Validation

CS 534: Machine Learning

Slides adapted from Lee Cooper and Ryan Tibshirani

Review: Bias & Variance Tradeoff

Bias, Variance, and Model Complexity

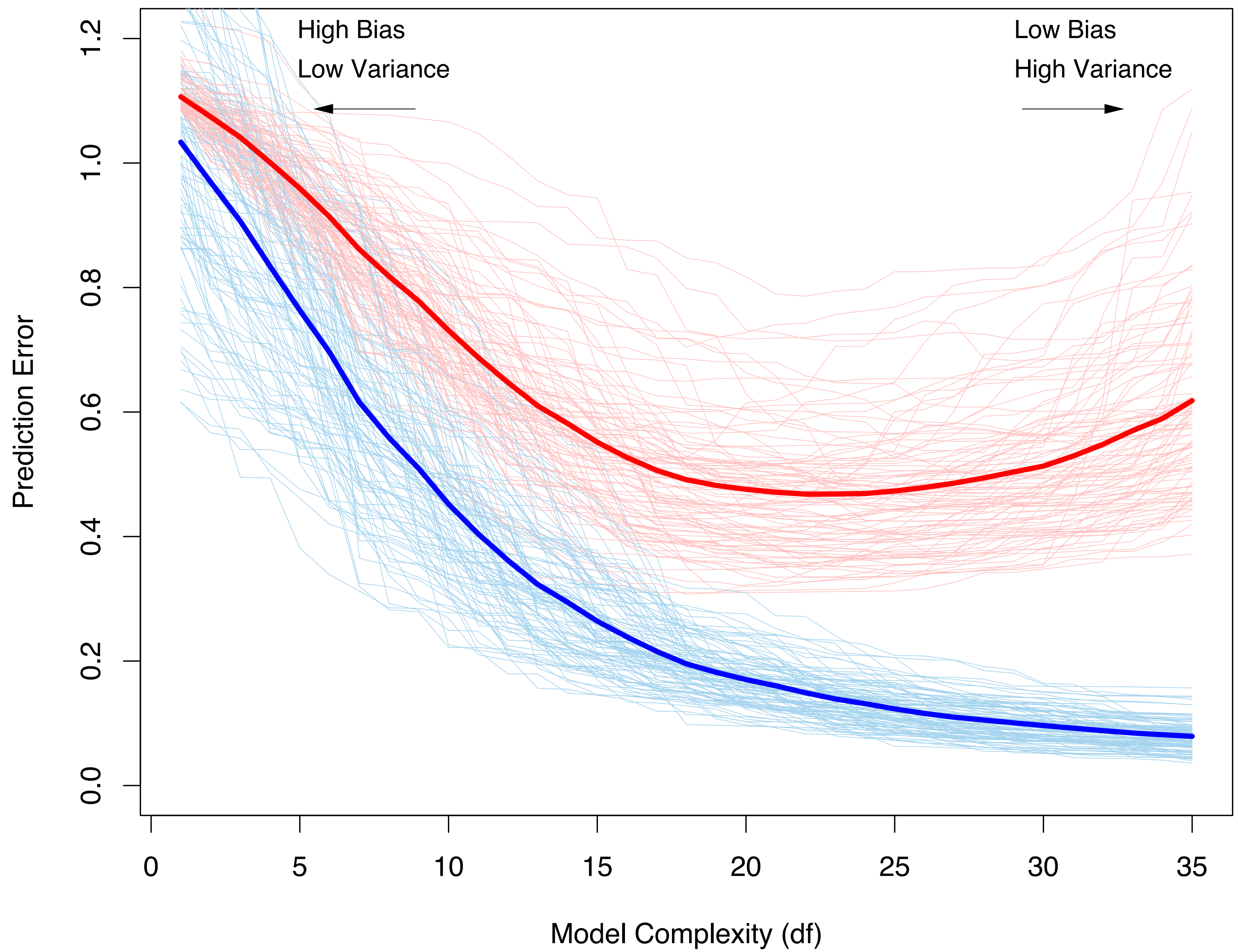


Figure 7.1 (Hastie et al.)

Bais-Variance Tradeoff: Key in ML

- Choice of hypothesis class introduces learning bias
  - More complex class —> less bias
  - More complex class —> more variance

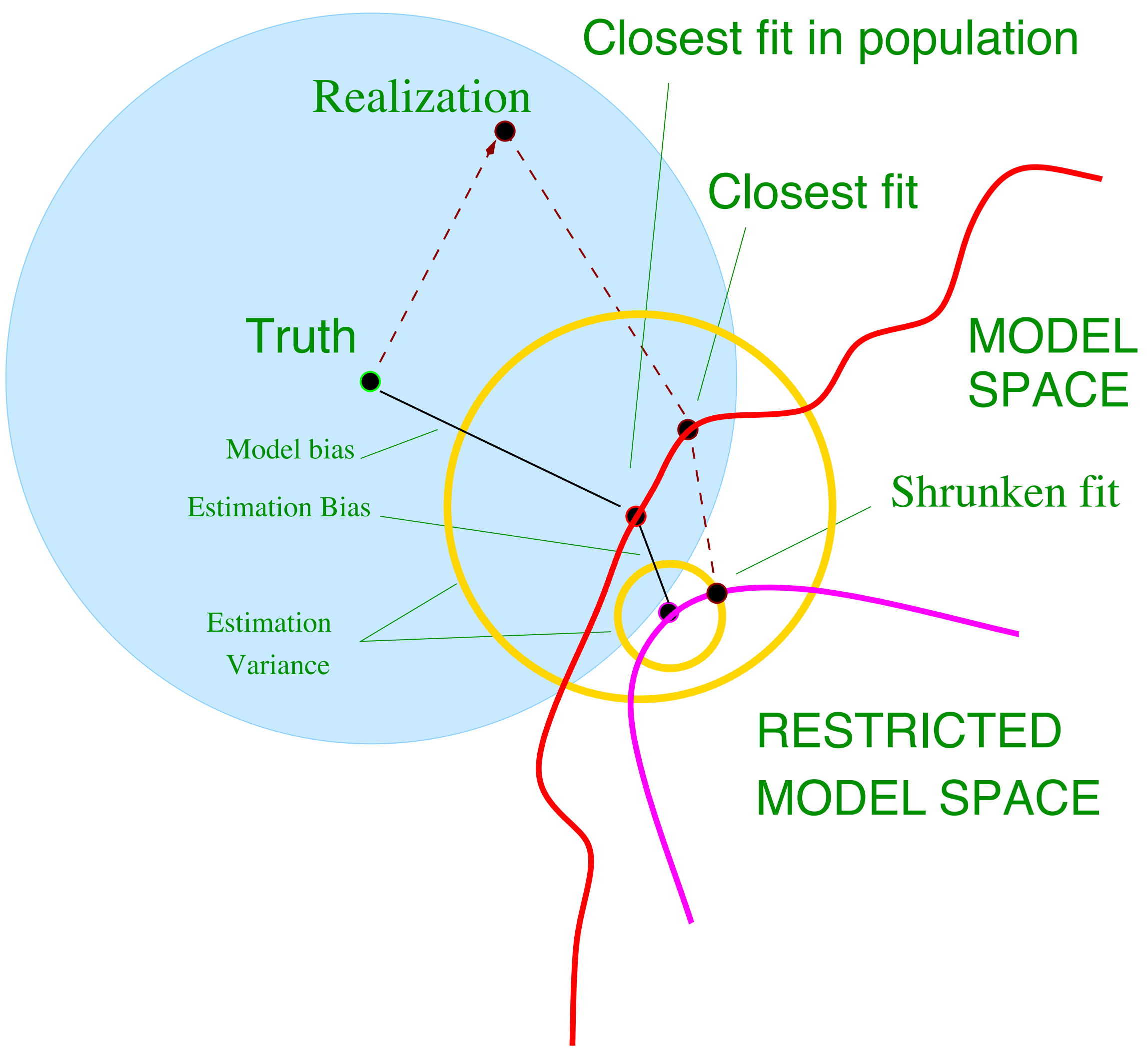


Figure 7.2 (Hastie et al.)

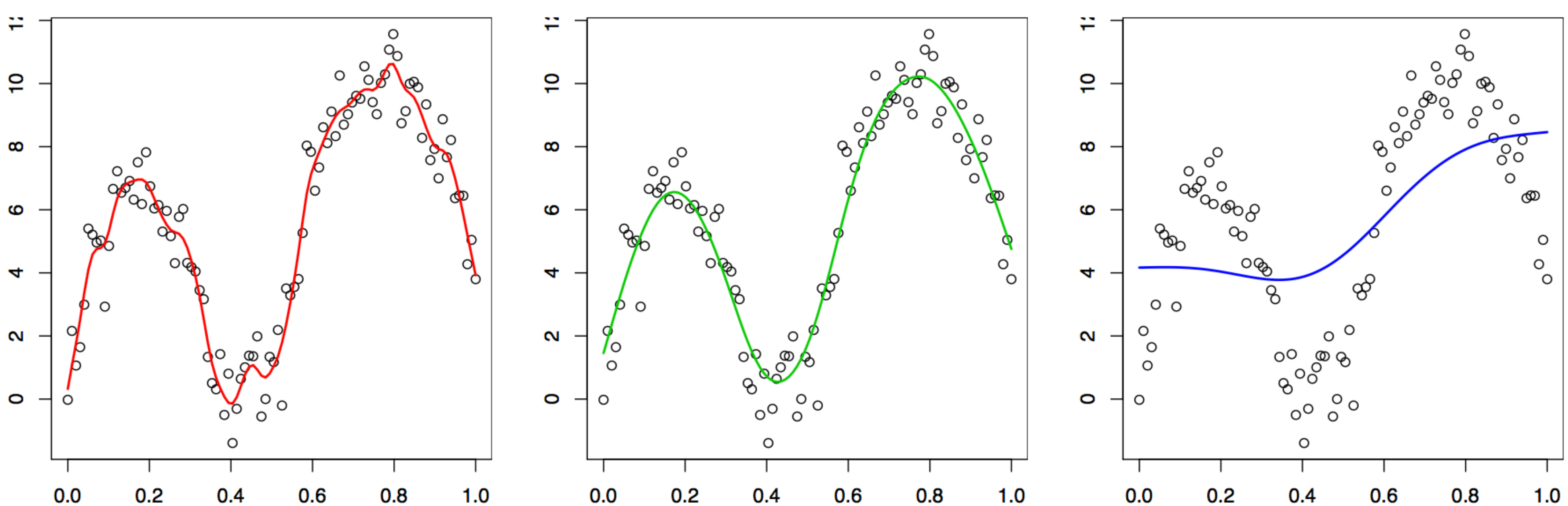
Fundamental Questions

- Model selection: How to compare performance of multiple models to choose the best (identify the best parameters or methods)?
- Model Assessment: What is the performance of the model on data that it has not seen yet?

Model Selection

Example: Smoothing Splines

$$\hat{f} = \text{argmin}_f \sum_i (y_i - f(\mathbf{x}_i))^2 + \lambda \int (f''(\mathbf{x}))^2 dx$$



λ too small　　λ just right　　λ too big

How to choose the tuning parameter?

Model Setup

- Suppose we observe some data (x_i, y_i), i = 1, …, n
- Prediction model $\hat{f}(\mathbf{X})$ that has been estimated from a training set $\mathcal{T}$
- Expected prediction error (EPE)

$$\begin{aligned}\mathrm{Err} &= E[L(Y, \hat{f}(\mathbf{X}))] \\ &= E[E[L(Y, \hat{f}(\mathbf{X}))|\mathcal{T}]] \\ &= E[\mathrm{Err}_{\mathcal{T}}]\end{aligned}$$

Training & Test Error

- Training error is average loss over the training sample

$$\text{TrainErr} = \frac{1}{N} \sum_i L(y_i, \hat{f}(\mathbf{x}_i))$$

- Test error is average loss over data that was not used to build our estimator

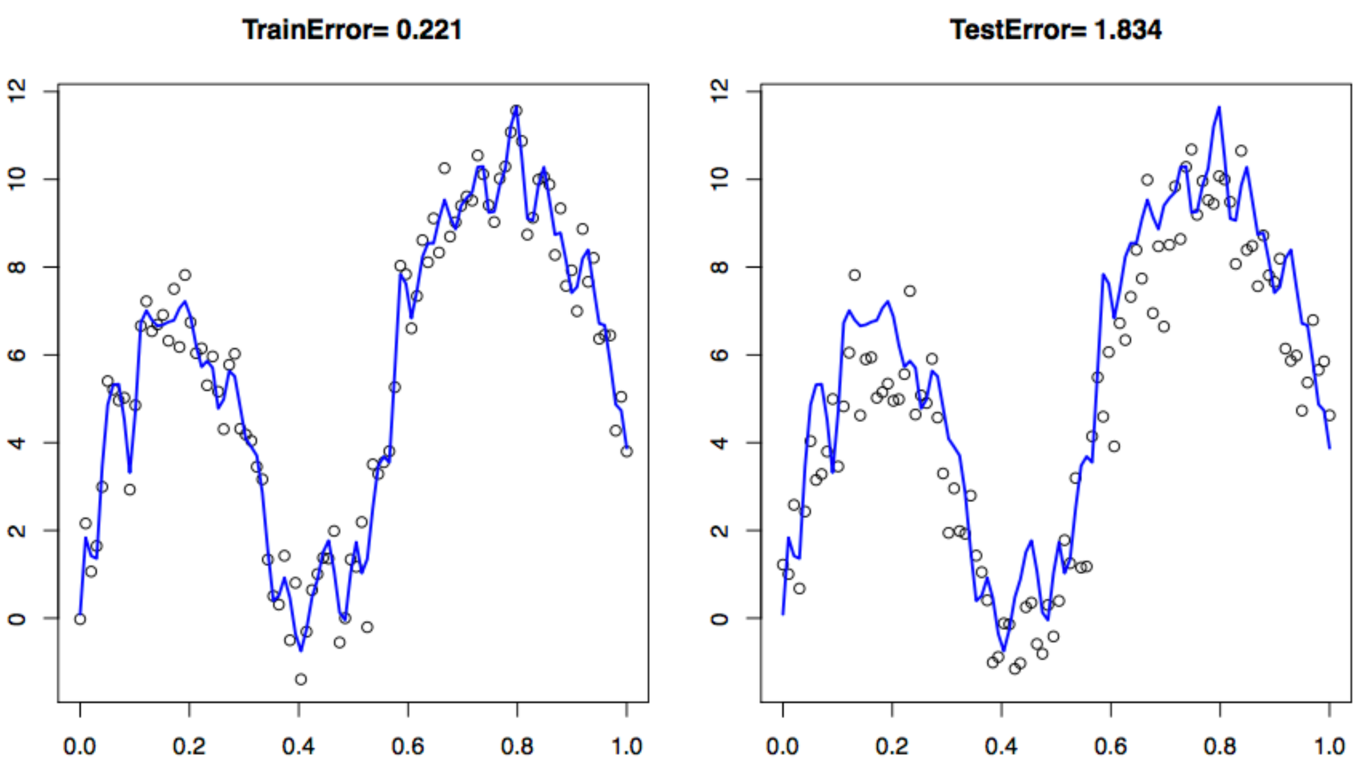
$$\text{TestErr} = \frac{1}{M} \sum_i L(y'_i, \hat{f}(\mathbf{x}'_i))$$

- Test error is estimate for EPE

Training Only?

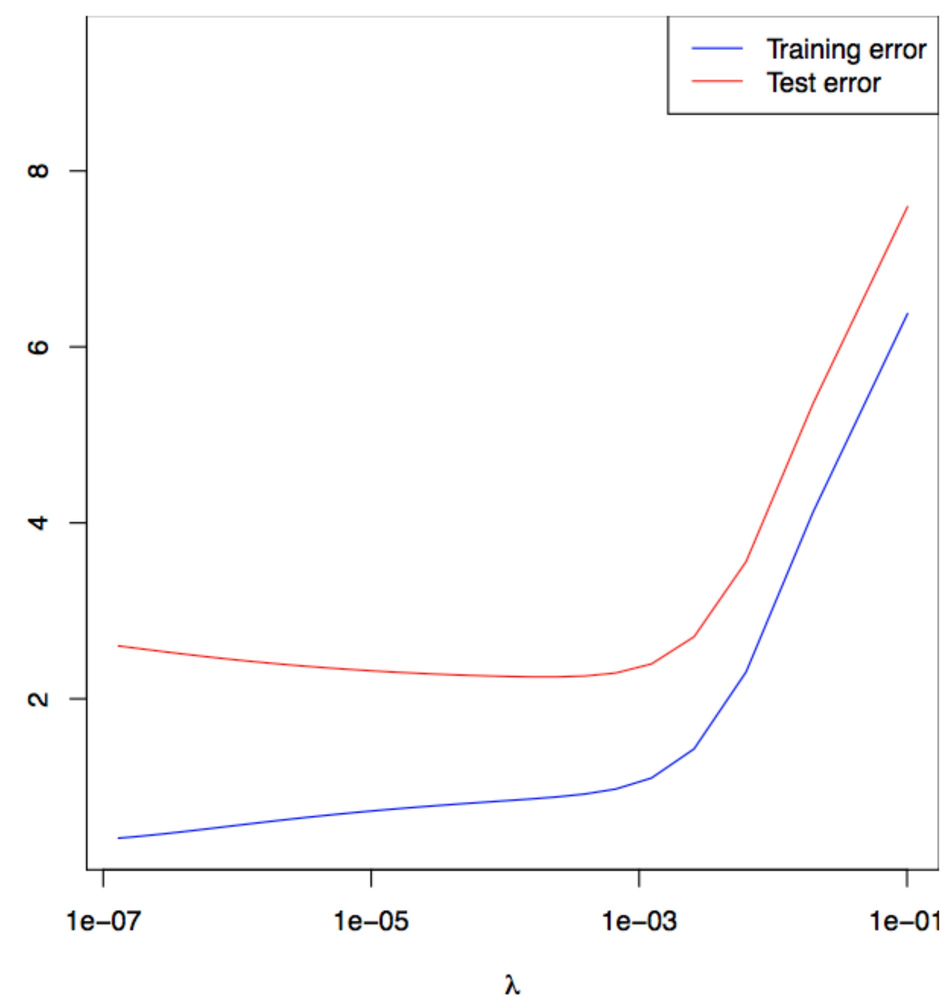
- What if we don't have test data? Should we use only training error?
- It seems like training and test error shouldn't be too different…
- Estimator adapts to the training data and thus will have an overly optimistic estimate of the generalization error!

Example: Smoothing Splines



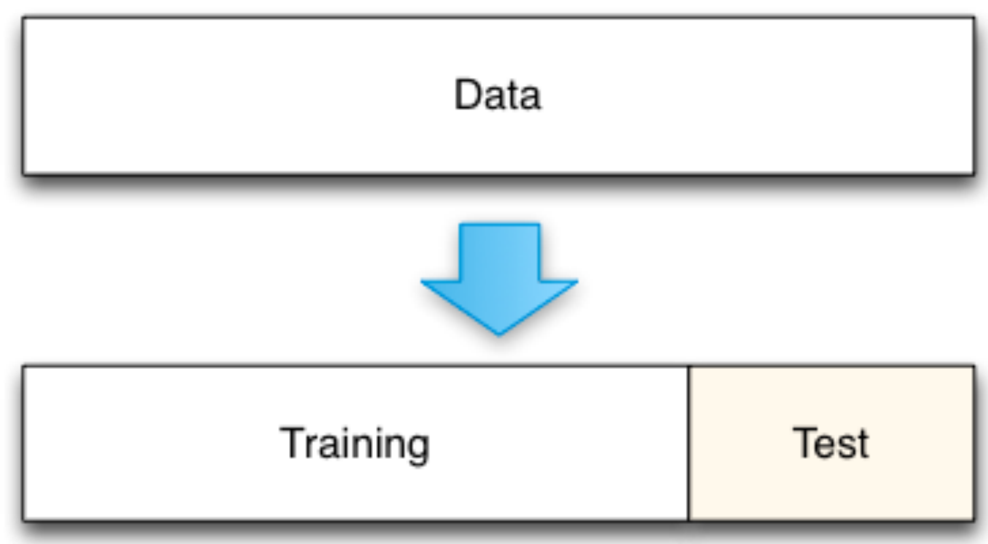
Small value of tuning parameter

Curves over 100 simulations for different parameters



Validation / Holdout Set Method

- Split data into two groups
  - Common split size: 70%-30%
  - Report error on holdout set
  - Train final model using all data
- Gold standard for measuring model's true prediction error



http://scott.fortmann-roe.com/docs/MeasuringError.html

Holdout Set Method: Properties

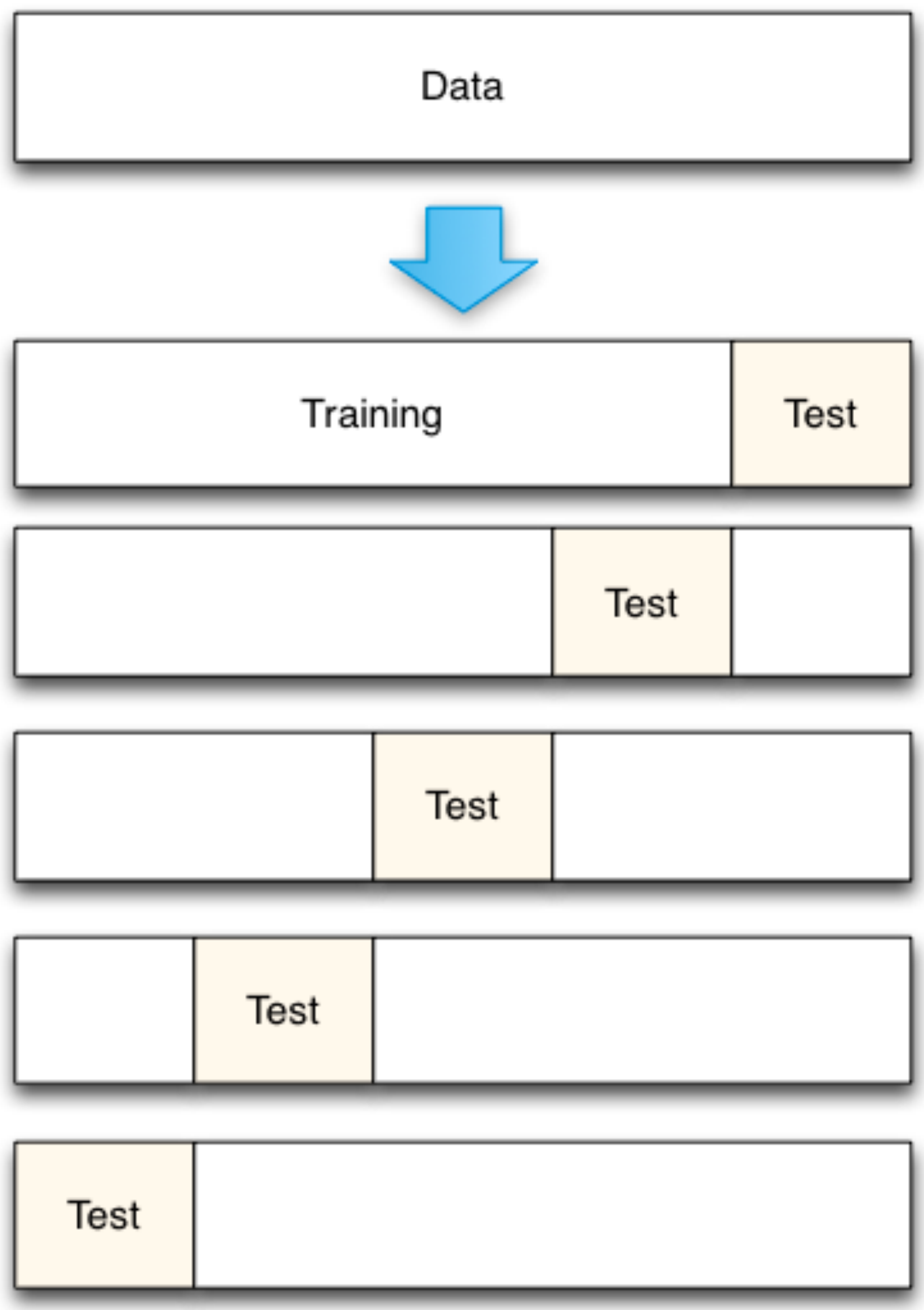
- Pros
  - No parametric or theoretic assumptions
  - Highly accurate with sufficient data
  - Simple to implement
  - Conceptually simple
- Cons
  - Potential conservative bias
  - Model contamination (use of holdout set prior to completion)
  - Size of holdout set impacts training sample

Cross-validation

K-fold Cross-validation

- Simple, intuitive way to estimate prediction error / generalization error
- Widely used method
- Procedure given training data and an estimator:
  - Split the training data into K parts or “folds”
  - Train on all but the kth part and validate on the kth part
  - Rotate and report average over K error measurements

5-fold Cross-validation Graphically



http://scott.fortmann-roe.com/docs/MeasuringError.html

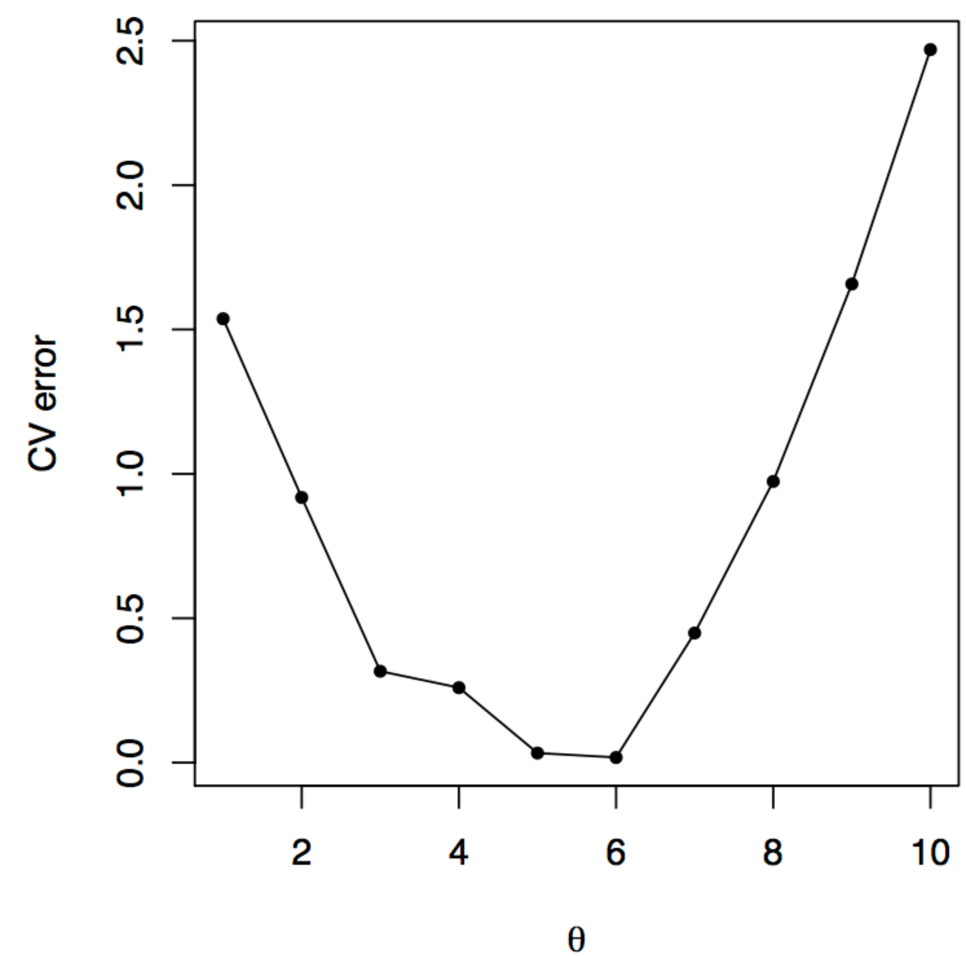
Cross-validation Error Curve

- Average error over all folds

$$\mathrm{CV}(\theta) = \frac{1}{n}\sum_{k=1}^{K}\sum_{i \in F_k}(y_i - \hat{f}_\theta^{-k}(\mathbf{x}_i))^2$$

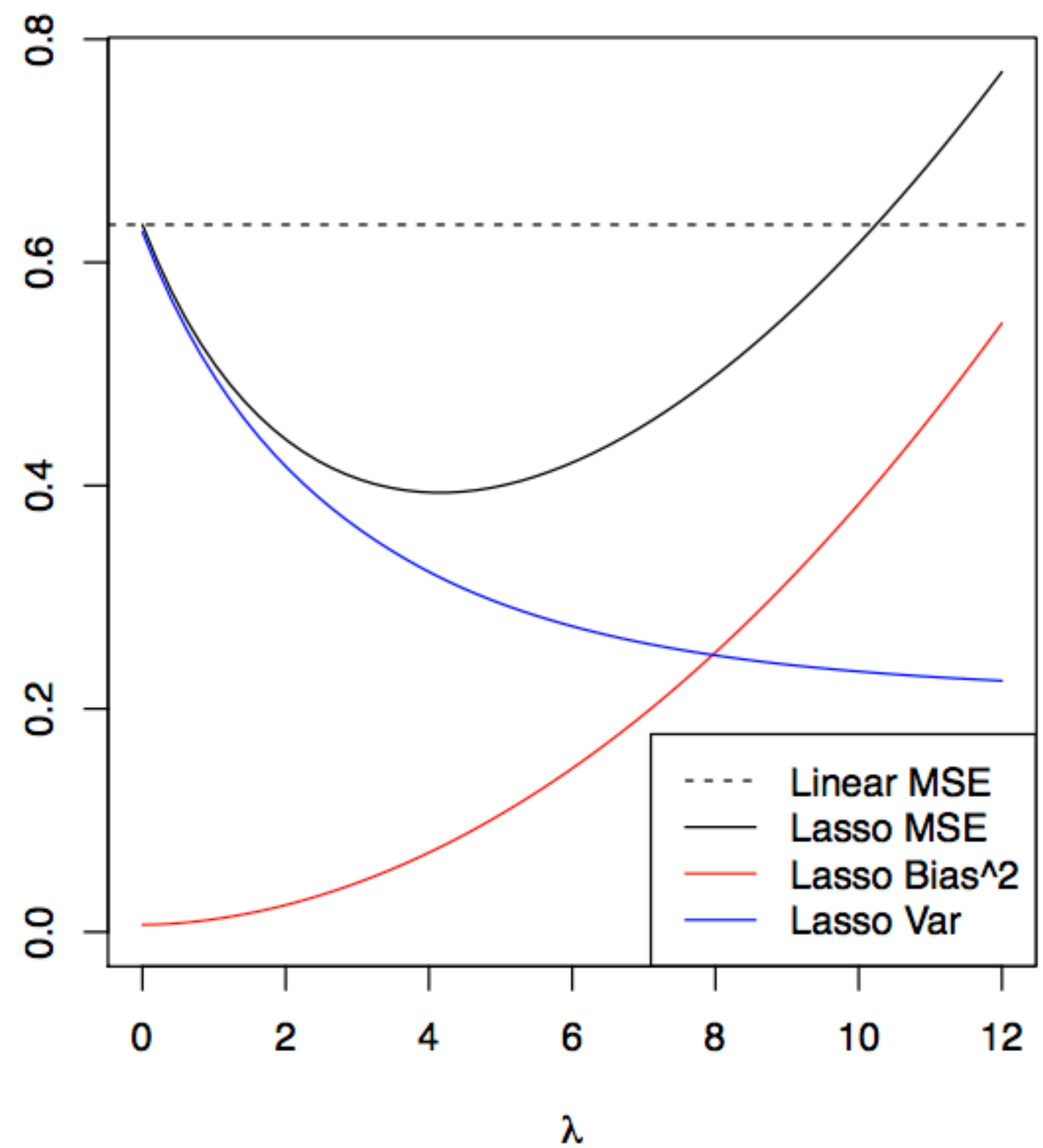
- Choose tuning parameter that minimizes the curve

$$\hat{\theta} = \mathrm{argmin}_{\theta \in \{\theta_1, \cdots, \theta_m\}} \mathrm{CV}(\theta)$$

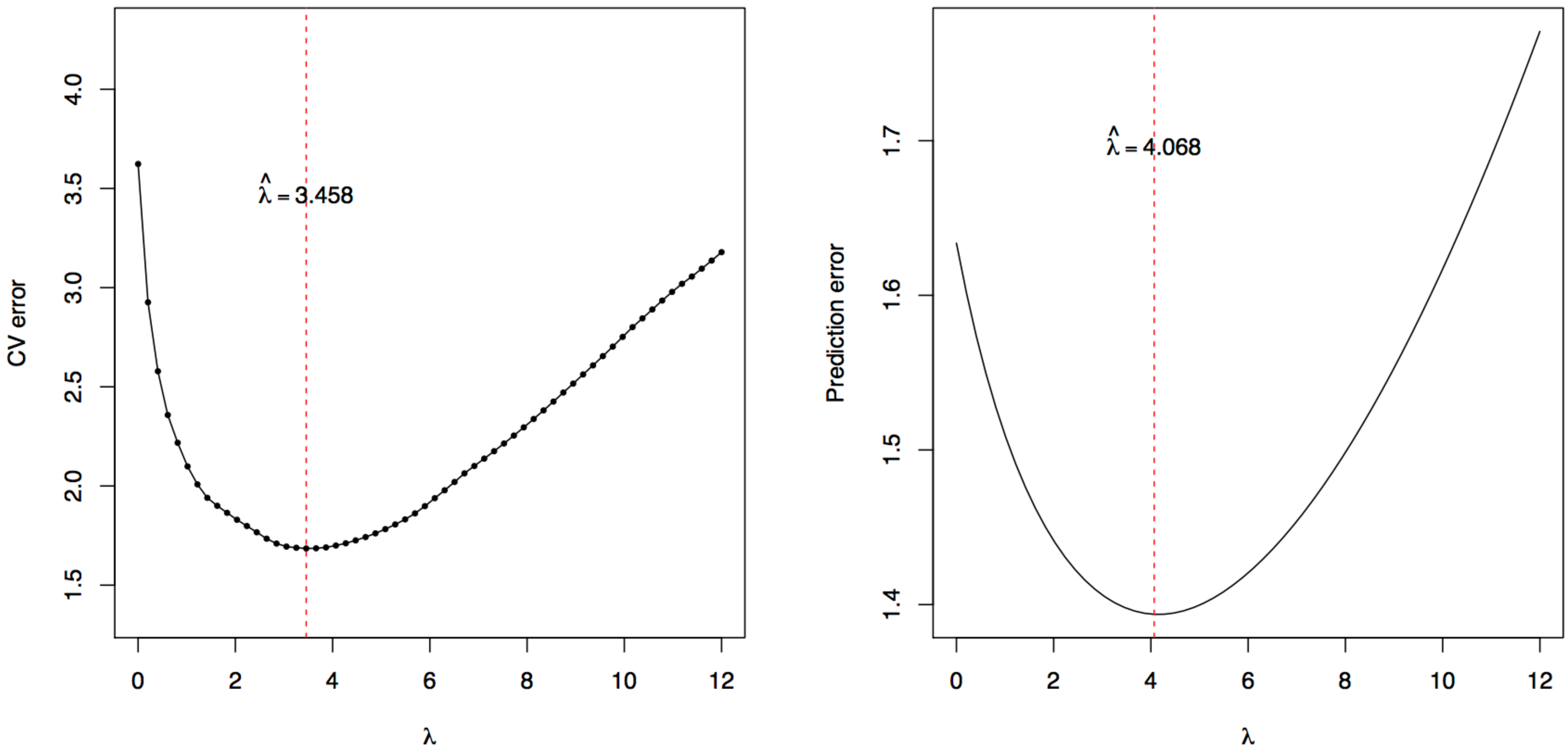


Example: Simulated Linear Model

- n = 50
- p = 30
- 10 non-zero coefficients



Example: Simulated Linear Model



Selected regularization parameter is close to real parameter

Cross-validation Standard Errors

- For k-fold cross-validation (small K << n), we can estimate standard deviation at each parameter

- Average validation errors:

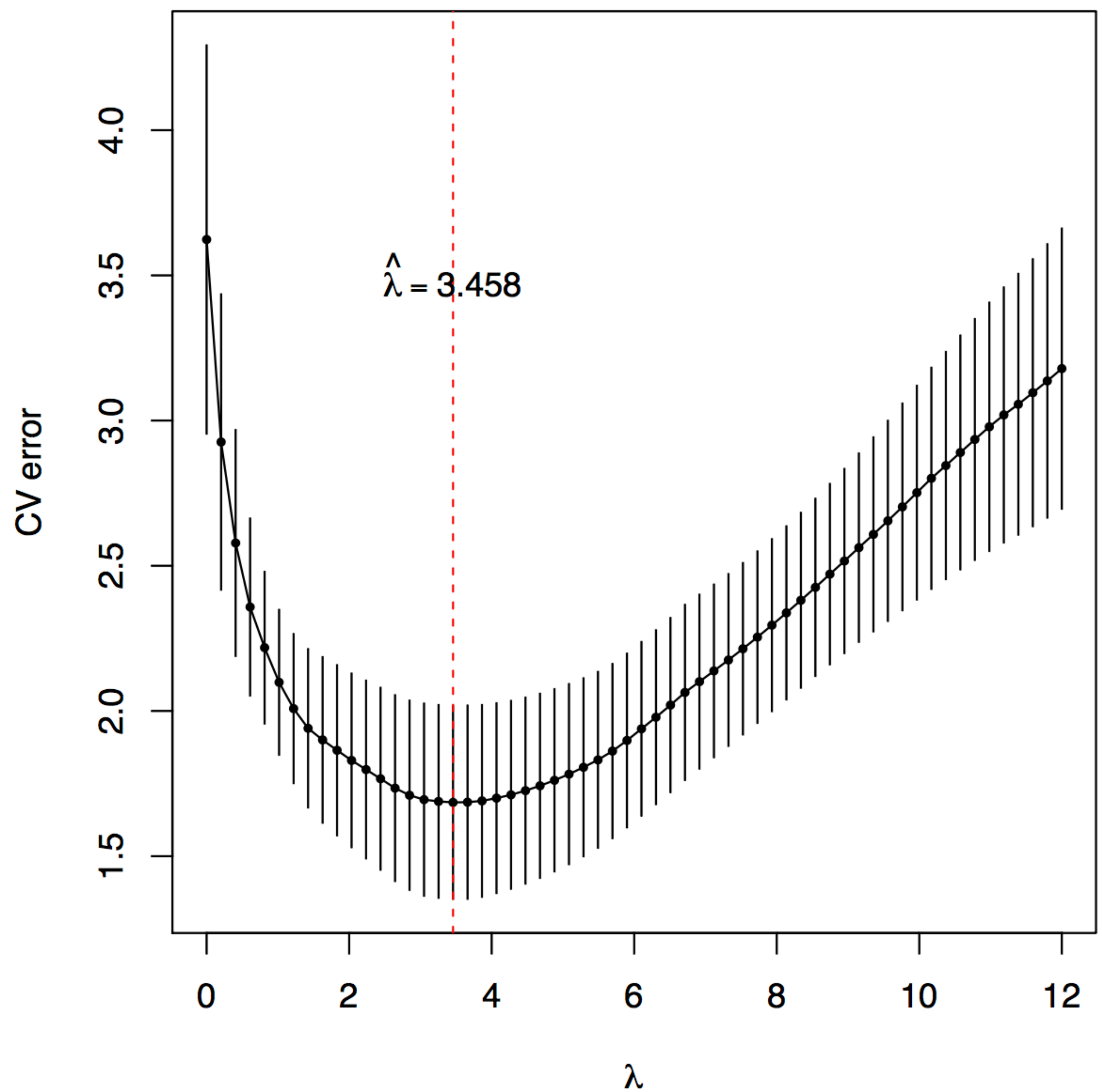
$$\mathrm{CV}_k(\theta) = \frac{1}{n_k} \sum_{i \in F_k} (y_i - \hat{f}_\theta^{-k}(\mathbf{x}_i))^2$$

- Sample standard deviation:

$$\mathrm{SD}(\theta) = \sqrt{\mathrm{var}(\mathrm{CV}_1(\theta), \cdots, \mathrm{CV}_K(\theta))}$$

- Standard error: $\mathrm{SE}(\theta) = \mathrm{SD}(\theta)/\sqrt{K}$

Example: Simulated Linear Model

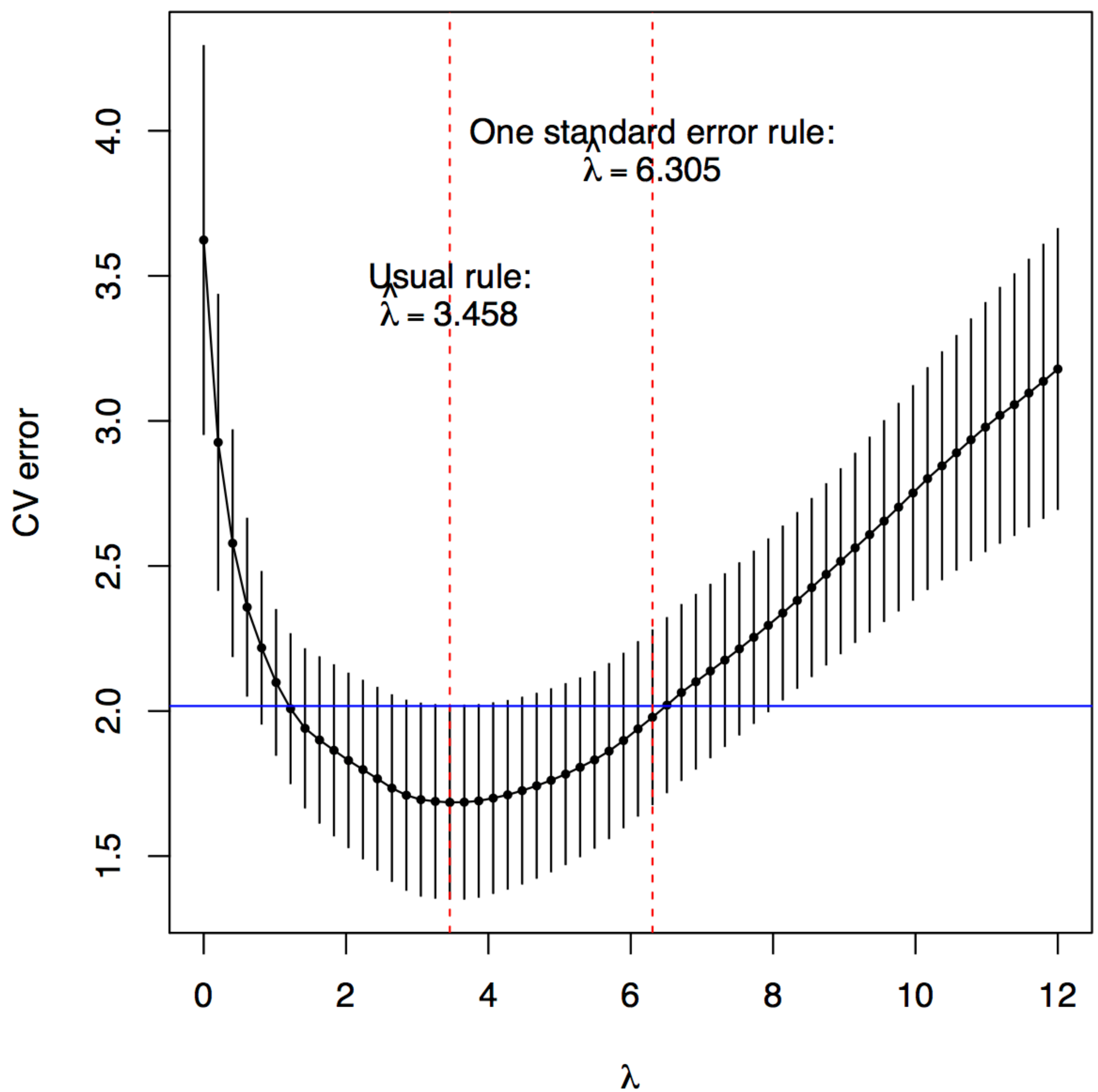


One Standard Error Rule

- Alternative rule for selection of tuning parameter
- Idea: "All else equal (up to one standard error), go for the simpler (more regularized) model"
- Find usual minimizer as before
- Move parameter in direction of increasing regularization such that cross-validation error curve is within one standard error

$$\mathrm{CV}(\theta) \leq \mathrm{CV}(\hat{\theta}) + \mathrm{SE}(\hat{\theta})$$

Example: One Standard Rule



Choice of K

- Want to train using as much data as possible
  - Allows for more complex models
  - Improves accuracy of the models
- Common values of K
  - K = 2 (two-fold cross validation)
  - K = 5, 10 (5-fold, 10-fold cross validation)
  - K = N (leave one out cross validation or LOOCV)

LOOCV

- N - 1 samples for training, 1 sample for test
- More samples for training, what can go wrong?
- How does it do for the bias / variance tradeoff?

LOOCV Bias

- Training with N-1 samples approximates training with N samples
- Large number of training samples means the average LOOCV estimation will be close to Err for a predictor trained on N samples

LOOCV Variance

- Not independent looks at the data
  - Any two training folds share N-2 samples
- No measure of sensitivity to training data
- Error can change considerably from one training dataset to another —> high variance!

2-fold CV: Bias

- Prediction accuracy for a model trained with N/2 samples could be lower than for a model trained with N samples
- Repeating two-fold CV over many training datasets, we would not expect the mean to converge to the true generalization error —> higher bias!

2-fold CV: Variance

- Training folds are completely independent of one another
- Provides a better measure of the sensitivity to training data —> lower variance

K = 5 vs K = 10

- Depends on the size of the training data available
- Returns back to the bias versus variance tradeoff
  - K = 5 will have higher bias, lower variance
  - K = 10 will have lower bias, higher variance

5-fold CV: Hypothetical Learning

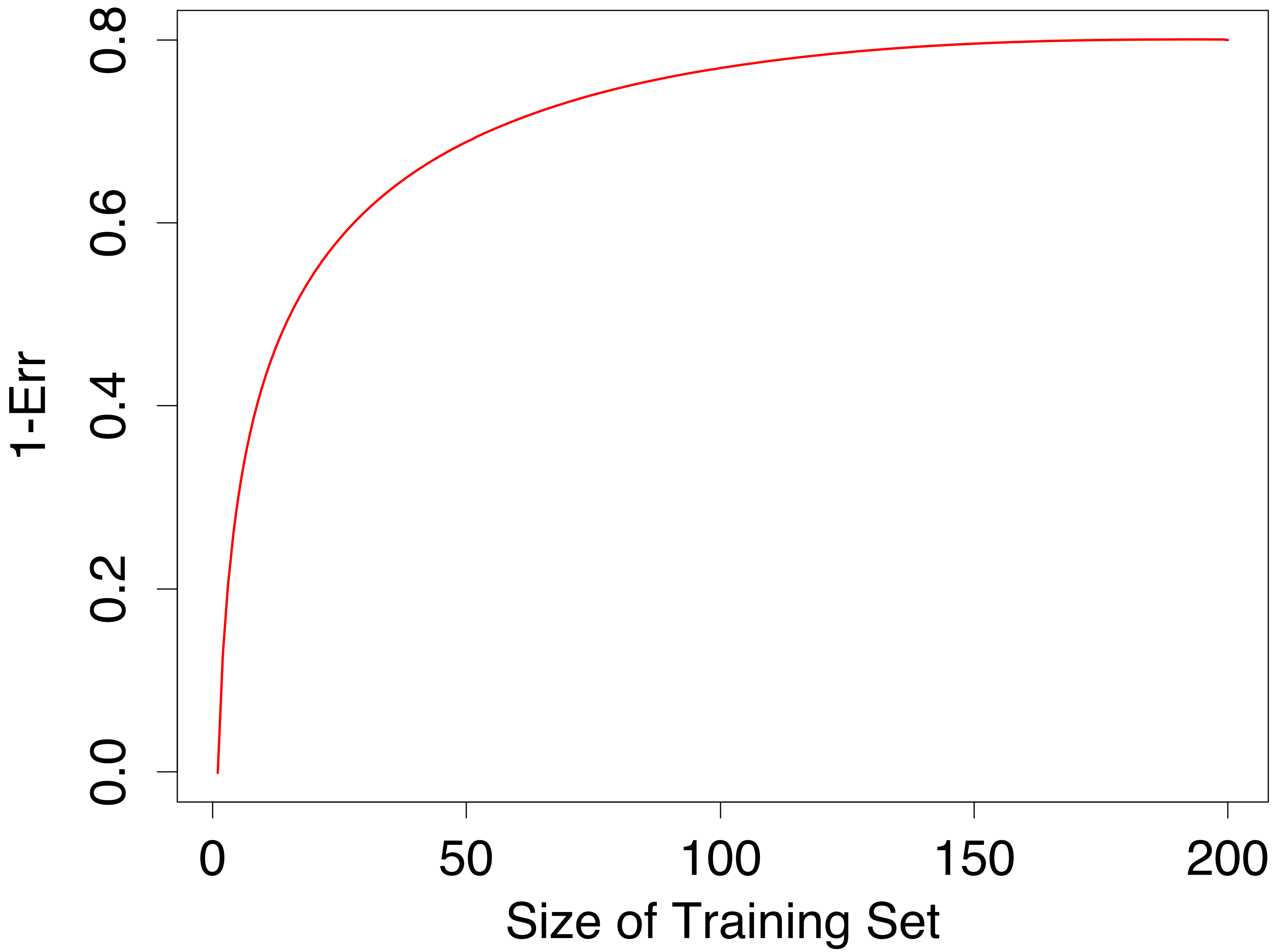
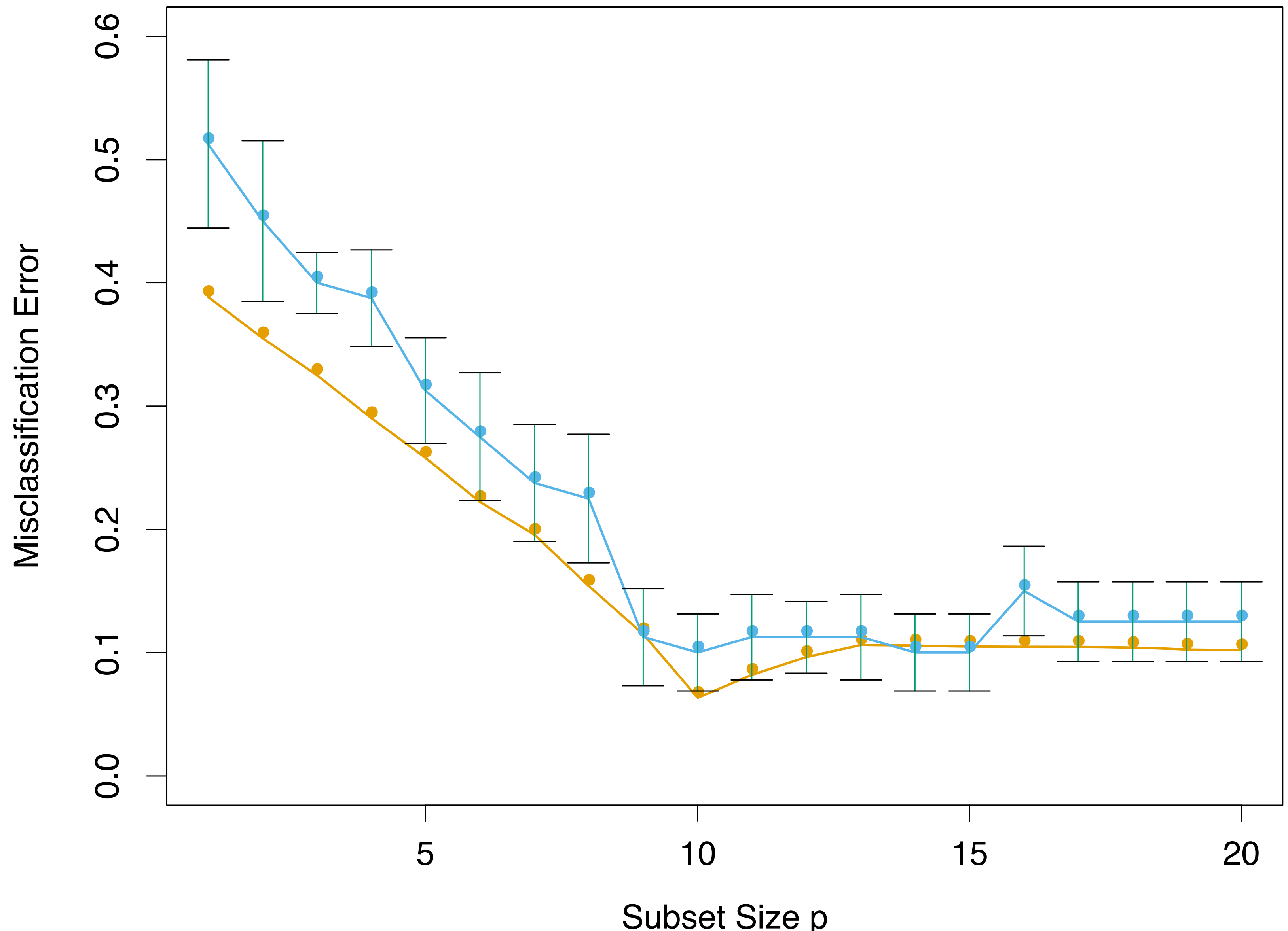


Figure 7.8 (Hastie et al.)

10-fold CV



prediction error = orange

CV = blue

Figure 7.9 (Hastie et al.)

Conditional and Expected Error

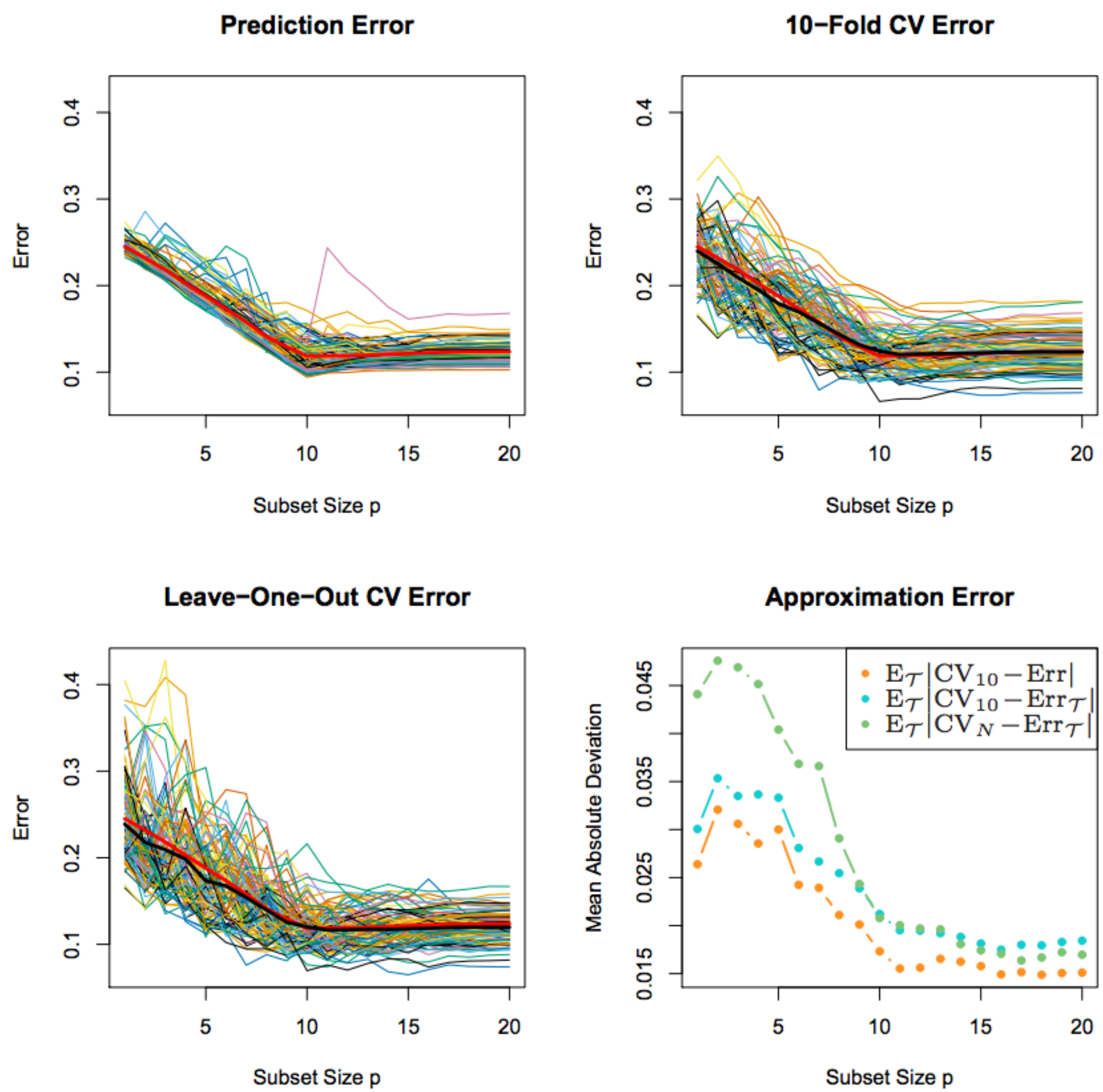


Figure 7.14 (Hastie et al.)

K-Fold Cross-Validation: Best Practices

- Typically choose K=5, K=10
- Depends on how much data is available, how sensitive our method is to amount of training data
- Be cautious with LOOCV
  - “Abundant” data should not use LOOCV

Cross-validation Question

- Classification problem with a large number of predictors
- Strategy 1:

  1. Find a “subset” of good predictors that show fairly strong (univariate) correlation with class labels

  2. Use this subset of predictors to build a multivariate classifier using K-fold CV

  3. Estimate the prediction error of the final model

Cross-validation Question

- Strategy 2:
  - Divide the samples into K-fold CV at random
  - For each fold
    1. Find a subset of good predictors that show fairly strong (univariate) correlation with class labels using all samples except those in fold k
    2. Build a multivariate classifier using the samples
    3. Use classifier to predict class labels for samples in fold k

Which Strategy is Right?

- Imagine a case where N = 50 samples of equal-sized classes and p = 5000 features independent of class labels
- True error rate of any classifier is 50%

Strategy 1

- Step 1: 100 predictors having highest correlation with class labels
- Step 2: Build a model based on these 100 predictors
- Step 3: Over 50 simulations, average CV error rate is 3%

What went wrong?

5-fold CV: Hypothetical Learning

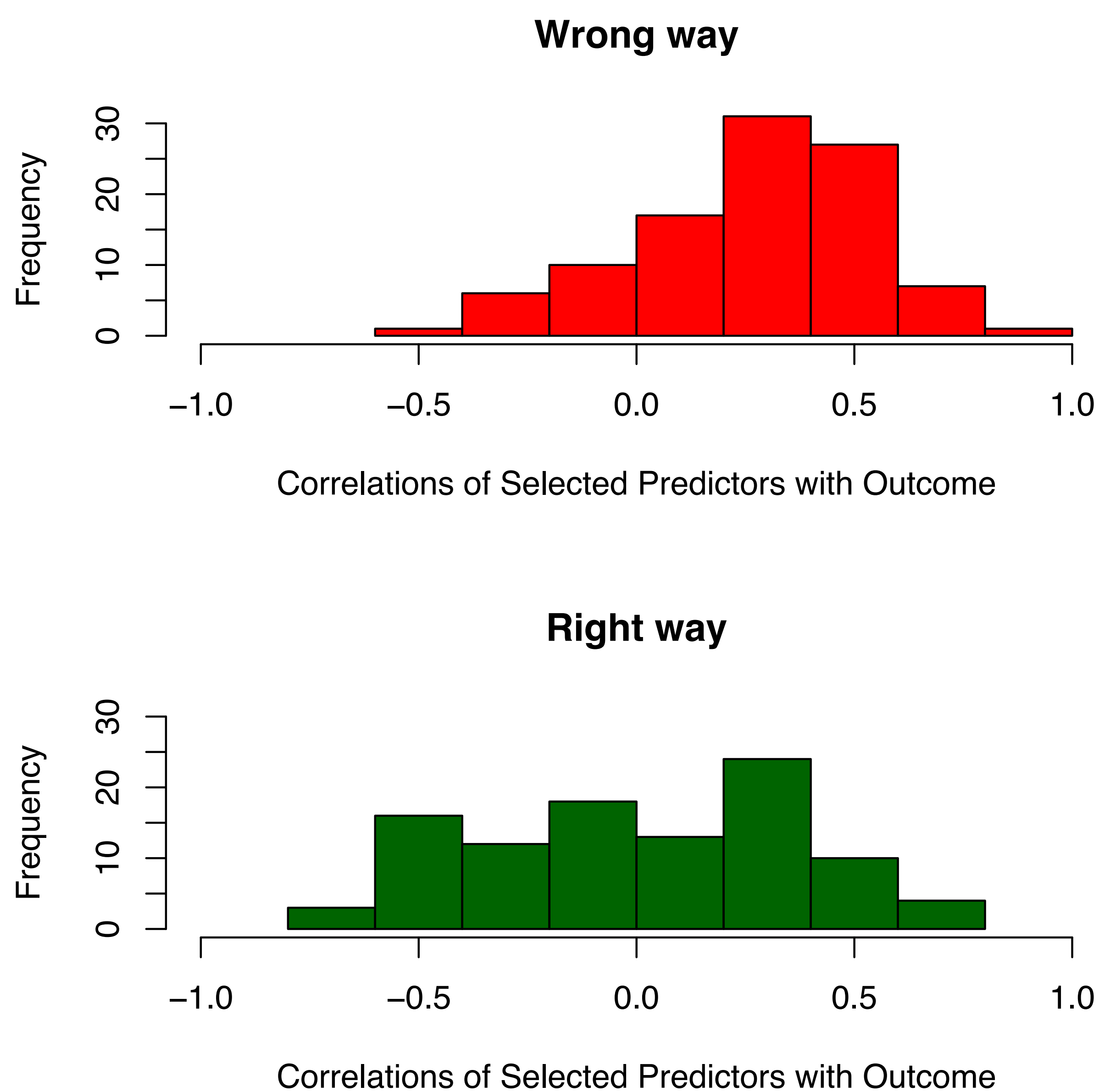


Figure 7.10 (Hastie et al.)

Generalized Cross-validation

- Shortcut for linear fitted models using squared error loss and LOOCV

- Consider ridge regression:

$$\hat{f}_\lambda(\mathbf{x}_i) = \mathbf{x_i}^\top \hat{\beta} = \mathbf{x_i}^\top (\mathbf{X}^\top X + \lambda \mathbf{I})^{-1} \mathbf{X}^\top y$$

- CV can be computed as:

$$\frac{1}{n} \sum_i (y_i - \hat{f}_\lambda(\mathbf{x}_i))^2 = \frac{1}{n} \sum_i \left[\frac{y_i - \hat{f}_\lambda(\mathbf{x}_i)}{1 - S_{ii}} \right]^2,$$

$$\text{where } \mathbf{S} = \mathbf{X}^\top (\mathbf{X}^\top X + \lambda \mathbf{I})^{-1} \mathbf{X}^\top y$$

Generalized Cross-validation

- For a general linear fitting model where:

$$\hat{y} = (\hat{f}(\mathbf{x}_1), \cdots, \hat{f}(\mathbf{x}_n)) = \mathbf{S}y$$

- General CV approximation is:

$$\mathrm{GCV}(\hat{f}) = \frac{1}{n} \sum_i \left[\frac{y_i - \hat{f}(\mathbf{x}_i)}{1 - \mathrm{Trace}(\mathbf{S})/N} \right]^2$$

Huge computational savings when trace of S can be computed more easily than individual elements S_{ii}

CV: Properties

- Pros
  - No parametric or theoretic assumptions
  - Highly accurate with sufficient data
  - Conceptually simple
- Cons
  - Computationally intensive
  - Must choose fold size
  - Potential conservative bias

Monte-Carlo Cross-Validation

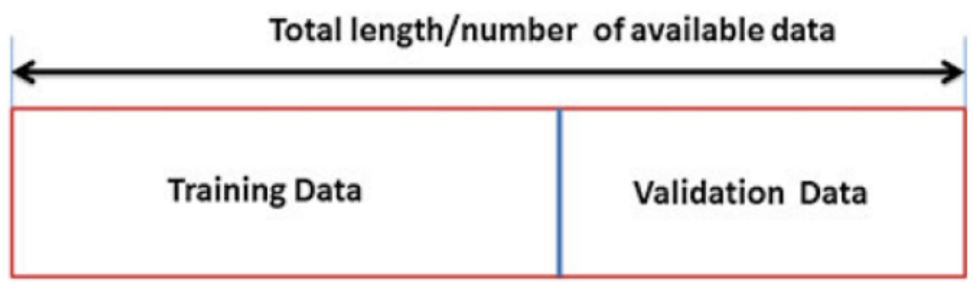
- AKA random sub-sampling
  - Randomly select (without replacement) some fraction of your data to form training set
  - Assign rest to test set
  - Repeat multiple times with new partitions
- Major difference to k-fold cross-validation: same point can appear in multiple test sets!

K-Fold vs Monte-Carlo

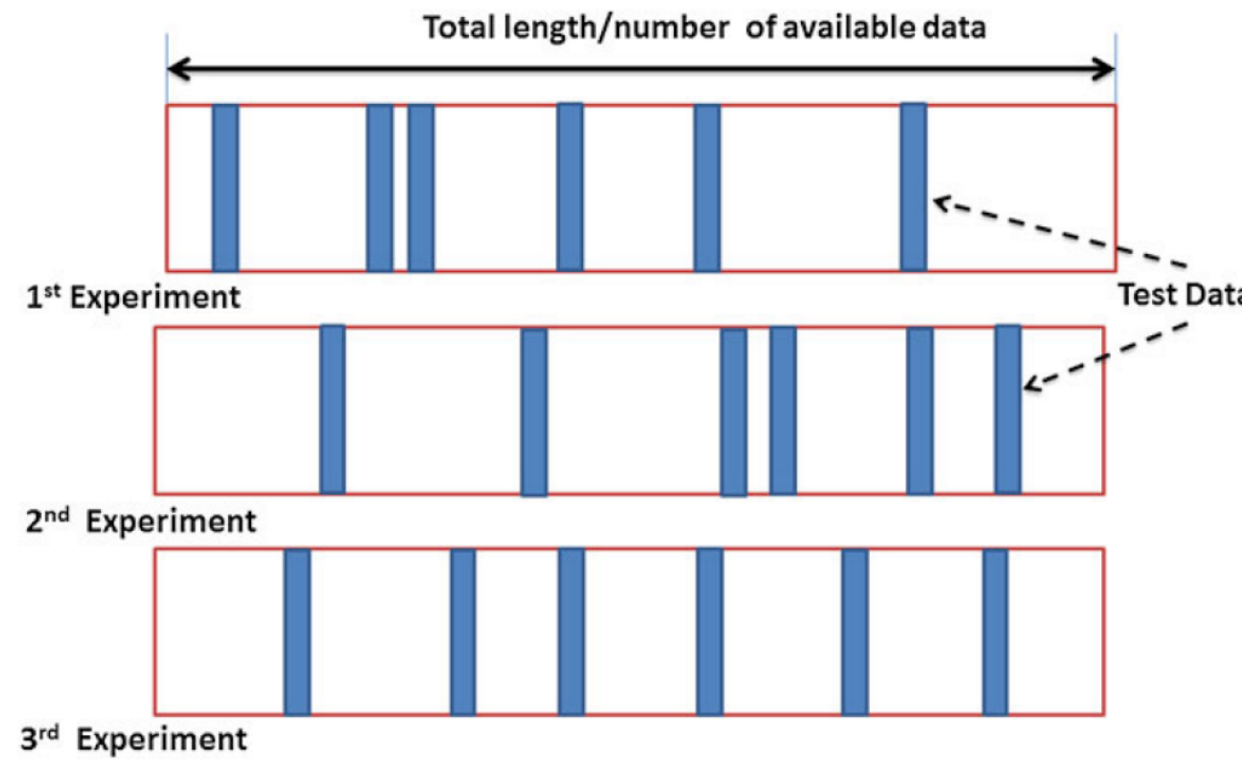
- Cross-validation only explores a few of the possible ways to partition the data
    - Unbiased estimate but with high variance
- Monte-Carlo allows you to explore many more possible partitions
    - Less variance but more biased estimate

Validation Methods: Graphically

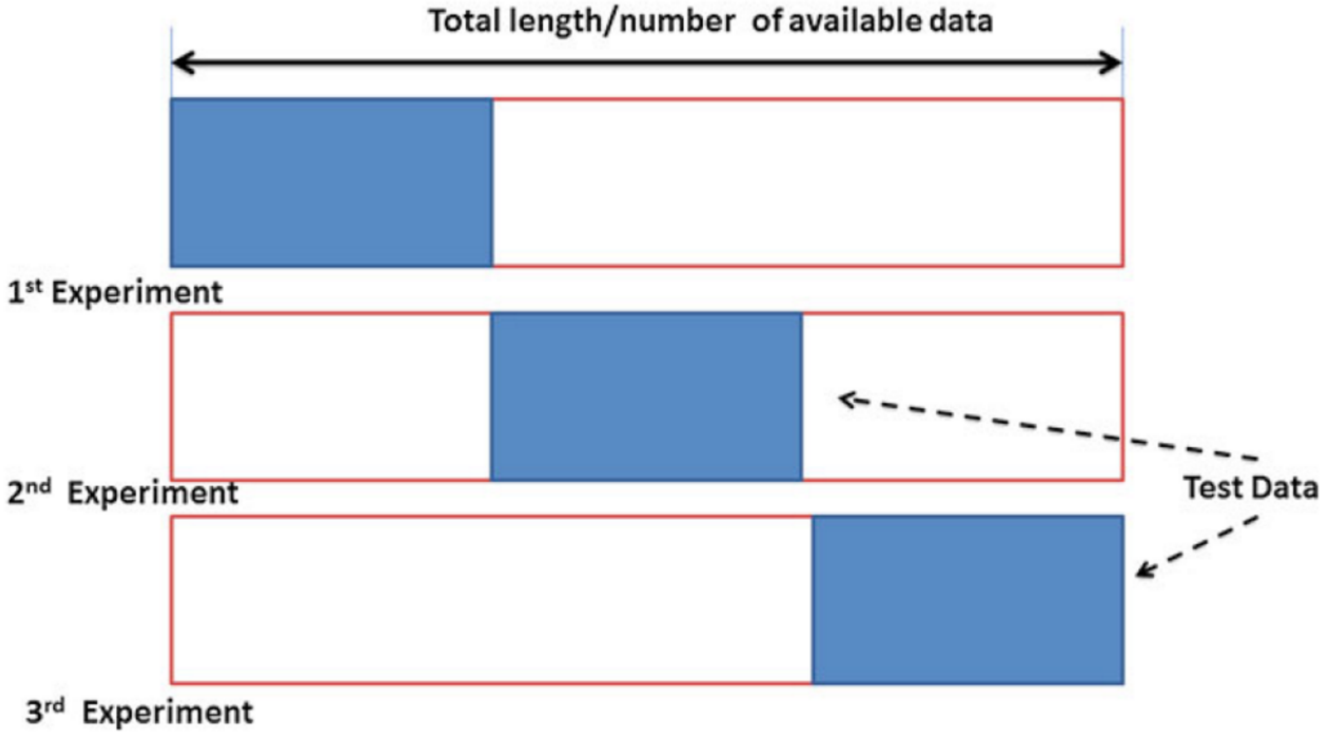
Holdout



Monte-Carlo cross-validation



k-fold cross-validation



Figures 3.6, 3.7, 3.8 (Remesan & Mathew. Hydrological Data Driven Modeling: A Case Study Approach)

CV Notes

- CV must be applied to the entire sequence of modeling steps
- Samples should be “left out” before any selection or filtering steps are applied
- Initial unsupervised screening steps can be done before samples are left out