Resampling Methods

ddebarr@uw.edu

2017-02-02

“Oh sure, going in that direction will totally minimize the objective function“
-- Sarcastic Gradient Descent
-- John Urschel, Baltimore Ravens Offensive Lineman, MIT PhD Candidate (Math)

Course Outline

1. Introduction to Statistical Learning
2. Linear Regression
3. Classification
4. Resampling Methods
5. Linear Model Selection and Regularization
6. Moving Beyond Linearity
7. Tree-Based Methods
8. Support Vector Machines
9. Unsupervised Learning
10. Neural Networks and Genetic Algorithms

Agenda

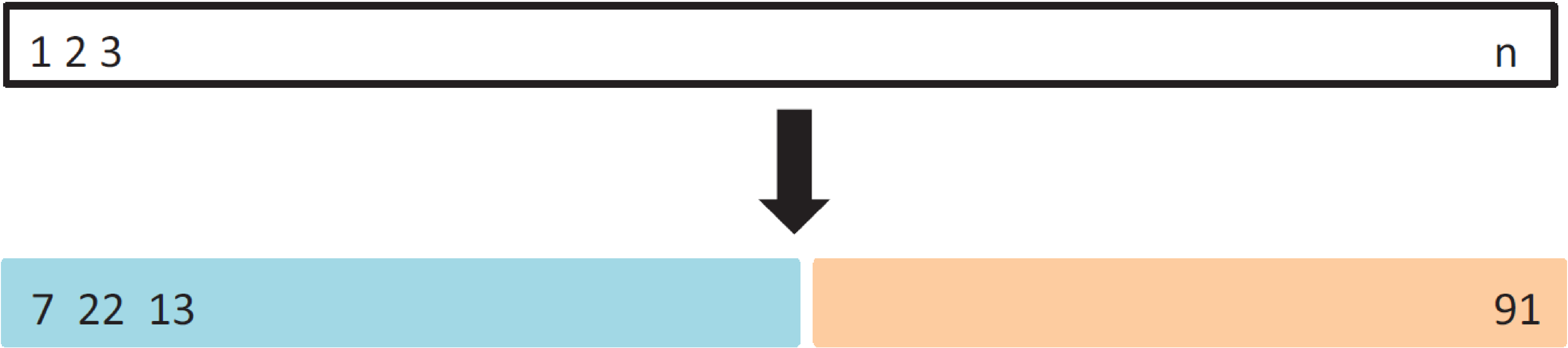
Discriminant Analysis
(from last week)

Resampling Methods

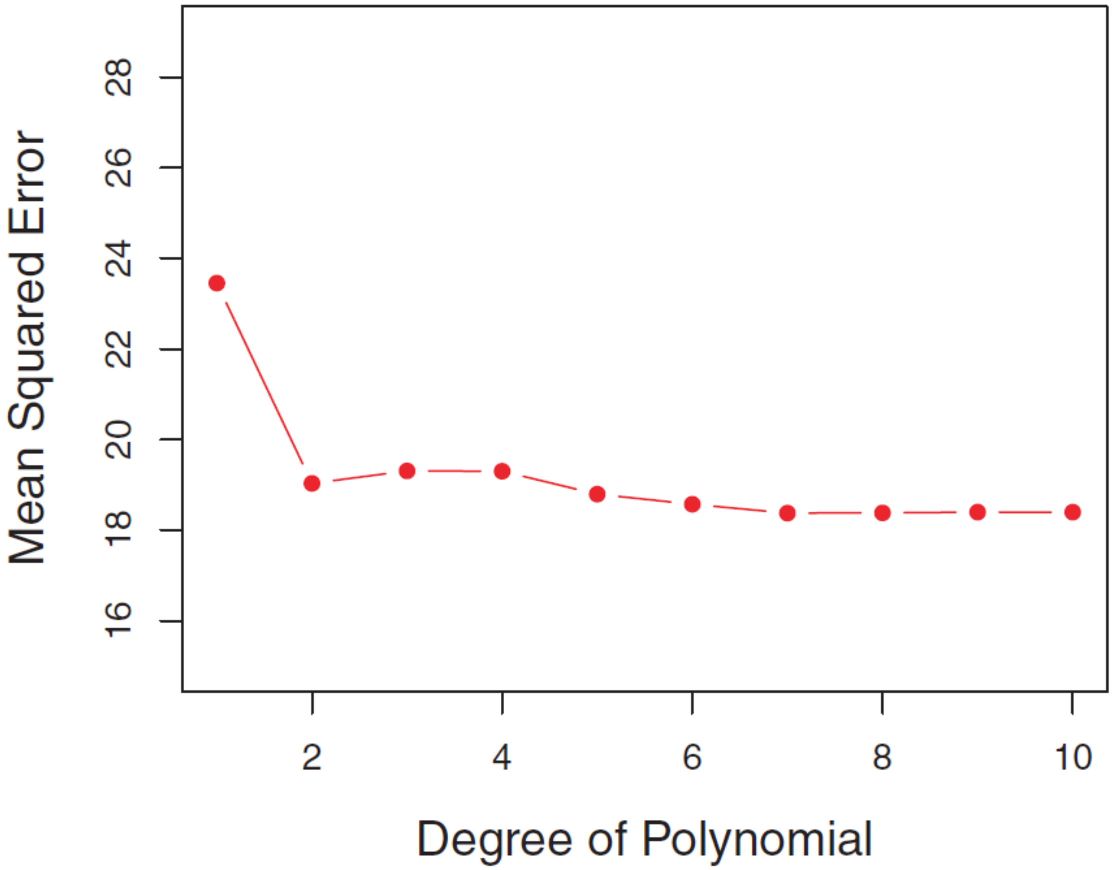
Hands-On Labs
(including caret)

Validation Set Approach

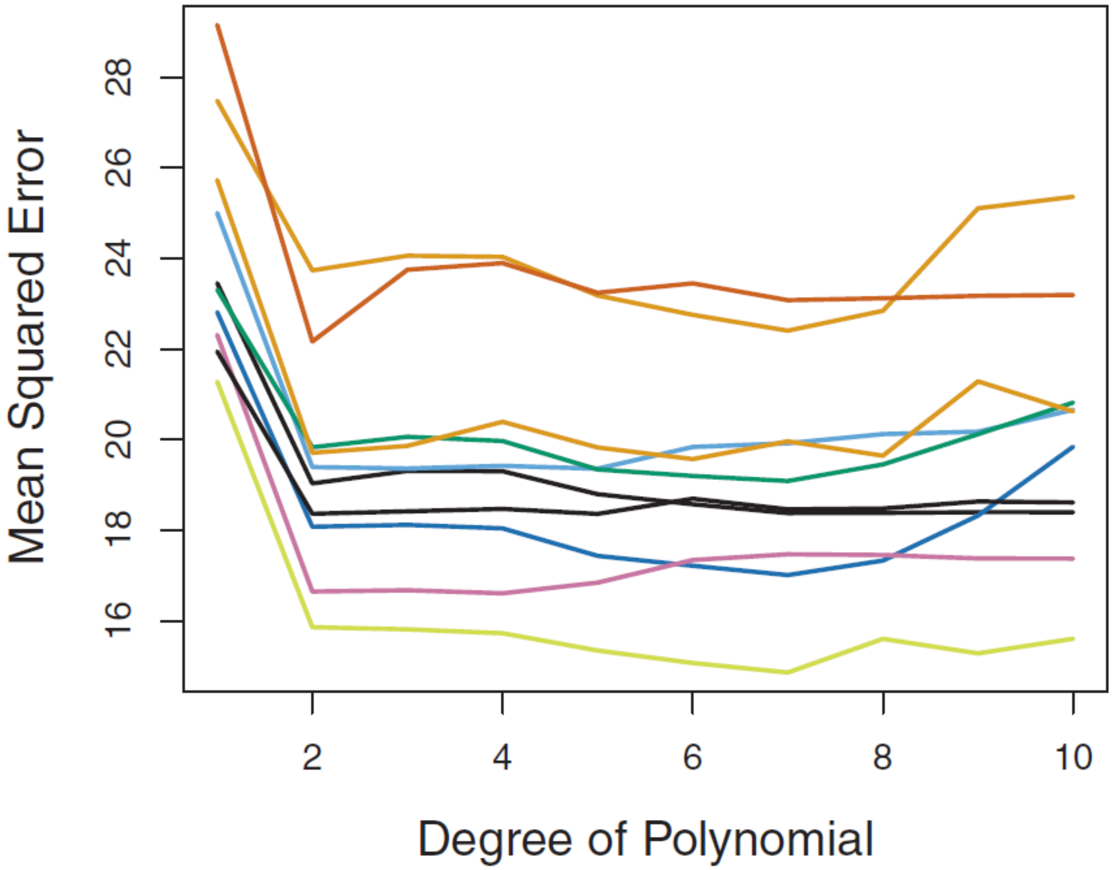
- Observations are randomly divided into training and validation sets
- 50/50 split can be used; but 80/20 may be more common
- Training set appears in blue; Validation set appears in beige



Repeated Validation Set Approach



One Train/Validation Set

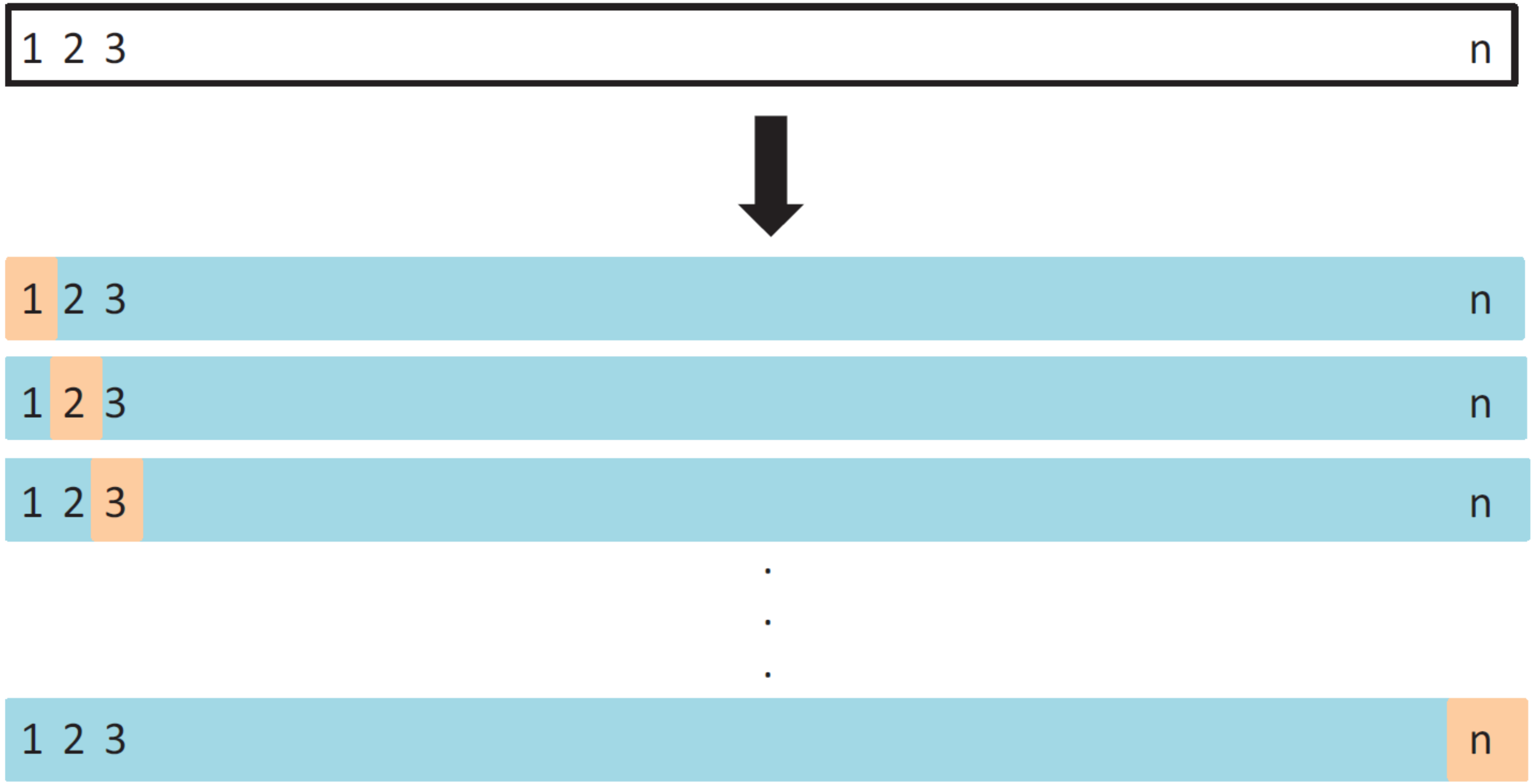


Ten Train/Validation Sets

MSE Reported for Validation Set

Leave One Out Cross Validation (LOOCV)

- Each observation takes its turn as the validation set
- ‘n’ models are constructed; one for each validation set



$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\mathrm{MSE}_i$$

LOOCV Short Cut for Linear Regression

- We can use the leverage statistic to turn the error estimates for the training set into a LOOCV estimate
- For multiple regression, we use the entries of the diagonal of the "projection" matrix (sometimes called the hat matrix, because it is used to derive $\hat{y}$)

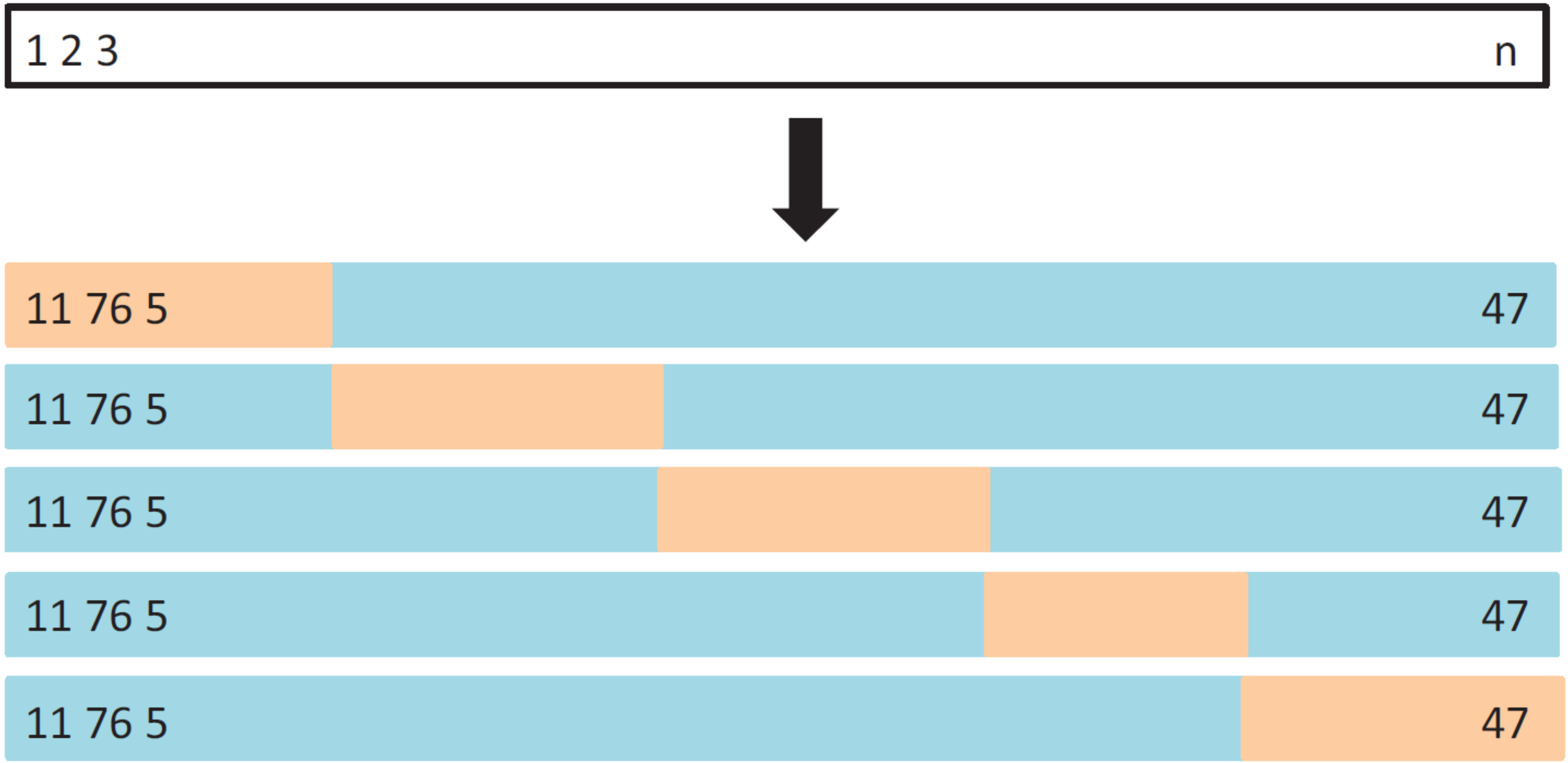
$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - h_i}\right)^2$$

$$h_i = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{i'=1}^{n}(x_{i'} - \bar{x})^2}$$

$$\mathbf{H} = \mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T$$

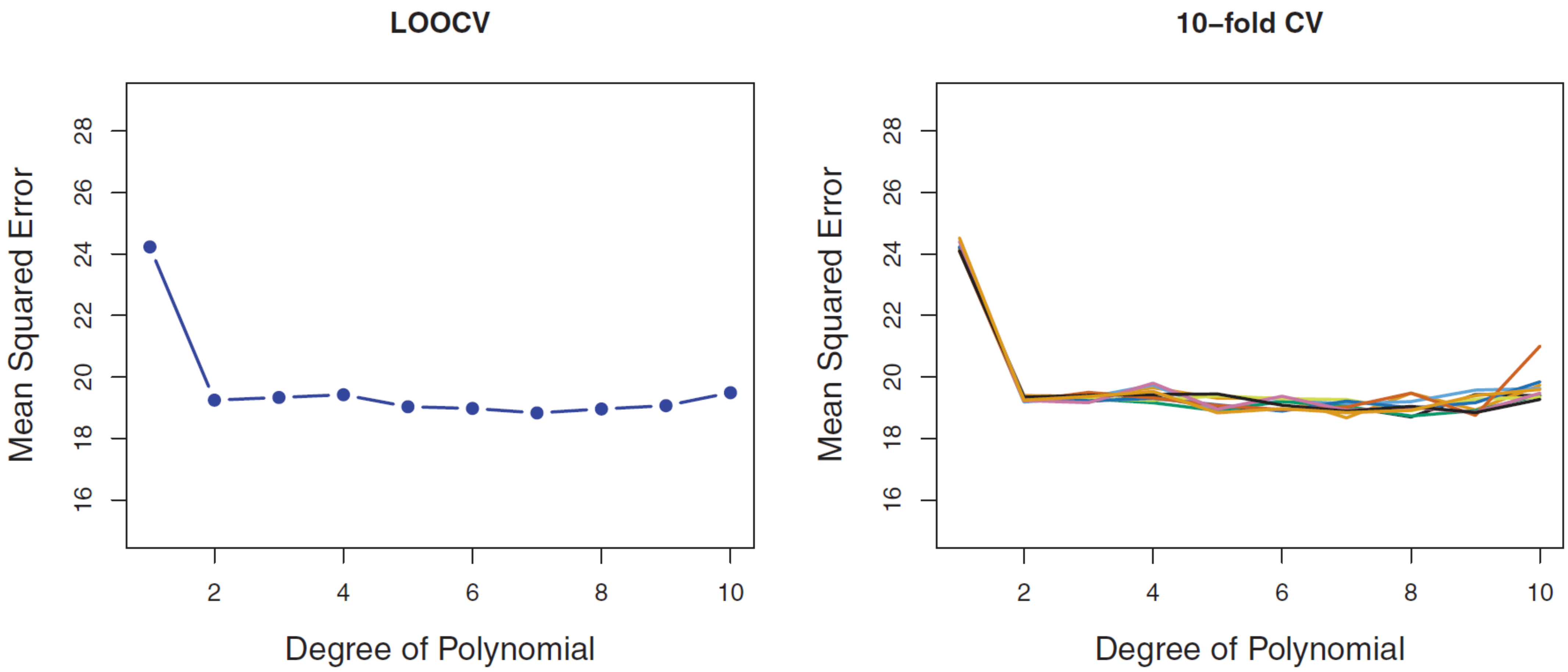
K-Fold Cross Validation

- Training set appears in blue; Validation set appears in beige
- 'k' models are constructed; one for each validation set

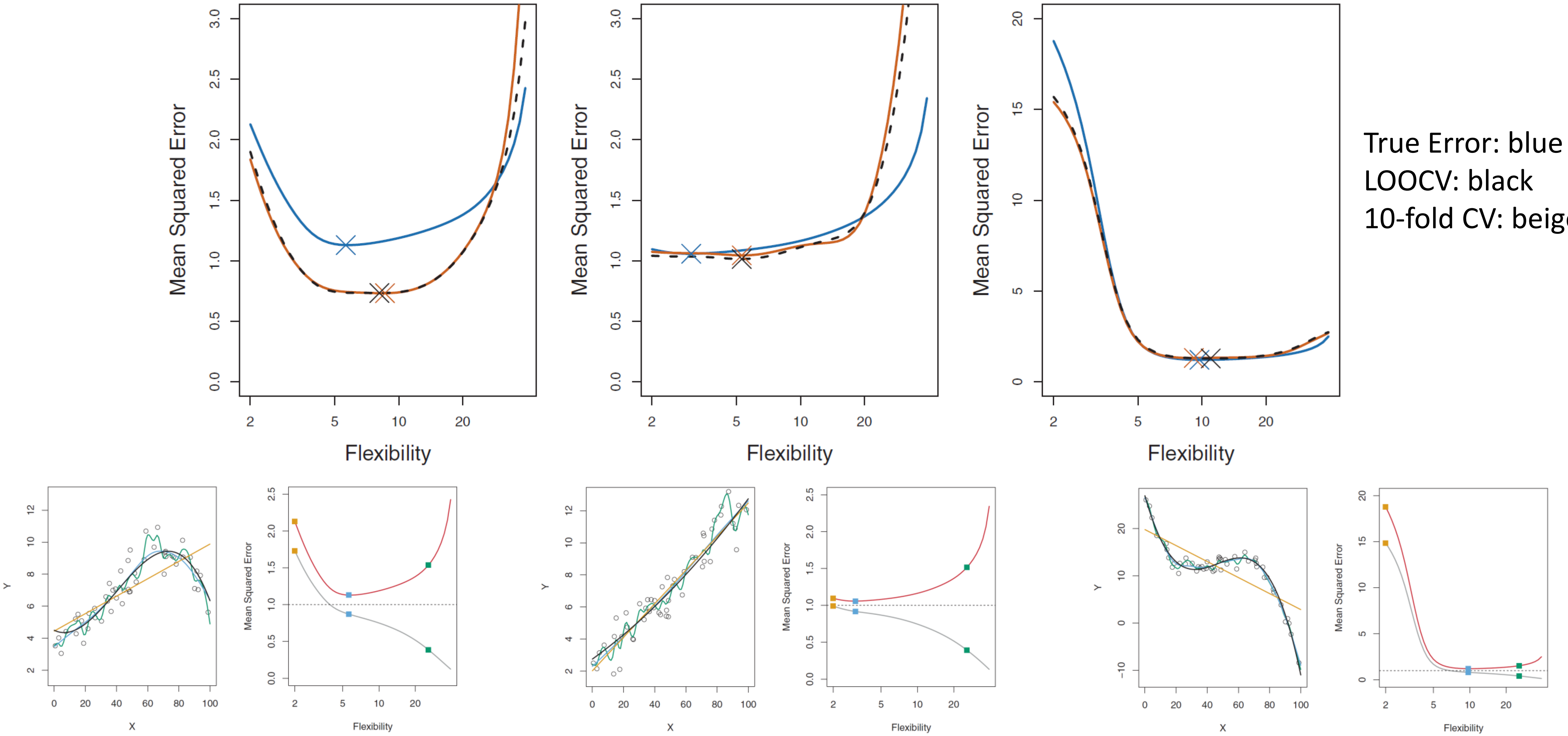


$$\mathrm{CV}_{(k)} = \frac{1}{k}\sum_{i=1}^{k}\mathrm{MSE}_i$$

LOOCV versus Repeated 10-fold CV



Cross Validation Applied to Simulated Data



True Error: blue
LOOCV: black
10-fold CV: beige

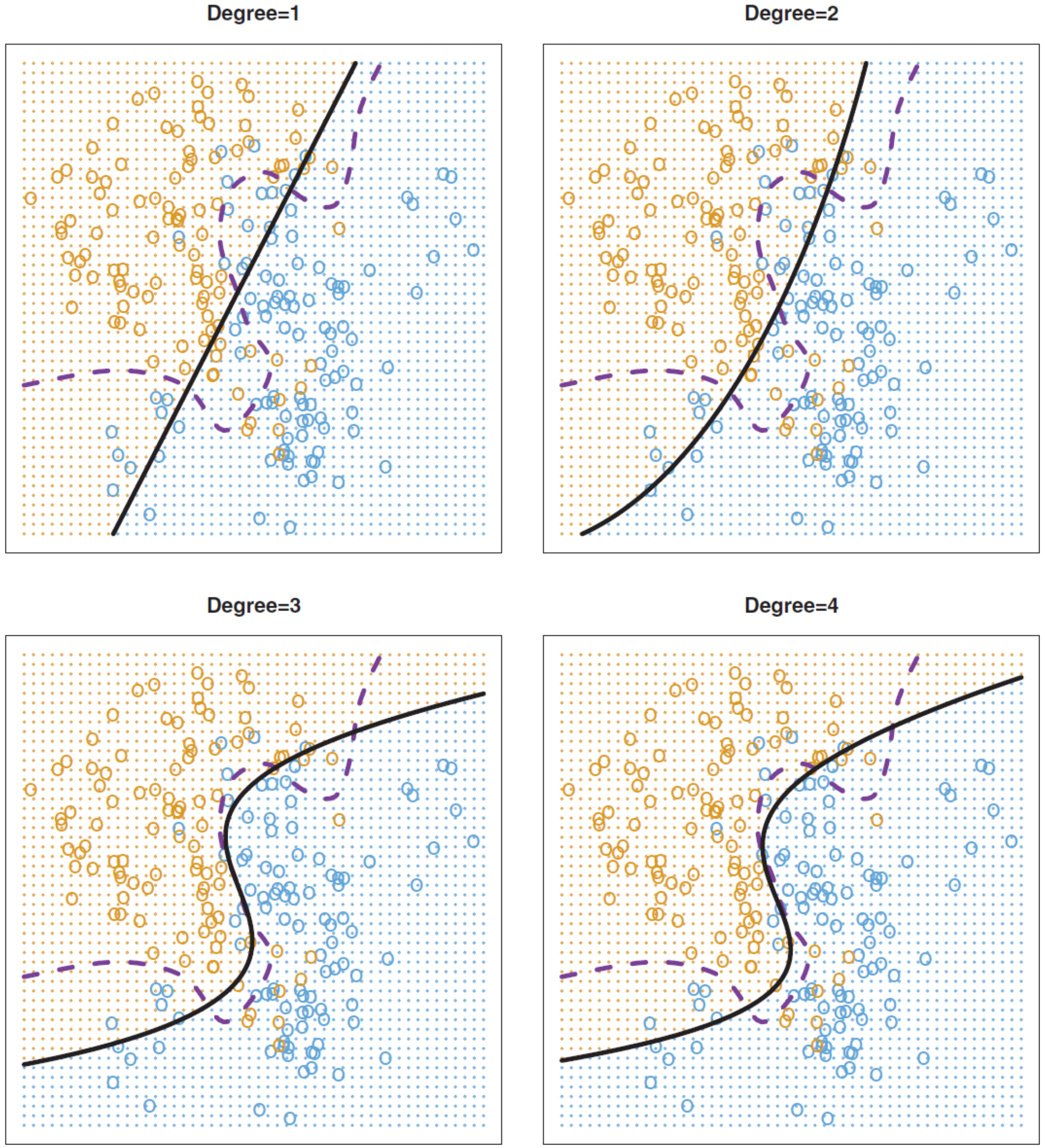
Cross Validation for Classification Problems

- Analogous definitions apply
- Example: LOOCV

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

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_2 + \beta_4 X_2^2$$

Example Quadratic Logistic Regression Model



Error = 0.201

Error = 0.197

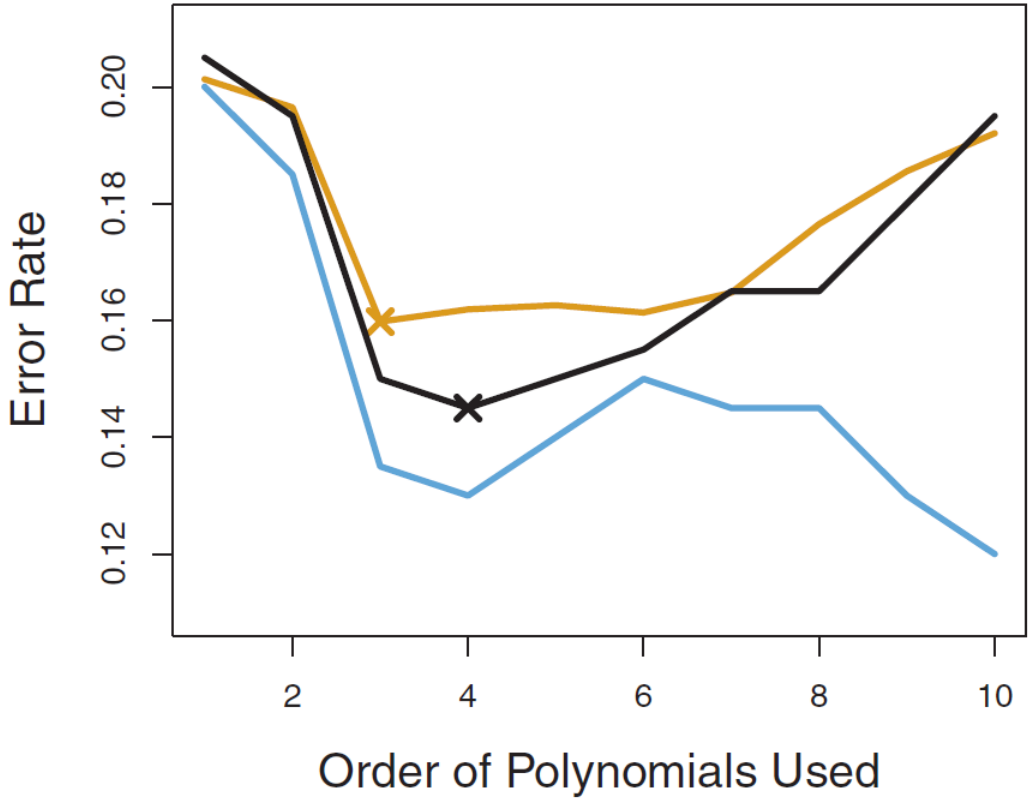
Bayes Error (dashed line) = 0.133

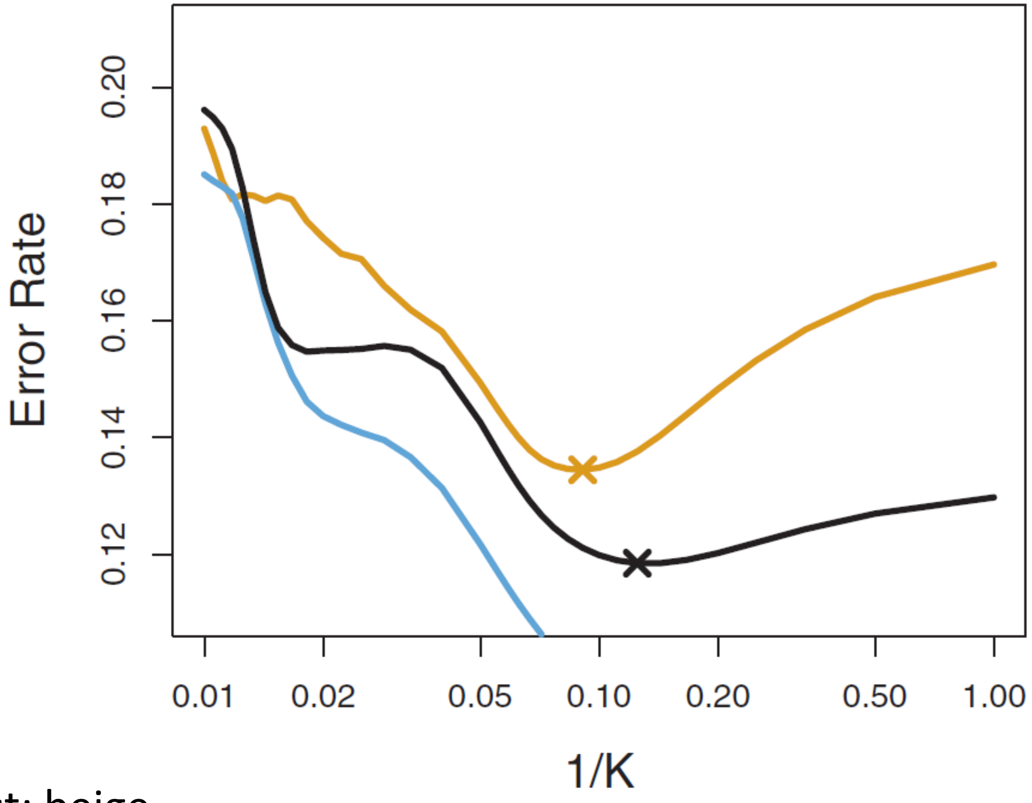
Error = 0.160

Error = 0.161

Test versus Cross Validation versus Train Error

Error rates for simulated data from previous slide





Test: beige
CV: black
Train: blue

Logistic regression

K Nearest Neighbor

Minimizing the Variance of a Sum of Possibly Correlated Variables

$$Var\left(\alpha X+(1-\alpha)Y\right)=Var\left(\alpha X\right)+Var\left((1-\alpha)Y\right)+2Cov\left(\alpha X,(1-\alpha)Y\right)$$

$$=\alpha^2 Var\left(X\right)+(1-\alpha)^2 Var\left(Y\right)+2\alpha(1-\alpha)Cov\left(X,Y\right)$$

… so the gradient is …

$$\frac{\partial}{\partial\alpha}\left(\alpha^2 Var\left(X\right)+(1-\alpha)^2 Var\left(Y\right)+2\alpha(1-\alpha)Cov\left(X,Y\right)\right)=2\alpha Var\left(X\right)+2(1-\alpha)(-1)Var\left(Y\right)+(2-4\alpha)Cov\left(X,Y\right)$$

$$=2\alpha Var\left(X\right)-(2-2\alpha)Var\left(Y\right)+(2-4\alpha)Cov\left(X,Y\right)$$

… solving for α when gradient equals 0 …

$$2\alpha Var\left(X\right)-(2-2\alpha)Var\left(Y\right)+(2-4\alpha)Cov\left(X,Y\right)=0$$

$$2\alpha Var\left(X\right)-2Var\left(Y\right)+2\alpha Var\left(Y\right)+2Cov\left(X,Y\right)-4\alpha Cov\left(X,Y\right)=0$$

$$2\alpha Var\left(X\right)+2\alpha Var\left(Y\right)-4\alpha Cov\left(X,Y\right)=2Var\left(Y\right)-2Cov\left(X,Y\right)$$

$$\alpha Var\left(X\right)+\alpha Var\left(Y\right)-2\alpha Cov\left(X,Y\right)=Var\left(Y\right)-Cov\left(X,Y\right)$$

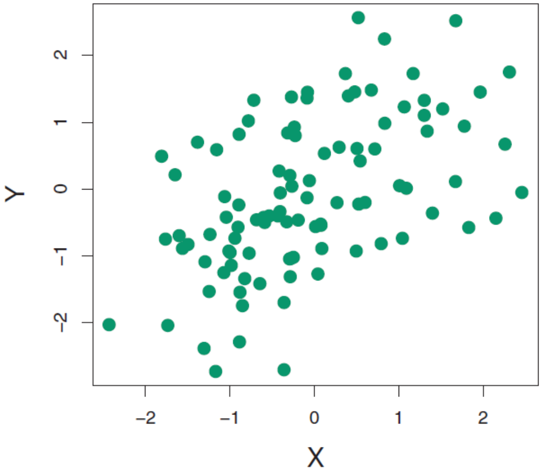
$$\alpha\left(Var\left(X\right)+Var\left(Y\right)-2Cov\left(X,Y\right)\right)=Var\left(Y\right)-Cov\left(X,Y\right)$$

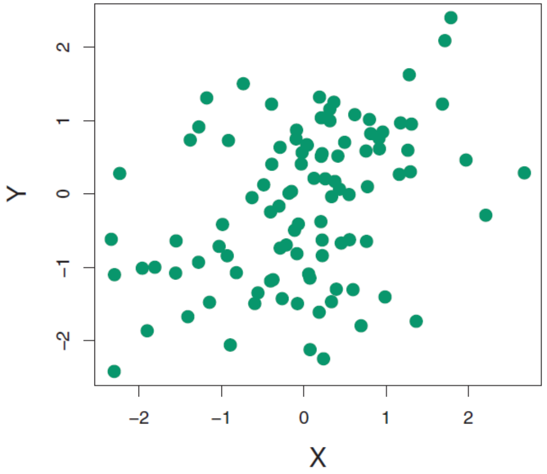
$$\alpha=\frac{Var\left(Y\right)-Cov\left(X,Y\right)}{Var\left(X\right)+Var\left(Y\right)-2Cov\left(X,Y\right)}$$

Simulated Data Sets

α is the proportion of money to be invested in X to minimize the variance of the return (risk)

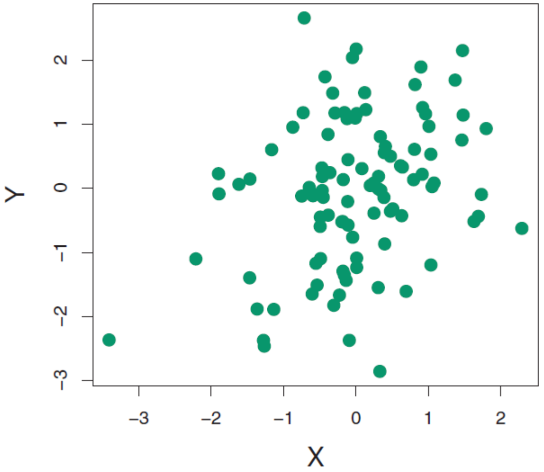
$\alpha = 0.576$

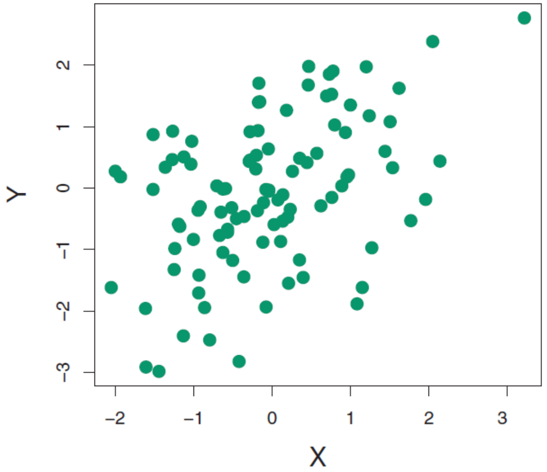




$\alpha = 0.532$

$\alpha = 0.657$





$\alpha = 0.651$

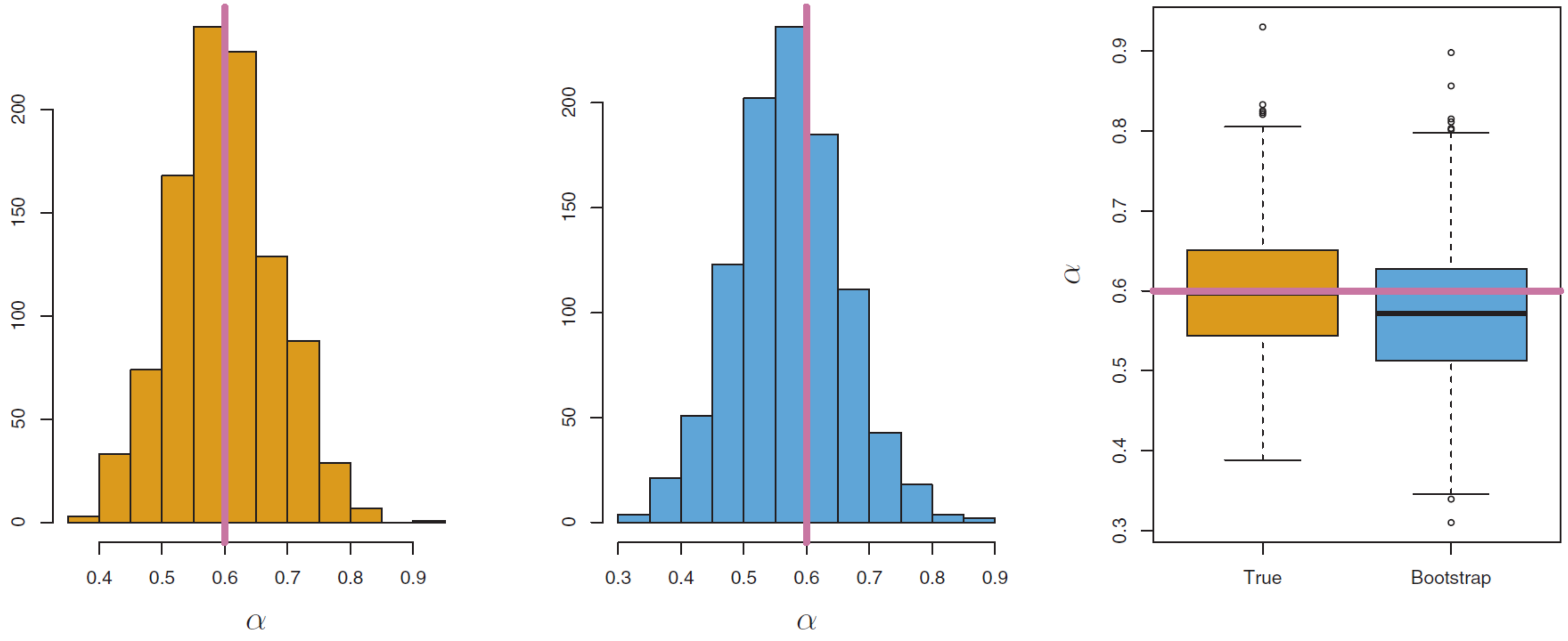
Mean and Standard Deviation of α Estimates for 1000 Samples of 100 Simulated Returns

$$\bar{\alpha} = \frac{1}{1,000} \sum_{r=1}^{1,000} \hat{\alpha}_r = 0.5996$$

$$\sqrt{\frac{1}{1,000 - 1} \sum_{r=1}^{1,000} (\hat{\alpha}_r - \bar{\alpha})^2} = 0.083$$

Unfortunately, they did *not* share the parameters used to generate the samples. ☹
I say we grab torches and pitch forks, then head over to the statistics department. Who's with me? ☺
Maybe they're just trying to prep us: we can't handle the truth [we won't know the truth for real data].

Simulated Values versus Bootstrap Values



Simulated Data
(based on True parameters)

Bootstrap Samples

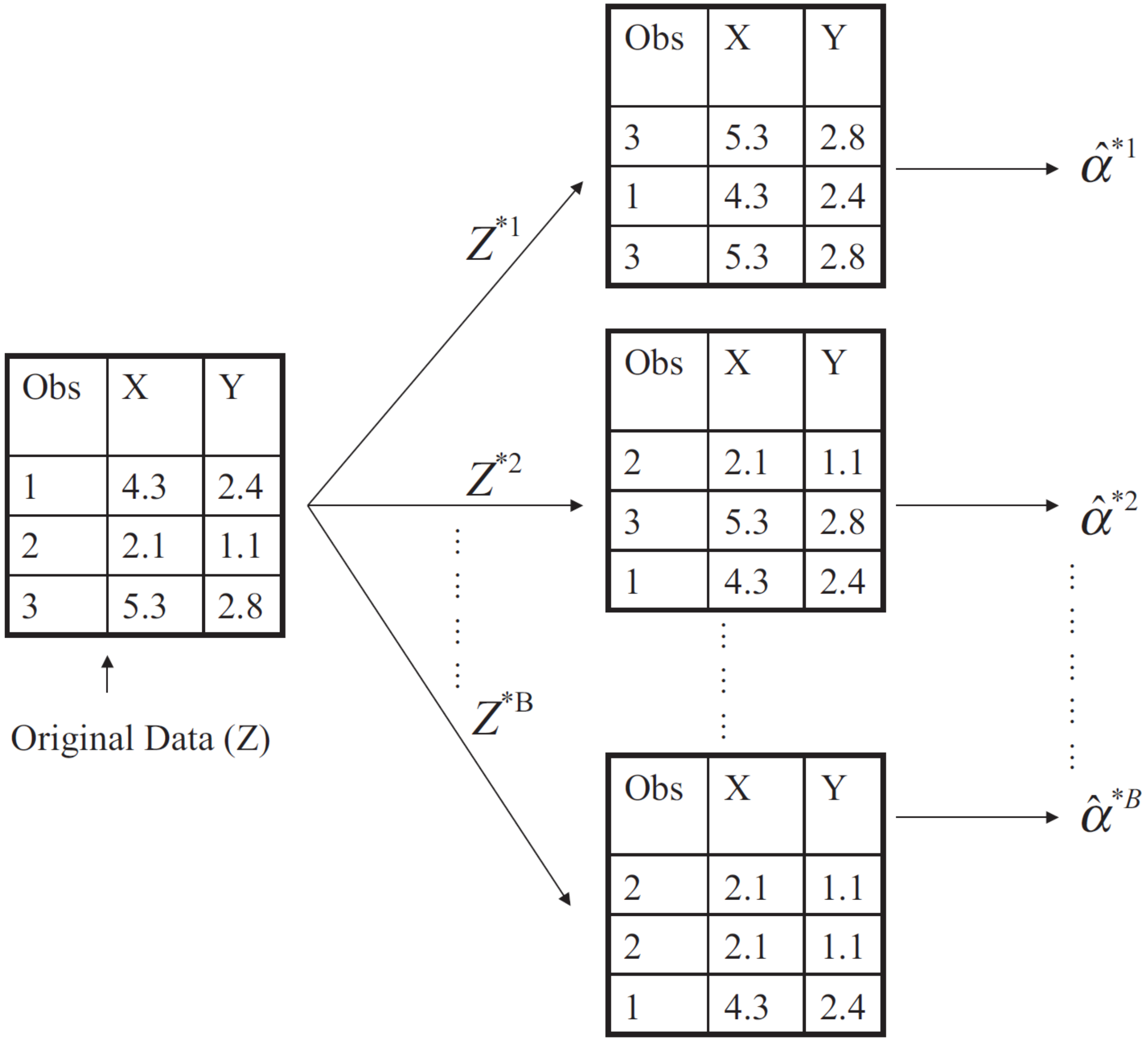
The Bootstrap: Quantifying Uncertainty via Standard Error

Notice that the “Standard Error” is simply the standard deviation of the alpha values from the bootstrap samples!

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

Bootstrap Sample

A bootstrap sample of a data set of ‘n’ observations is created by drawing ‘n’ random samples from the data set *with* replacement



Yet Another Example

How do we compute a 95% confidence interval for correlation?

One way …

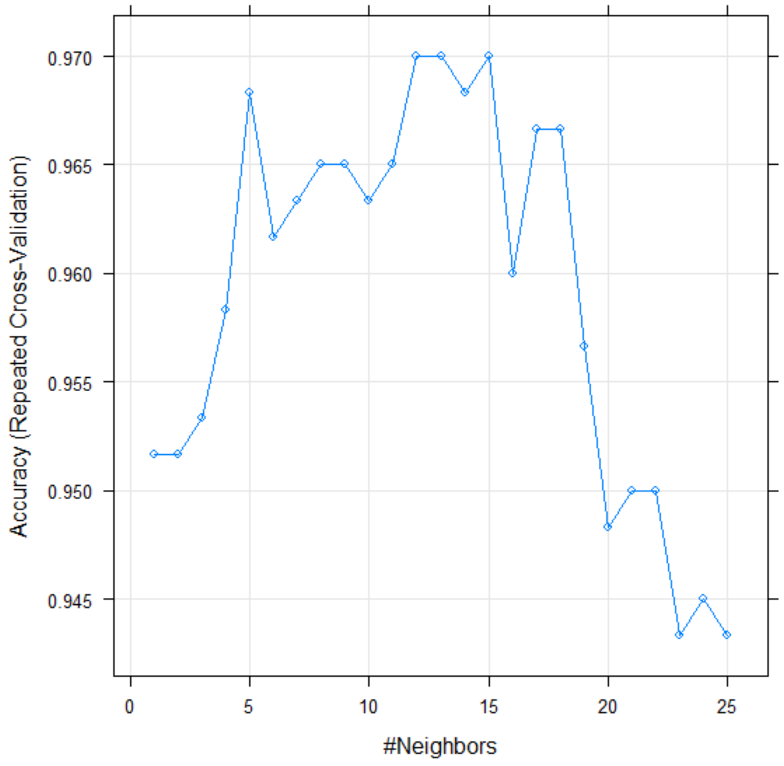
```
library(MASS)
X = mvrnorm(1000, mu = c(0, 0), Sigma = matrix(c(9, -4.5, -4.5, 4), nrow=2))

Pearson.Correlation.Confidence.Interval = function(vector1, vector2, confidence = 0.95) {
   z = qnorm(1 - (1 - confidence) / 2)
   n = length(vector1)
   r = cov(vector1, vector2) / (sd(vector1) * sd(vector2))
   return(tanh(atanh(r) + z * c(-1, 0, 1) * sqrt(1 / (n - 3))))
}
```

Is there an easier way? [maybe you need to understand uncertainty for some business metric]

```
library(boot)
Pearson.Correlation = function(X, index) { return(cov(X[index,1], X[index,2]) / (sd(X[index,1]) * sd(X[index,2]))) }
boot.ci(boot(X, statistic = Pearson.Correlation, R = 10000), conf = 0.95, type="bca")
```

Repeated 5 Fold Cross Validation via CARET

```
set.seed(2^17-1)
library(caret)
library(class)
input = iris    # flower classification
summary(input)
indices = tapply(1:nrow(input), input[,5], sample)
trn = rbind(input[indices$setosa[1:40],],
            input[indices$versicolor[1:40],],
            input[indices$virginica[1:40],])
tst = rbind(input[indices$setosa[41:50],],
            input[indices$versicolor[41:50],],
            input[indices$virginica[41:50],])
selection = train(trn[,1:4], trn[,5], method = "knn", metric = "Accuracy", maximize = T,
                  trControl = trainControl(method = "repeatedcv", number = 5, repeats = 5),
                  tuneGrid = data.frame(k = 1:25))
table(tst[,5], knn(trn[,1:4], tst[,1:4], trn[,5], k = selection$bestTune))
plot(selection)
```



Agenda

Discriminant Analysis
(from last week)

Resampling Methods

Hands-On Labs
(including caret)