))\right]^2 + \text{Var}(\varepsilon)$$

Bias-Variance Tradeoff: Variance

$$E(y_0 - \hat{f}(x_0))^2 = \text{Var}(\hat{f}(x_0)) + \left[\text{Bias}(\hat{f}(x_0))\right]^2 + \text{Var}(\varepsilon)$$

Coding - Build a plot for train/test scores vs flexibility

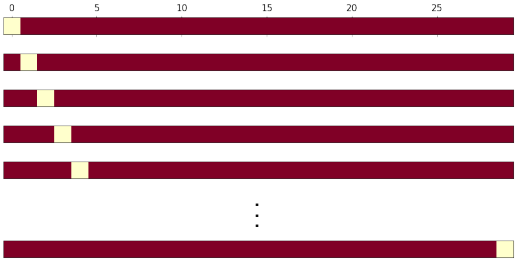
Section 2

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

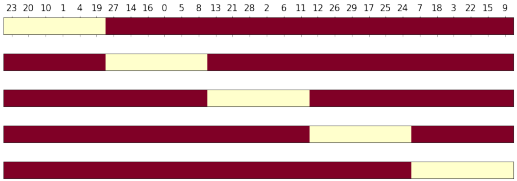
$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{Err}_i$$

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 $\text{Err}_i = \frac{1}{\ell} \sum_{(x_j, y_j) \in U_i} \text{I}(y_j \neq \hat{y}_j)$

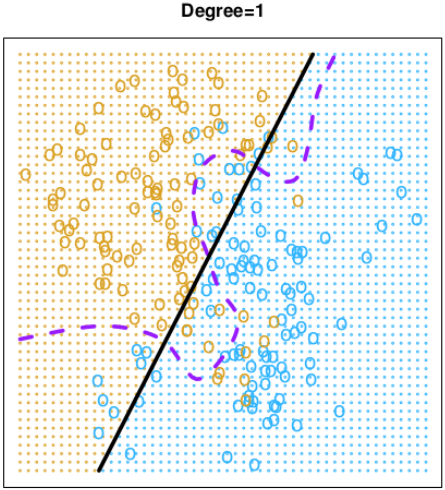
- Rinse and repeat



Return

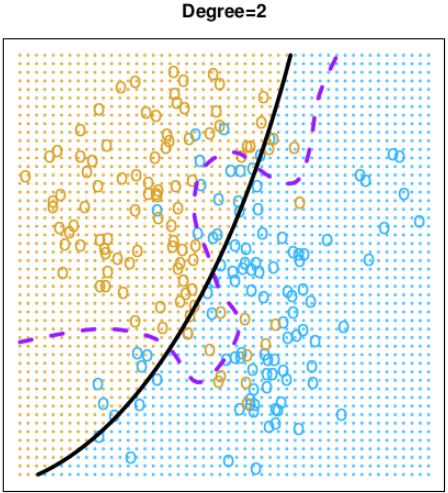
$$CV_{(k)} = \frac{1}{k} \sum_{i=1}^{k} \text{Err}_i$$

Example on simulated data: Linear



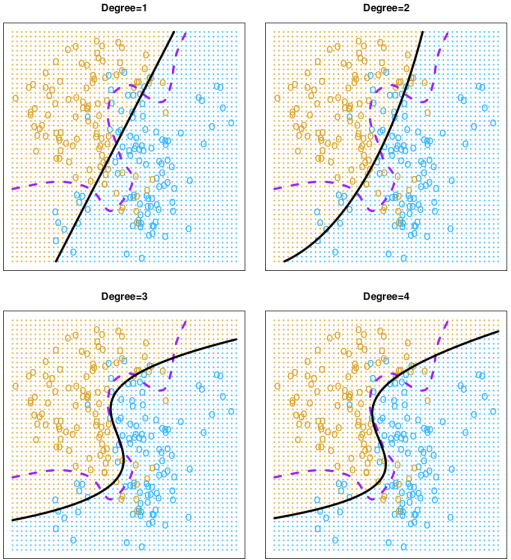
- Purple: Bayes decision boundary.
  - Error rate: 0.133
- Black: Logistic regression
  - $\log(p/(1-p)) = \beta_0 + \beta_1 X_1 + \beta_2 X_2$
  - Error rate: 0.201

Example on simulated data: Quadratic logistic regression



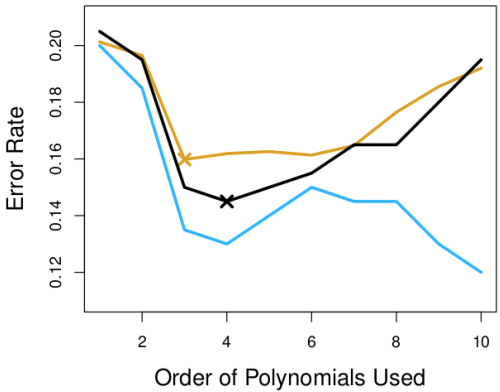
- Purple: Bayes decision boundary.
  - Error rate: 0.133
- Black: Logistic regression
  - $\log(p/(1-p)) = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_2 + \beta_4 X_2^2$
  - Error rate: 0.197

Example on simulated data: all the polynomials!



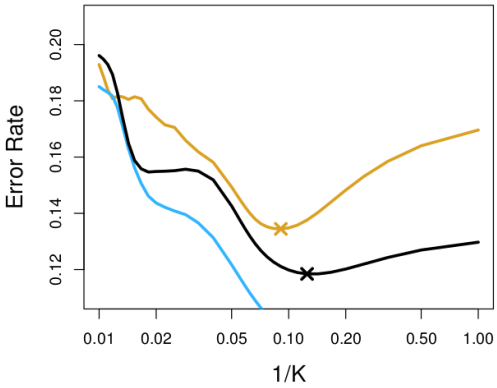
- Purple: Bayes decision boundary.
  - Error rate: 0.133
- Black: Logistic regression
  - Deg 1 Error rate: 0.201
  - Deg 2 Error rate: 0.197
  - Deg 3 Error rate: 0.160
  - Deg 4 Error rate: 0.162

Decide degree based on CV



- Test error (brown)
- Training error (blue)
- 10-fold CV error (black)

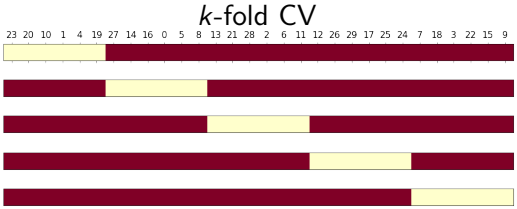
Similar game for KNN



- Test error (brown)
- Training error (blue)
- 10-fold CV error (black)

Coding - k-fold for Classification section

TL;DR



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

Use $k = 5$ or 10 usually

k-fold CV for classification

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

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

Next time

10	M	Oct 3	Leave one out CV	5.1.1, 5.1.2	
11	W	Oct 5	k-fold CV	5.1.3	
12	F	Oct 7	More k-fold CV, k-fold CV for classification	5.1.4-5	
13	M	Oct 10	Resampling methods: Bootstrap	5.2	HW #4 Due
14	W	Oct 12			
15	F	Oct 14	Subset selection	6.1	
16	M	Oct 17	Shrinkage: Ridge	6.2.1	HW #5 Due
17	W	Oct 19	Shrinkage: Lasso	6.2.2	
18	F	Oct 21	Dimension Reduction	6.3	
	M	Oct 24	No class - Fall break		
19	W	Oct 26	More dimension reduction; High dimensions	6.4	
20	F	Oct 28	Polynomial & Step Functions.	7.1,7.2	HW #6 Due
	M	Oct 31	Review		
	W	Nov 2	***Midterm #2***		