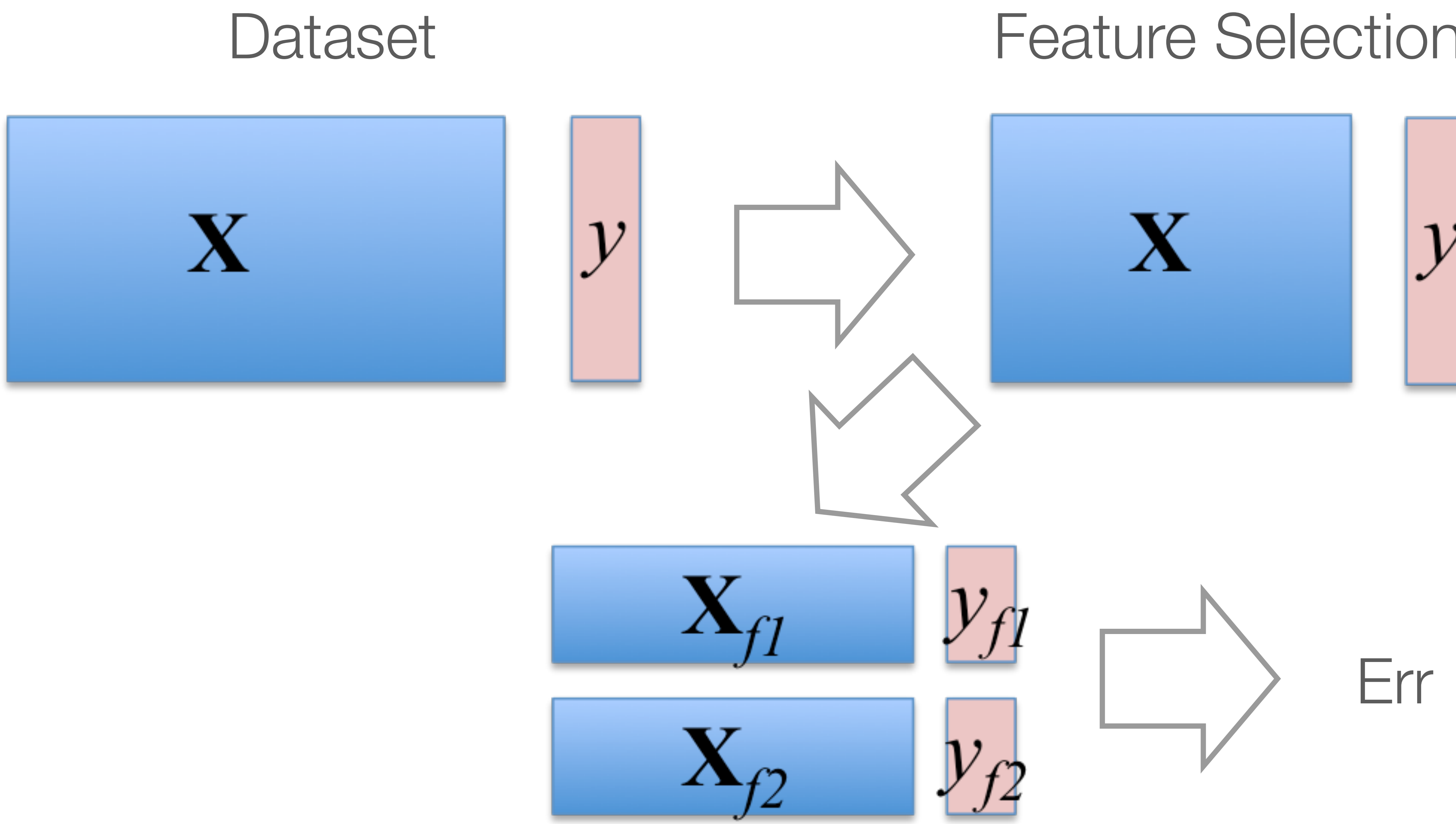
Model Selection & Bootstrap

CS 534: Machine Learning

Slides adapted from Lee Cooper, Piyush Rai, and Ryan Tibshirani

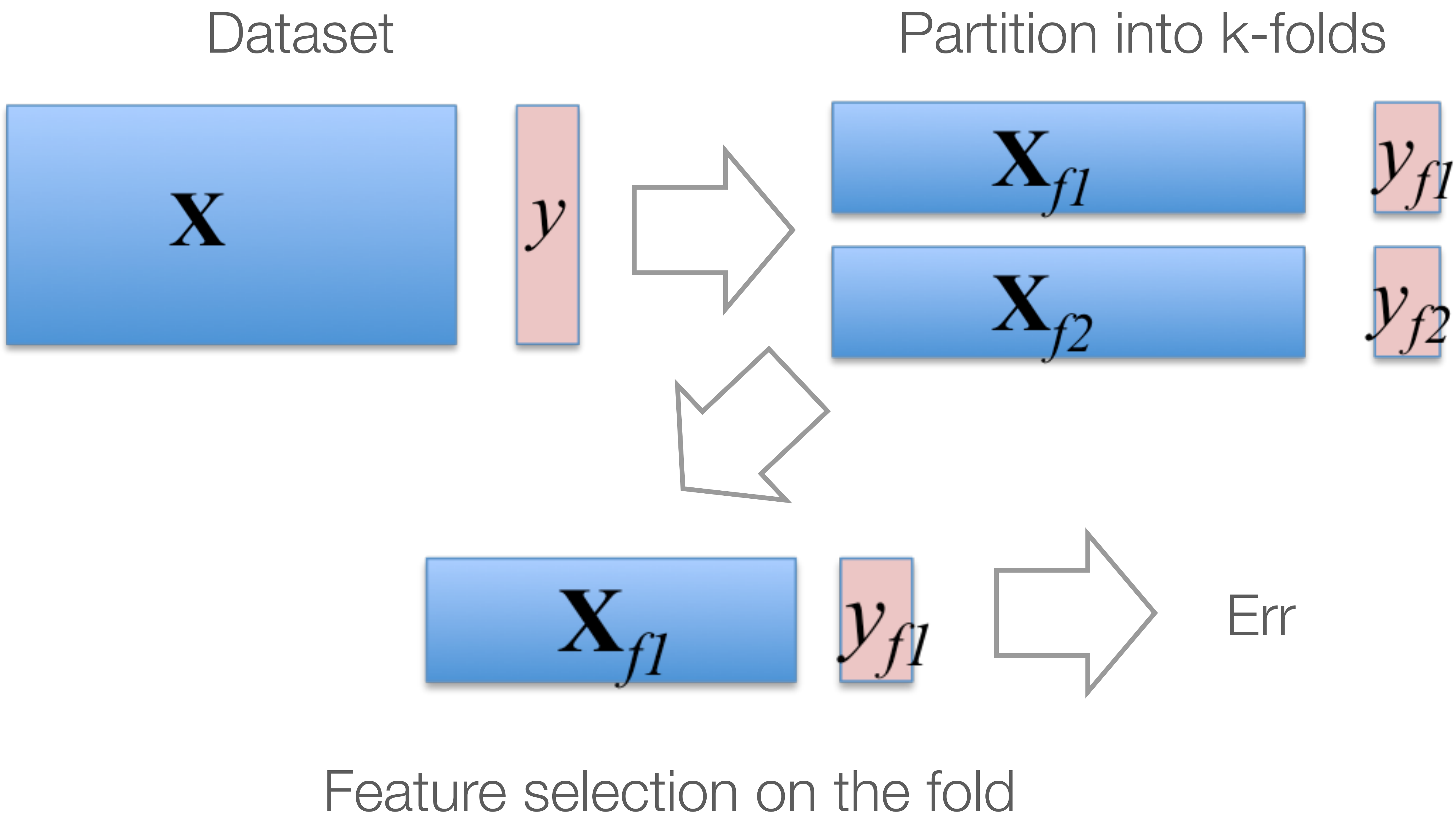
Review: Validation

Example: Improper Validation



Cross-validation on selected features

Example: Proper Validation

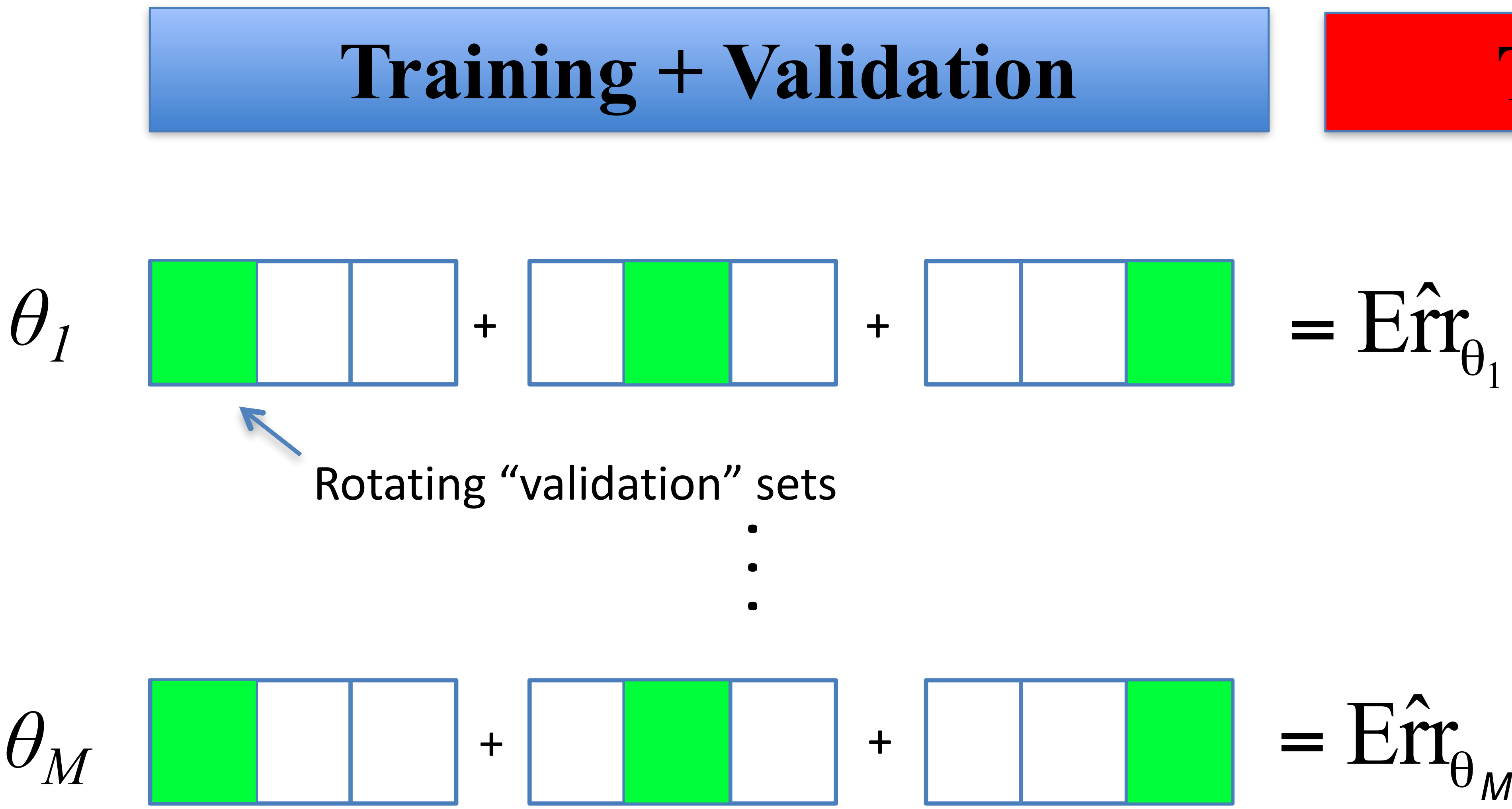


Model Selection

CV & Model Selection

- Consider an algorithm with parameters θ that needs to be tuned
- How to do both model selection and model assessment within a cross-validation framework?

Nested CV (K=3)

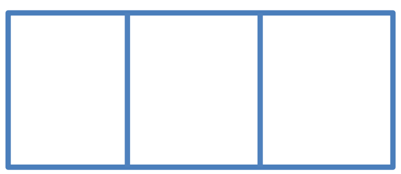


$$\theta^* = \underset{i}{\operatorname{argmin}} \ \hat{\mathrm{Err}}_{\theta_i}$$

Model Selection
(do not report this error!)

Nested CV (K=3)

Build optimal model using your non-testing samples



$$+ \quad \theta^* = \underset{i}{\operatorname{argmin}} \ \hat{Err}_{\theta_i} \Rightarrow \mathit{Model}^*$$

Report test error on testing samples (report this)

$$\mathit{Model}^* \quad + \quad \boxed{\textbf{Test}} \Rightarrow Err$$

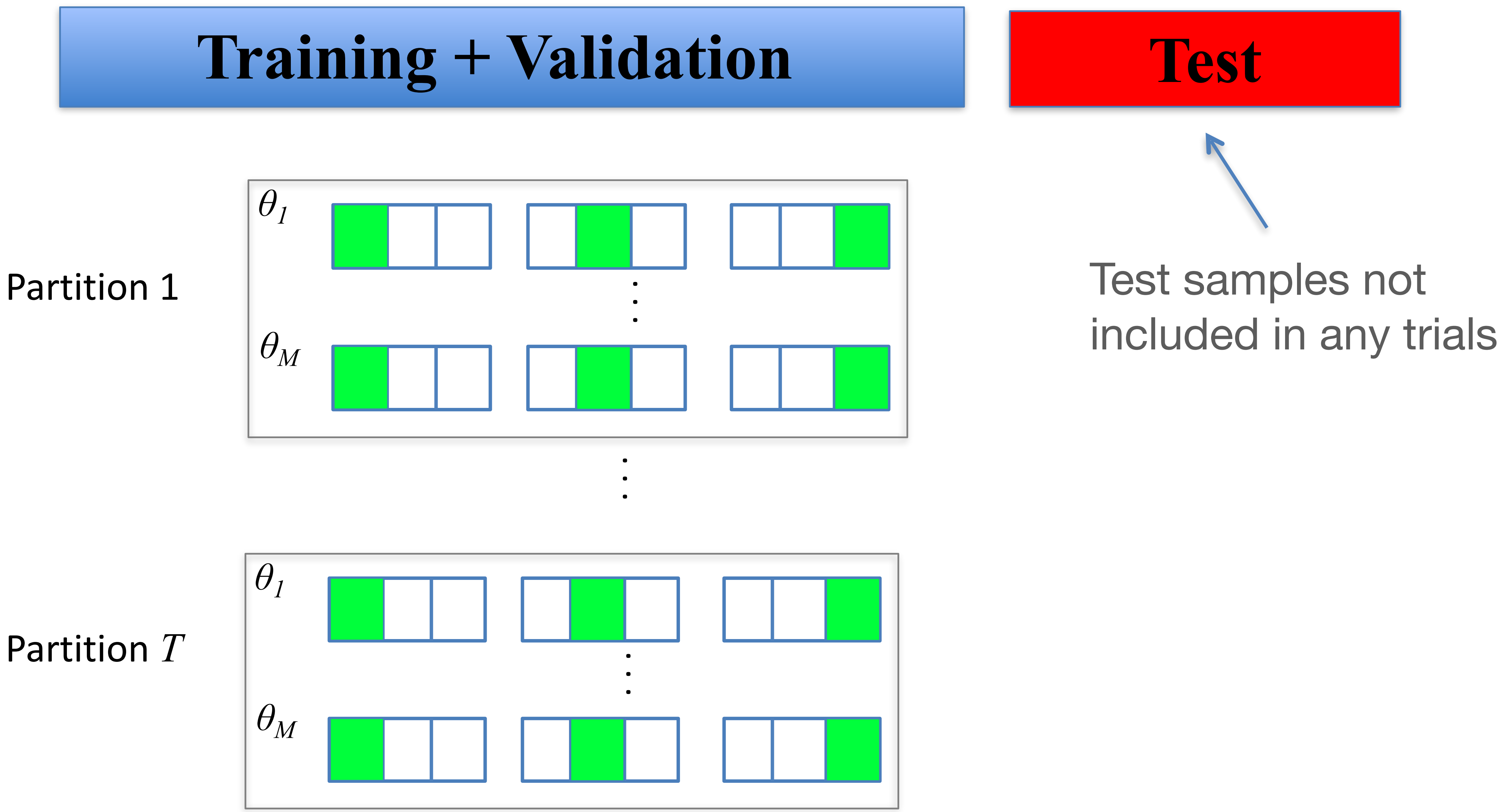
Nested CV

1. Generate T partitions of training + validation samples only

2. Use validation errors from all partitions to estimate the optimal parameters

3. Train a single model with the optimal parameters and evaluate on test samples

Nested CV: Pictorially



Nested CV: The Wrong Way

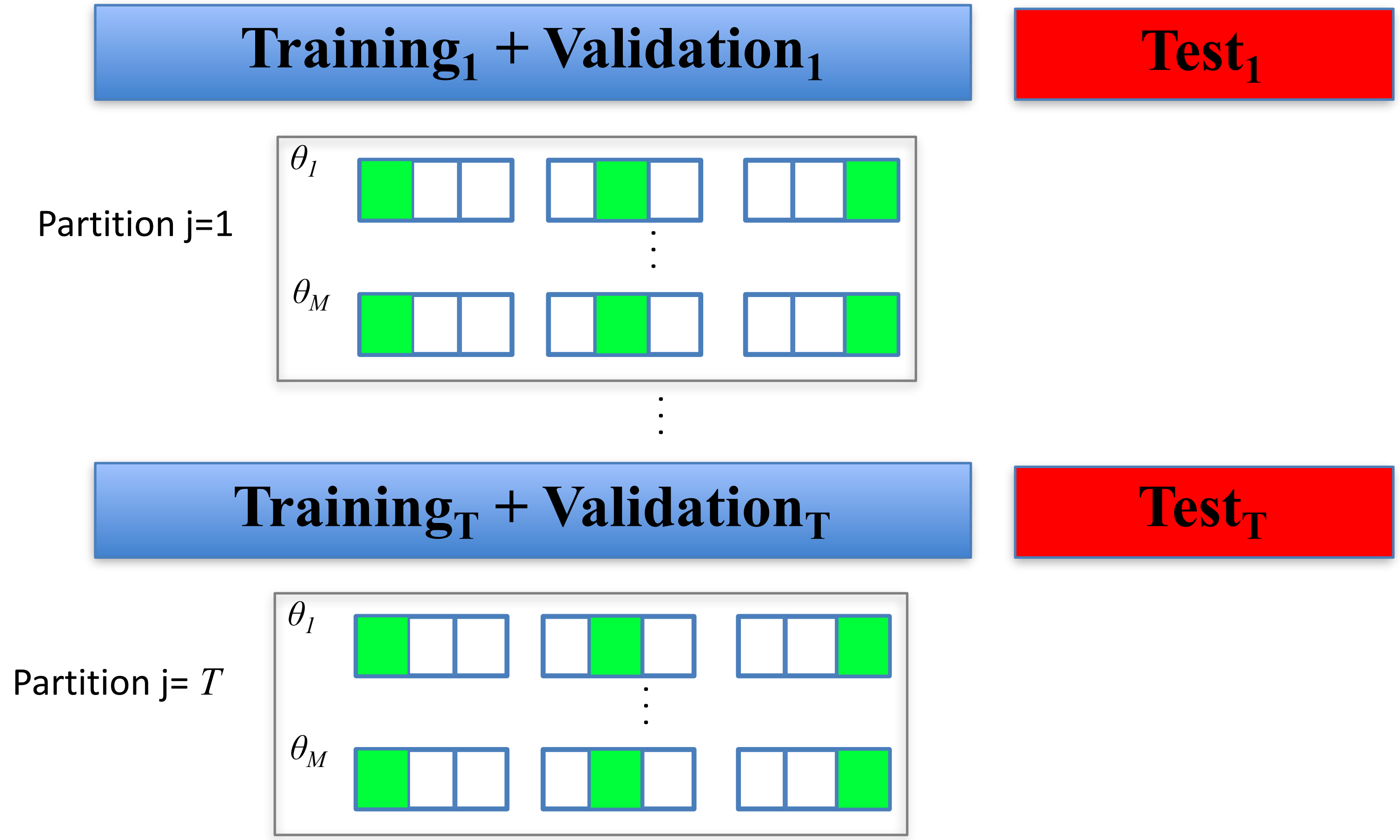
1. Estimate best parameter for all partitions

$$\theta^* = \text{argmin}_{\theta_i} \sum_{\text{partitions}} \overline{\text{Err}}_{\theta_i}$$

2. Fit a model using θ^* and evaluate on all Test_j

$$\text{Err} = \sum_{\text{partitions}} L(\hat{f}_{\theta^*}, \text{Test}_j)$$

Nested CV: The Correct Way



1. Estimate best parameter for one partition

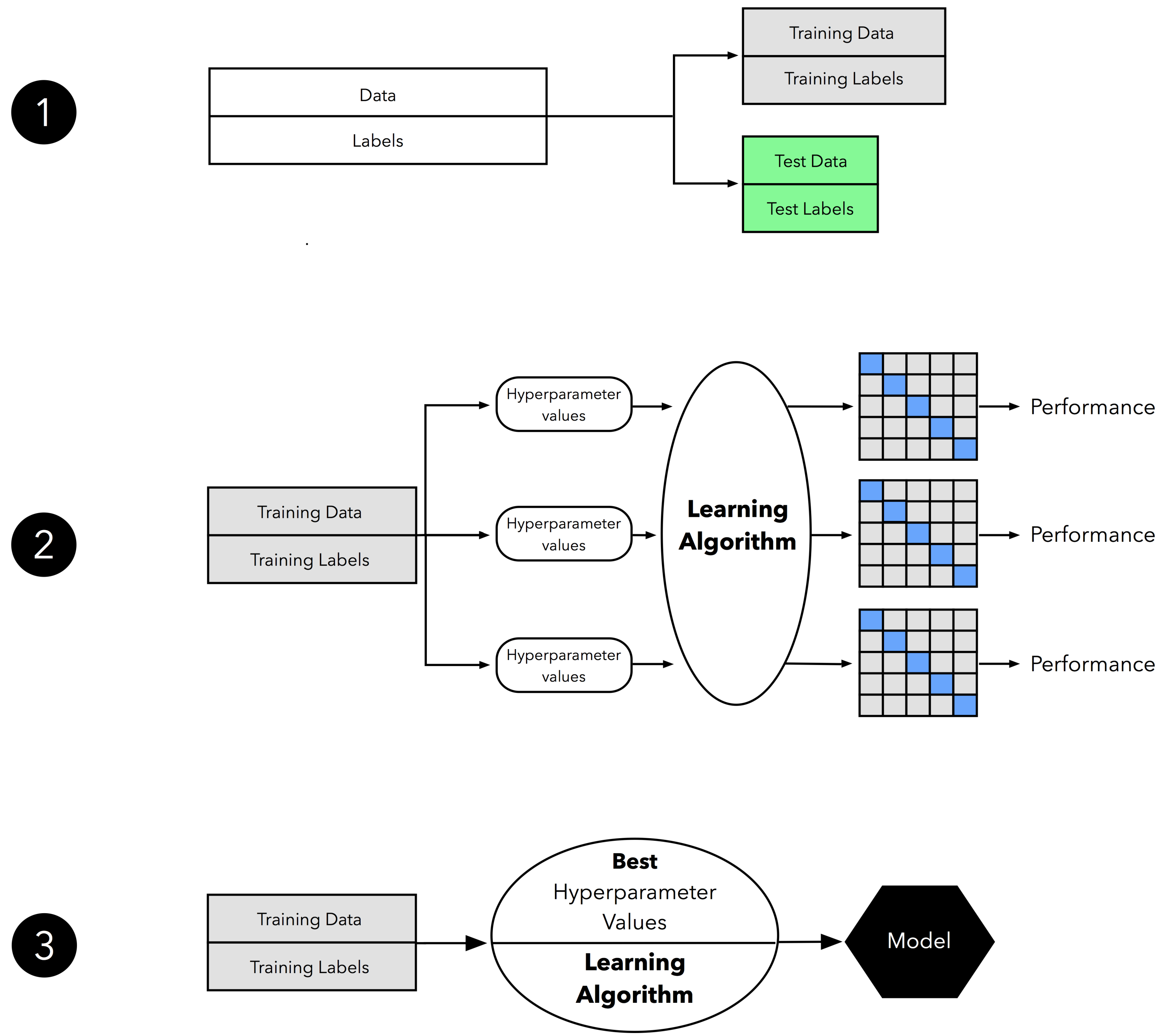
$$\theta_j^* = \text{argmin}_i \; \overline{\text{Err}}_{\theta_i}$$

2. Apply the best parameter for each partition to that partition's test samples only

$$\text{Err} = \sum_{\text{partitions}} L(\hat{f}_{\theta_j^*}, \text{Test}_j)$$

Best model from partition j

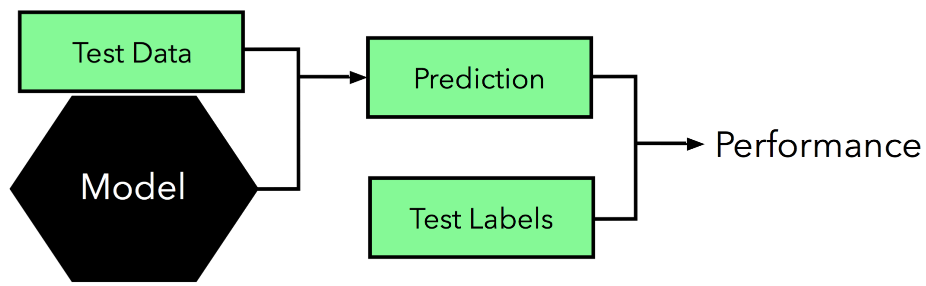
Best Practices: Model Selection



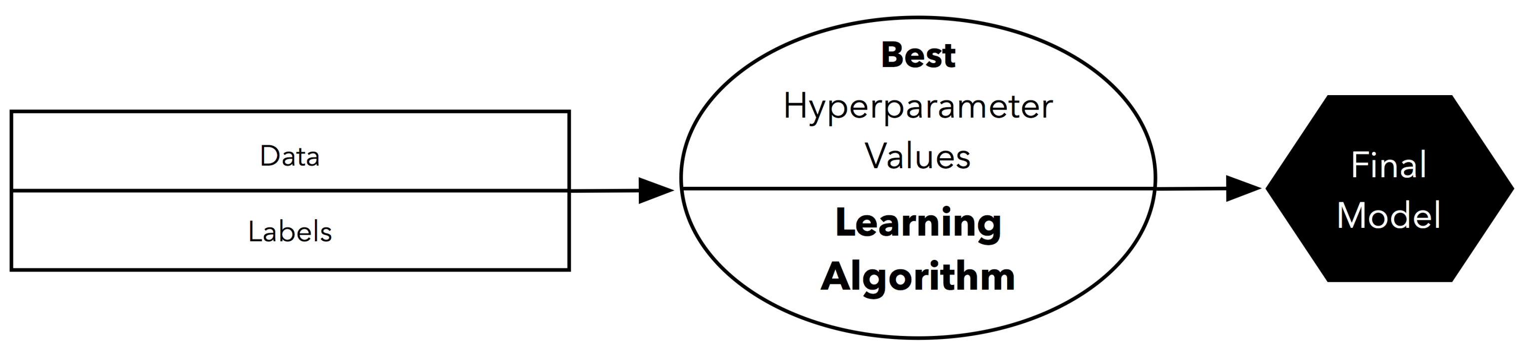
https://sebastianraschka.com/blog/2016/model-evaluation-selection-part3.html

Best Practices: Model Selection

4



5



https://sebastianraschka.com/blog/2016/model-evaluation-selection-part3.html

Validation: Takeaway

- Validation can be confusing topic
- Guidelines:
    - If you have to choose an error from multiple possible errors, then this error cannot be reported as test/generalization error
    - You cannot use the same samples to estimate both optimal model parameters and test/generalization error

Review: Training Error

- Estimator adapts to the training data and thus will have an overly optimistic estimate of the generalization error!
- Generalization error:

$$\mathrm{Err}_{\mathcal{T}} = E_{X^0,Y^0}[L(Y^0, \hat{f}(X^0))|\mathcal{T}]$$

- Expected error:

$$\mathrm{Err} = E_{\mathcal{T}}[E_{X^0,Y^0}[L(Y^0, \hat{f}(X^0))|\mathcal{T}]]$$

Training Error Optimism

- Training error is less than true error

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

- In-sample error

$$\mathrm{Err}_{\mathrm{in}} = \frac{1}{N}\sum_i E_{Y^0}[L(Y_i^0, \hat{f}(X_i))|\mathcal{T}]$$

- Optimism

$$\mathrm{op} = \mathrm{Err}_{\mathrm{in}} - \mathrm{TrainErr}$$

Rationale for Optimism

- Expect good performance at or close to x_i in training set and future samples unlikely to coincide with same x_i
- Noise: imagine drawing a new response at the same x_i using conditional distribution

Average Optimism

- Optimism is usually positive since training error is biased downward
- Average optimism (expectation of training sets)

$$w = E_y(\mathrm{op})$$

- For squared error, 0-1, and other loss functions

$$w = \frac{2}{N}\sum_i \mathrm{Cov}(\hat{y}_i, y_i)$$

Harder we fit the data, higher the optimism

Optimism of Linear Fit

- Linear fit with additive error model and d inputs

$$\mathbf{y} = f(\mathbf{X}) + \epsilon$$

- Covariance simplifies to

$$\sum_i \mathrm{Cov}(\hat{y}_i, y_i) = d\sigma_\epsilon^2$$

- Average in-sample prediction error

$$E_y(\mathrm{Err_{in}}) = E_y(\mathrm{TrainErr}) + 2\frac{d}{N}\sigma_\epsilon^2$$

R^2

- "Goodness" of fit measure
- Easy interpretation —the percentage of variation in data explained by the model

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

- What is wrong with this predictor?

Adjusted R^2

- Adjust for model size

$$R_a^2 = 1 - \frac{n-1}{n-d-1}(1 - R^2)$$

- Interpretation — percentage of variation explained by only the independent variables that actually affect the dependent variable

Mallows C_p Statistic

- Under squared error loss with d parameters:

$$C_p = \text{TrainErr} + 2\frac{d}{N}\hat{\sigma}_\epsilon^2 \leftarrow \text{estimated from low bias model}$$

- Linear regression C_p statistic

$$C_p = \frac{\text{RSS}_d}{\hat{\sigma}_p^2} + 2d - N$$

- Think of the statistic as lack of fit + complexity parameter

Mallows C_p Statistic

- Easy to compute
- Closely related to adjusted R^2 and AIC
- For full model, $C_p = p$ exactly
- Disadvantage is the need to estimate the variance with full set of predictors

Akaike Information Criterion (AIC)

- Estimate of in-sample error when log-likelihood loss function is used

- Used as model selection criteria (takes into account both error and model complexity)

- Linear models:

$$\text{AIC} = 2\frac{d}{N} - \frac{2}{N}\log(\mathcal{L})$$

AIC: Estimation of In-sample Error

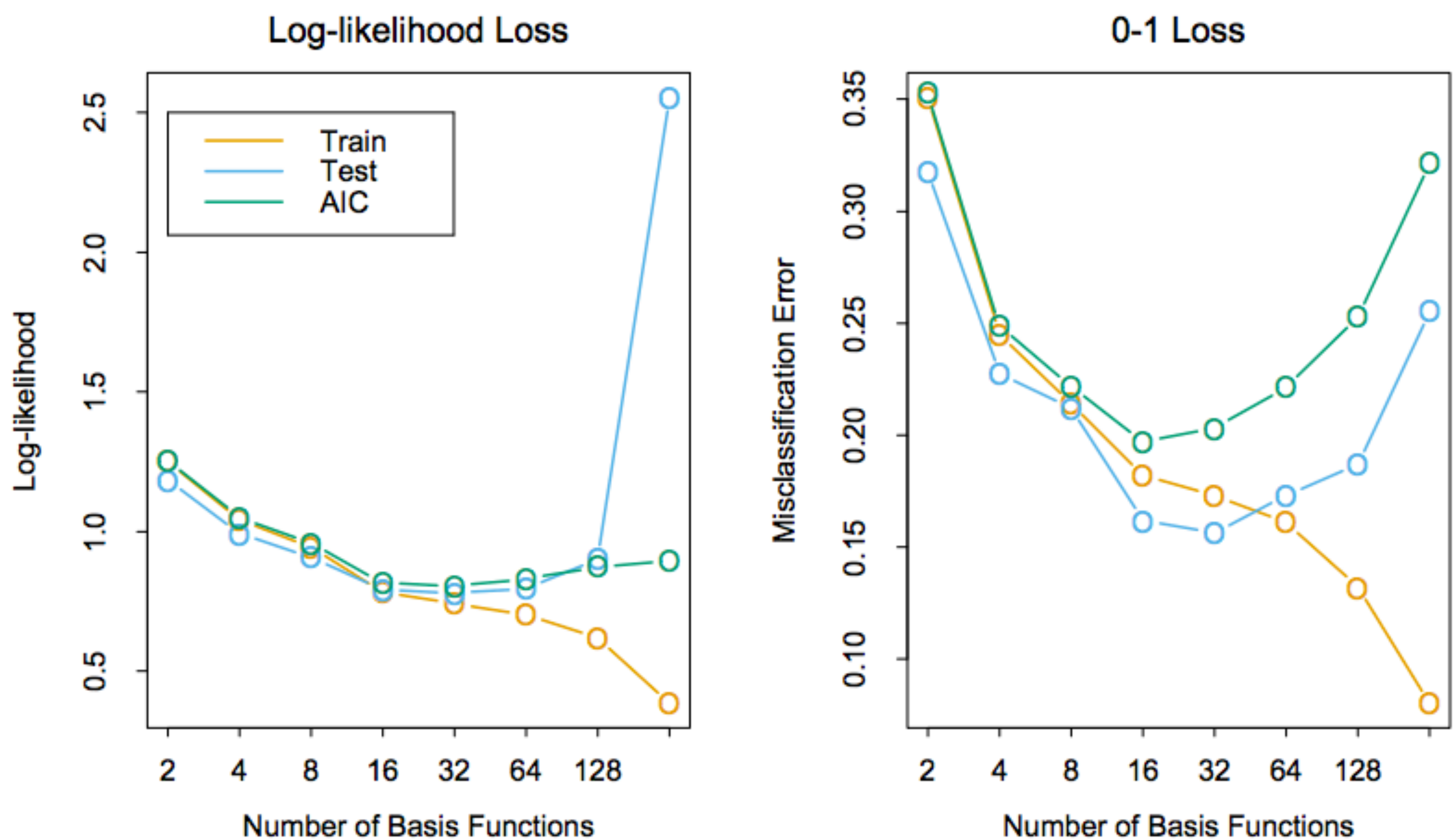


Figure 7.4 (Hastie et al.)

Bayesian Information Criterion (BIC)

- Applicable in settings with maximization of log-likelihood
- Also known as Schwarz criterion
- General form:

$$\mathrm{BIC} = d\log(N) - 2\log(\mathcal{L})$$

Linear Regression: AIC and BIC

- Criterion

$$\mathrm{AIC} = N \log \frac{SSE_d}{N} + 2d$$
$$\mathrm{BIC} = N \log \frac{SSE_d}{N} + d \log(N)$$

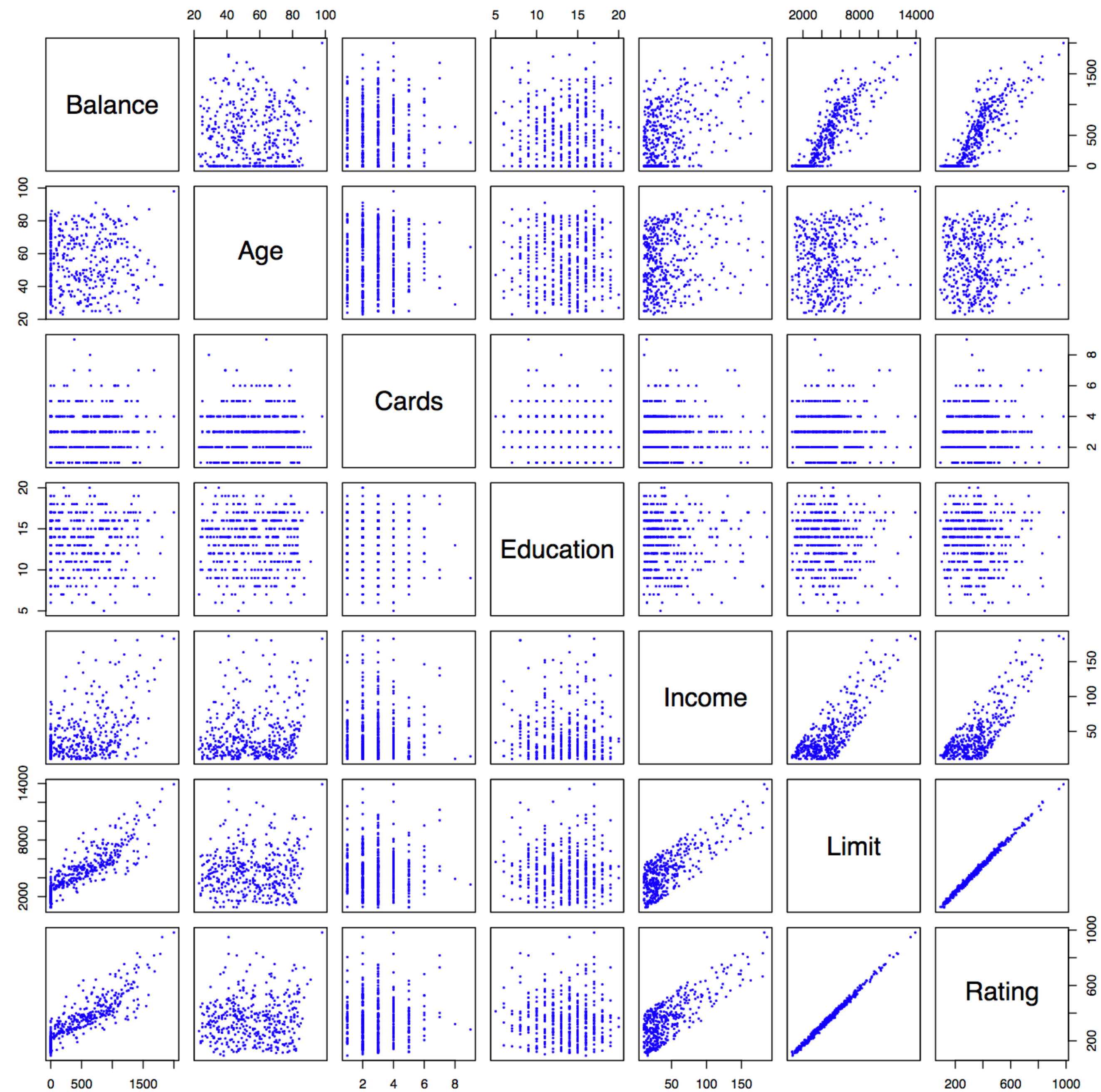
- What does this tell us about the two models?

AIC vs BIC

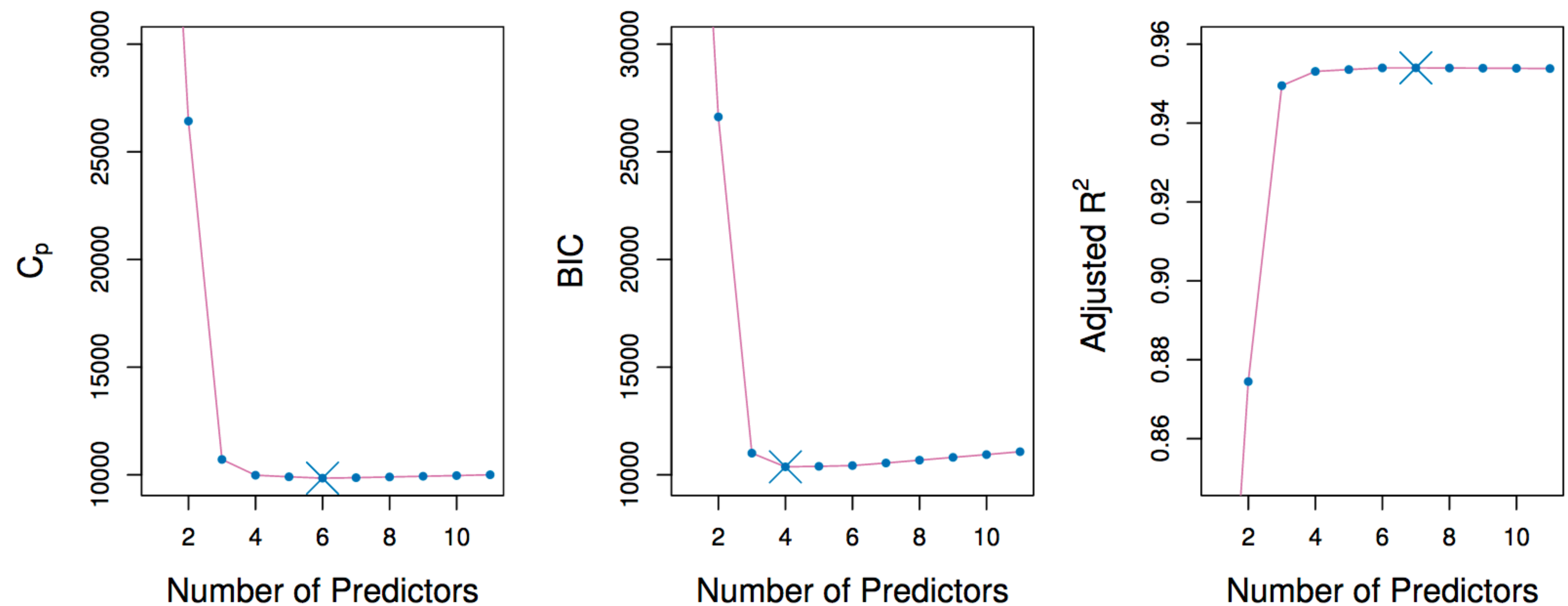
- BIC is asymptotically consistent
  - Probability BIC will select the correct model with large sample size approaches 1
- AIC favors complex models as N becomes large
- BIC chooses models that are too simple
- No clear choice between the two

Example: Credit Card Data

- Predicting credit card default
- Features: Balance, age, number of cards, education, income, credit card limit, credit rating

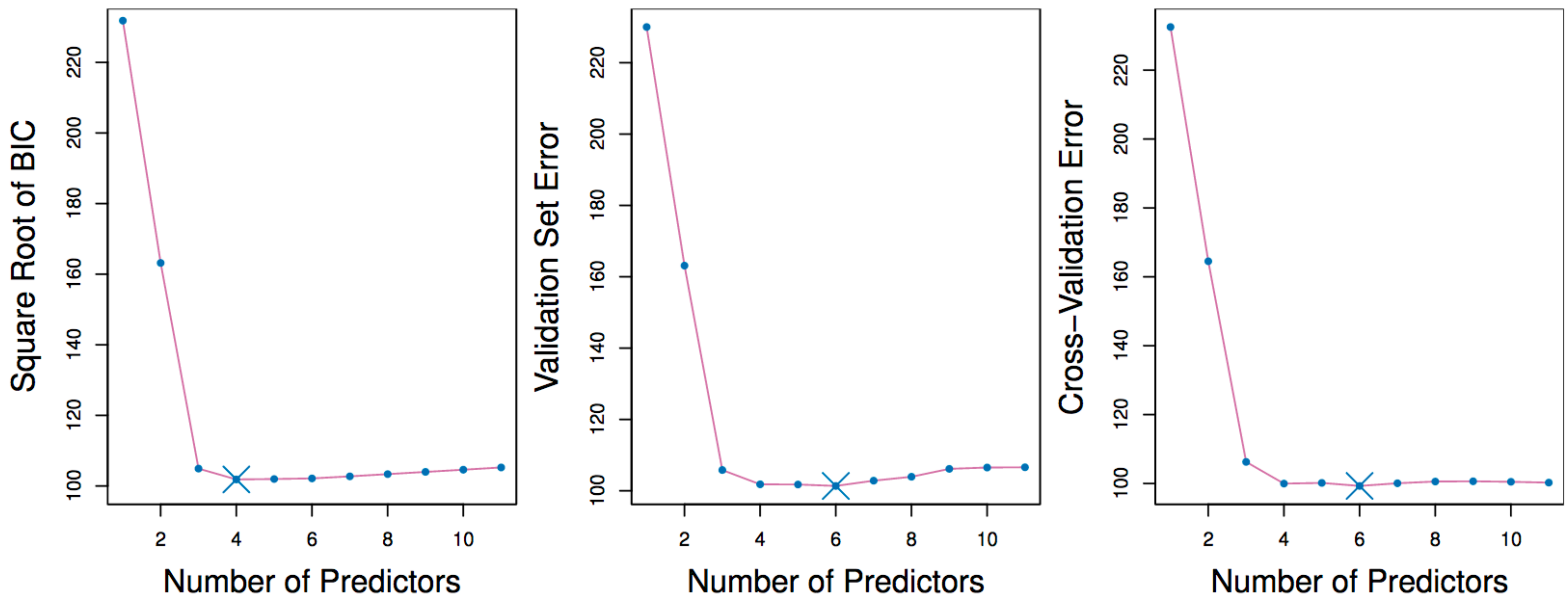


Example: Credit Card Data



https://lagunita.stanford.edu/c4x/HumanitiesandScience/StatLearning/asset/model_selection.pdf

Example: Credit Card Data



https://lagunita.stanford.edu/c4x/HumanitiesandScience/StatLearning/asset/model_selection.pdf

Minimum Description Length (MDL)

- Turn model selection into a communication / coding problem
- Idea: Best model should lead to best way to compress the available data
- Why does this make sense?

Data Compression Basics

- If we want to send a message z out of a possible m messages, what is the best way to encode it for the shortest code?
    - Example: If we use a binary code {0,1} and had only four messages, we could use {0, 10, 110, 111} — instantaneous prefix code
- We could imagine that we may want to use how often messages are being sent — shorter codes for more frequent messages

Shannon's Theorem

- Code lengths $l_i = -\log_2 P(z_i)$
- Average message length satisfies

entropy of distribution

$$E[\text{length}] \geq -\sum \Pr(z_i) \log_2(\Pr(z_i))$$

- Optimal lower bound on the best coding scheme

Classification as Coding

- Sender has access to training data (x_i, y_i), and needs communicate the labels to receiver
- Receiver has the examples but not the labels
- A perfect classifier will permit the receiver to reproduce the labels for the training examples

Minimum Description Length (MDL)

- MDL measures number of bits to encode a probability distribution
- MDL for model measures number of bits for the posterior distribution

$$\text{Length}(M) = -\log P(\mathbf{y}|\mathbf{X}, \mathbf{w}, M) - \log P(\mathbf{w}|M)$$

average code length for discrepancy between model and actual target values

average code length for transmitting model parameters

Minimum Description Length (MDL)

- Complex posterior distribution —> complex model
- Choose the model with the lowest MDL
- Can think of it as equivalent to preferring the best regularized model

Recall: Learning & VC Dimension

- VC dimension: Measures relevant size of hypothesis space
- Bound on generalization error

$$\epsilon(\hat{h}) \leq \left(\min_{h \in \mathcal{H}} \epsilon(h) \right) + O\left(\sqrt{\frac{VC(\mathcal{H})}{m} \log \frac{m}{VC(\mathcal{H})} + \frac{1}{m} \log \frac{1}{\delta}} \right)$$

Model Selection & VC Dimension

- Ideally select a model from a nested sequence of models of increasing VC dimensions
  $h_1 < h_2 < \dots$

- Model selection criterion: Find the model that achieves the lowest upper bound on the generalization error

$$\text{Expected error} \leq \text{Training error} + \text{Complexity penalty}$$

Structural Risk Minimization (SRM)

- Choose the hypothesis class that minimizes the upper bound on the expected error

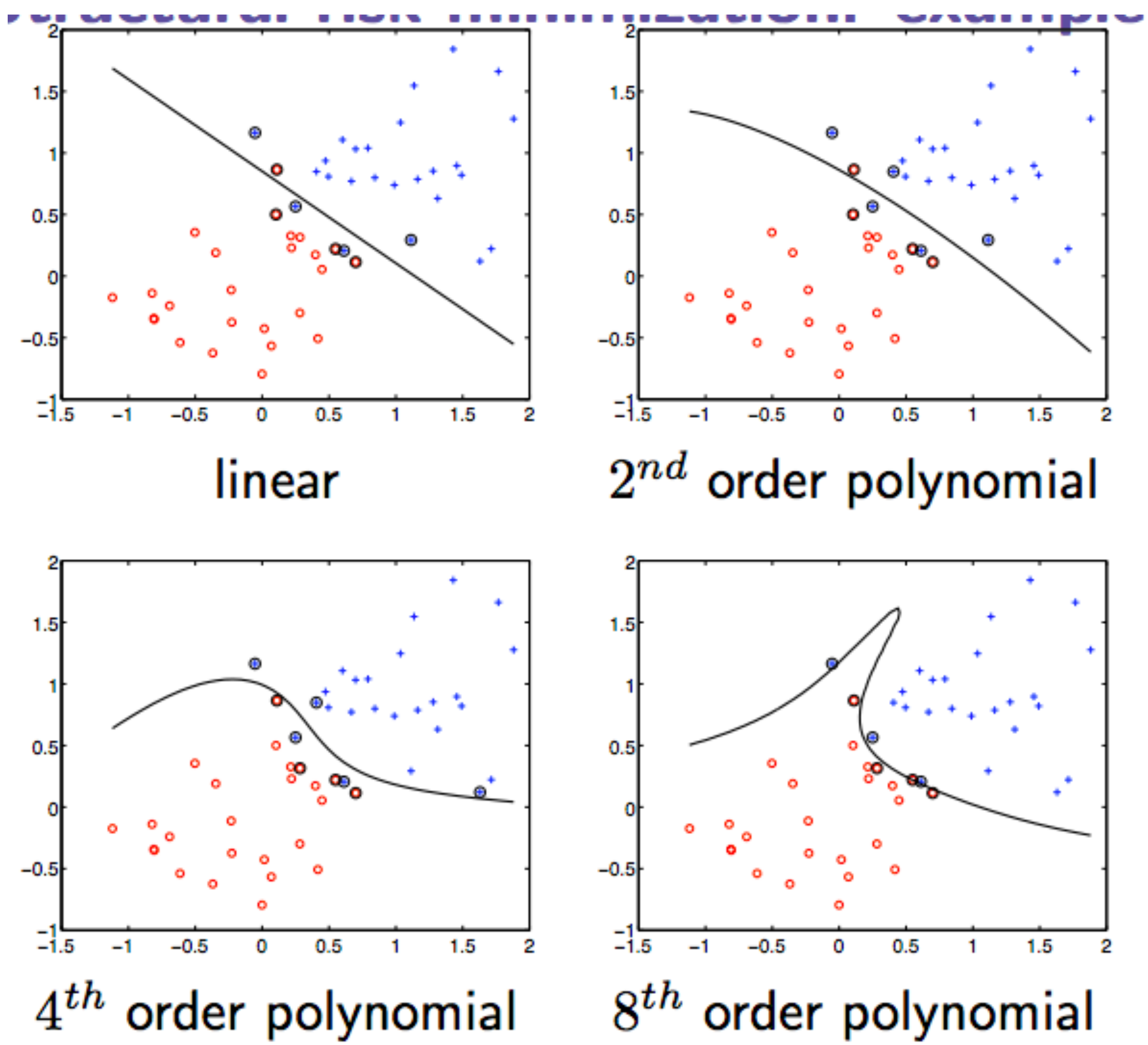
$$\epsilon(\hat{h}_i) \leq \hat{\epsilon}_N(\hat{h}_i) + \sqrt{\frac{\mathrm{VC}_i(\log(2N/\mathrm{VC}_i) + 1)) - \log(\delta/4)}{N}}$$

- Although upper bound can be loose, it can be good criteria for model selection

- Difficulty is calculating VC dimension

Example: SRM

Model 1 $K(\mathbf{x}_1, \mathbf{x}_2) = (1 + (\mathbf{x}_1^T \mathbf{x}_2))$
Model 2 $K(\mathbf{x}_1, \mathbf{x}_2) = (1 + (\mathbf{x}_1^T \mathbf{x}_2))^2$
Model 3 $K(\mathbf{x}_1, \mathbf{x}_2) = (1 + (\mathbf{x}_1^T \mathbf{x}_2))^3$
... ...



http://www.ai.mit.edu/courses/6.867-f04/lectures/lecture-12-ho.pdf

Example: SRM

- N = 50, delta = 0.005

Model	d_{VC}	Empirical fit	$\epsilon(n, d_{VC}, \delta)$
1^{st} order	3	0.06	0.5501
2^{nd} order	6	0.06	0.6999
4^{th} order	15	0.04	0.9494
8^{th} order	45	0.02	1.2849

- SRM would select linear model

http://www.ai.mit.edu/courses/6.867-f04/lectures/lecture-12-ho.pdf

Model Size Comparison

Plot of relative error in using chosen model versus the best model

$$100 \times \frac{\mathrm{Err}_{\mathcal{T}}(\hat{\alpha}) - \min_\alpha \mathrm{Err}_{\mathcal{T}}(\alpha)}{\max_\alpha \mathrm{Err}_{\mathcal{T}}(\alpha) - \min_\alpha \mathrm{Err}_{\mathcal{T}}(\alpha)}$$

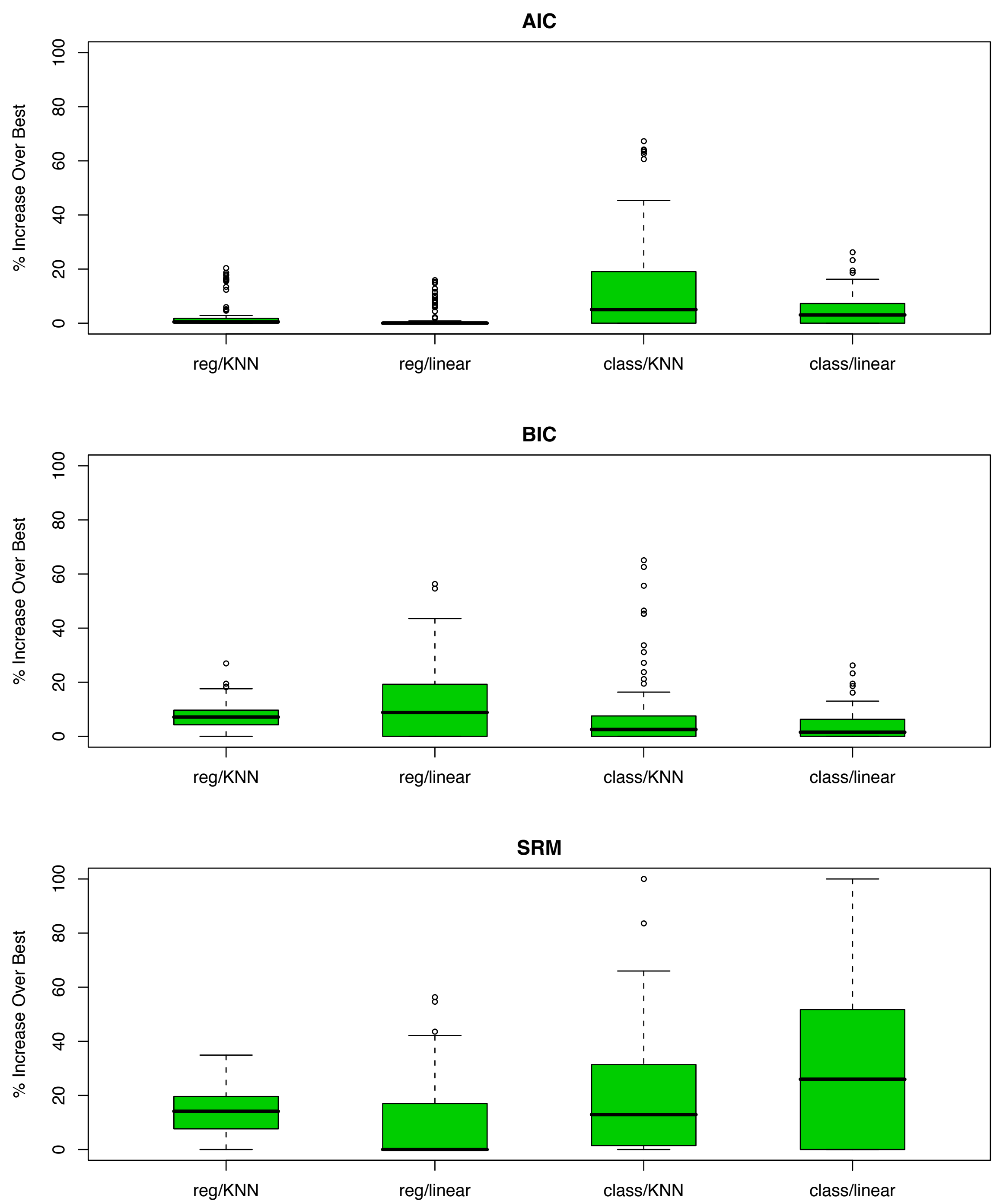


Figure 7.7 (Hastie et al.)

Revisiting Feature Selection

Why Feature Selection

- Some algorithms scale (computationally) poorly with increased dimension
- Irrelevant features can confuse some algorithms
- Redundant features adversely affect regularization
- Removal of features can increase generalization
- Reduction of data set and resulting model size

Feature Selection Methods

- Methods agnostic to the learning algorithm
  - Preprocessing based methods
  - Filter feature selection methods
- Wrapper methods (keep learning in loop)
  - Repeated runs of learner with different set of features
  - Can be computationally expensive

Filter Feature Selection

- Based on heuristics but much faster than wrapper methods

- Use statistical measure to assign a scoring to each feature

- Methods are often univariate and consider the feature independently, or with regard to the dependent variable.

Filter Feature Measures

- Correlation criteria: Rank features in order of their correlation with the labels

$$R(\mathbf{x}_d, \mathbf{y}) = \frac{\text{Cov}(\mathbf{x}_d, \mathbf{y})}{\sqrt{\text{Var}(\mathbf{x}_d)\text{Var}(\mathbf{y})}}$$

- Mutual information criterion: High mutual information means high relevance

$$MI(\mathbf{x}_d, \mathbf{y}) = \sum_{\mathbf{x}_d \in \{0,1\}} \sum_{\mathbf{y} \in \{-1,+1\}} P(\mathbf{x}_d, \mathbf{y}) \frac{\log P(\mathbf{x}_d, \mathbf{y})}{P(\mathbf{x}_d)P(\mathbf{y})}$$

Wrapper Method

- Forward and backward search (covered in linear regression lecture)
  - Greedily add / remove features
  - Inclusion / removal uses cross-validation
  - Can use any of the criterion covered earlier in class to determine when to stop

Measure of Uncertainty

- Suppose we have independent samples drawn from some population

$$x_1, \cdots, x_n \sim P_\theta$$

- We estimate our parameter of interest $\hat{\theta}$ (e.g., coefficient weights, etc)

- We want to know the variance of our parameter(s) or even construct approximate confidence intervals

- What if we can't make usual assumptions (e.g., normality)?

Bootstrap

Bootstrap Method

Metaphor for a "self-sustaining process that proceeds without external help"

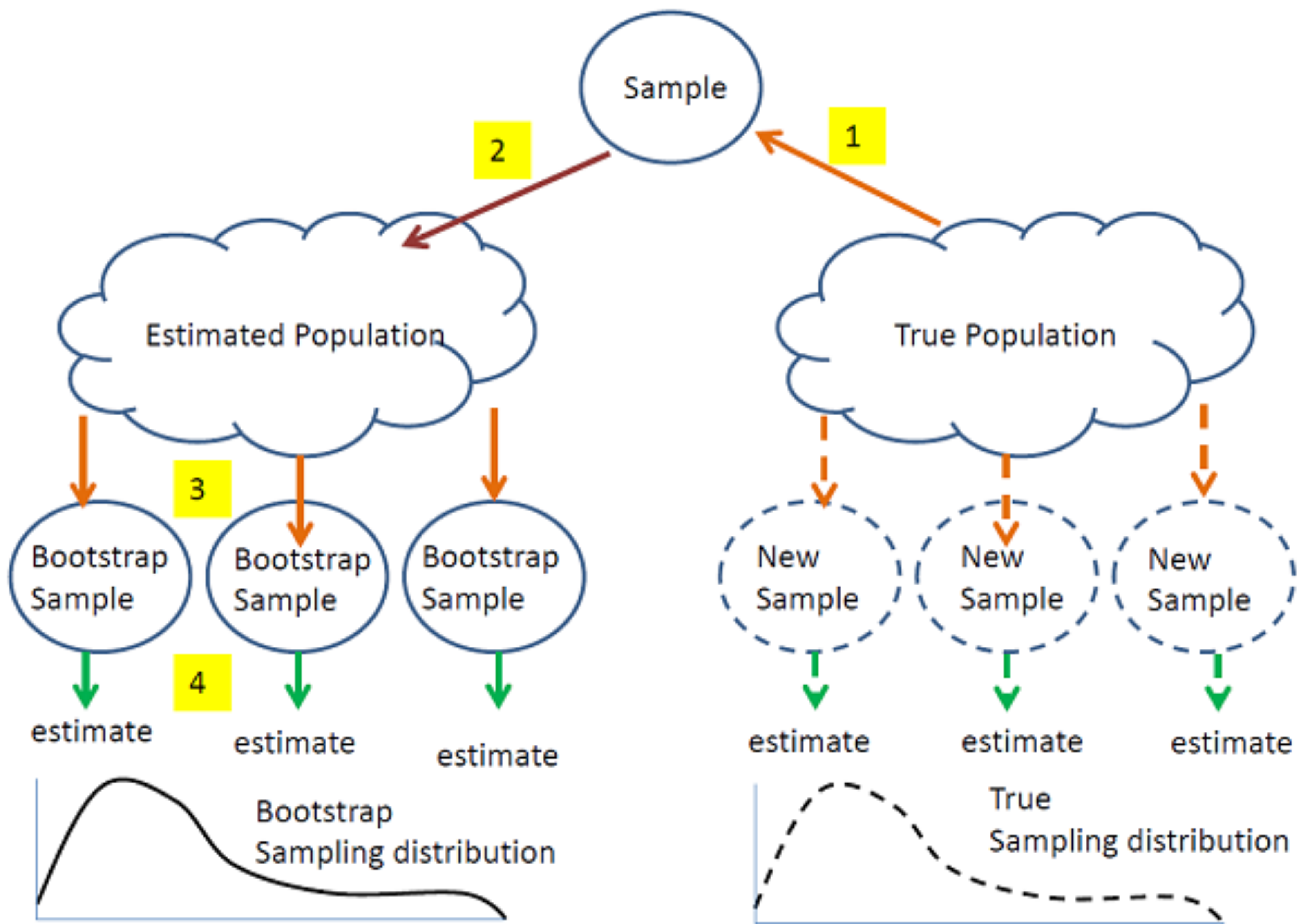
http://www.gmw.rug.nl/~huisman/sgs/2012_10_25_Bootstrap.pdf

Bootstrapping (Efron, 1979)

- Fundamental resampling tool in statistics
- General and most widely used tool to estimate measures of uncertainty associated with a given statistical model (e.g., confidence intervals, bias, variance, etc.)
- Resampling technique with replacement
- Distribution-independent or non-parametric

Bootstrap: Idea

"The population is to the sample as the sample is to the bootstrap samples"



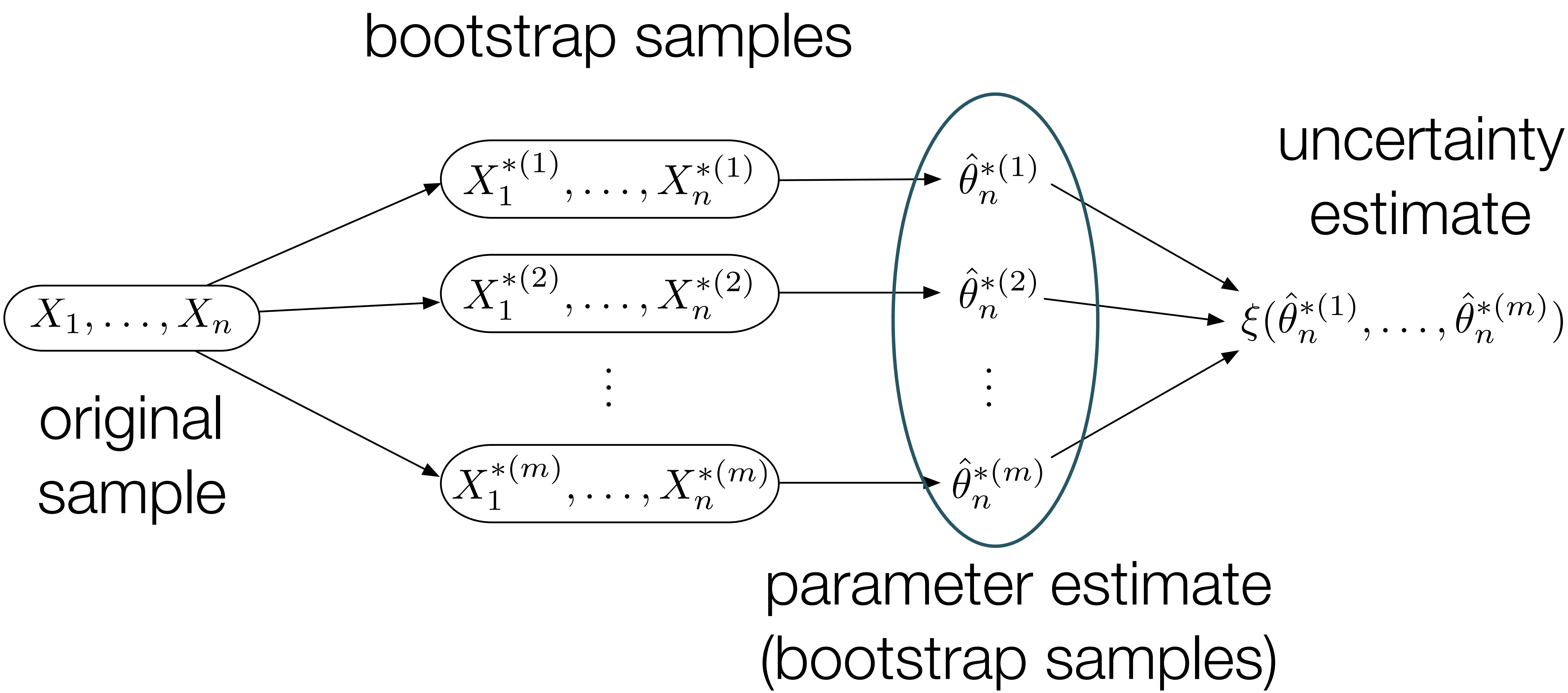
https://onlinecourses.science.psu.edu/stat555/node/119

Bootstrap Method: Uncertainty

Given a sample of size n

- Draw B samples of size n with replacement from the sample (bootstrap samples)
- Compute for each bootstrap sample the statistic of interest (e.g., learn the weights)
- Estimate the sample distribution of the statistic method by the bootstrap sample distribution

Bootstrap Method: Uncertainty



Bootstrap: Measuring Uncertainty

- Estimating standard errors

$$\mathrm{SE}(\hat{\theta}) = \sqrt{\frac{1}{B}\sum_{b=1}^{B}(\theta_b - \frac{1}{B}\sum_{r=1}^{B}\theta_r)^2}$$

- Estimating bias

$$E(\hat{\theta}) \approx \frac{1}{B}\sum_{b=1}^{B}(\theta_b - \hat{\theta})$$

- Estimating confidence

$$\mathbb{P}(2\hat{\theta} - q_{1-\alpha/2} \leq \theta \leq 2\hat{\theta} - q_{\alpha/2}) = 1 - \alpha$$

Bootstrap: Number of Points

- Sampling with replacement from N samples

$$\Pr(i \in B) = 1 - (1 - \frac{1}{N})^N$$
$$\approx 0.632$$

- Each bootstrap sample will contain roughly 63.2% of the original instances

Simple Example: Bootstrap

- Suppose that we wish to invest a fixed sum of money in two financial assets that yield returns of X and Y, where X and Y are random quantities

- Fraction of money in X with remaining in Y

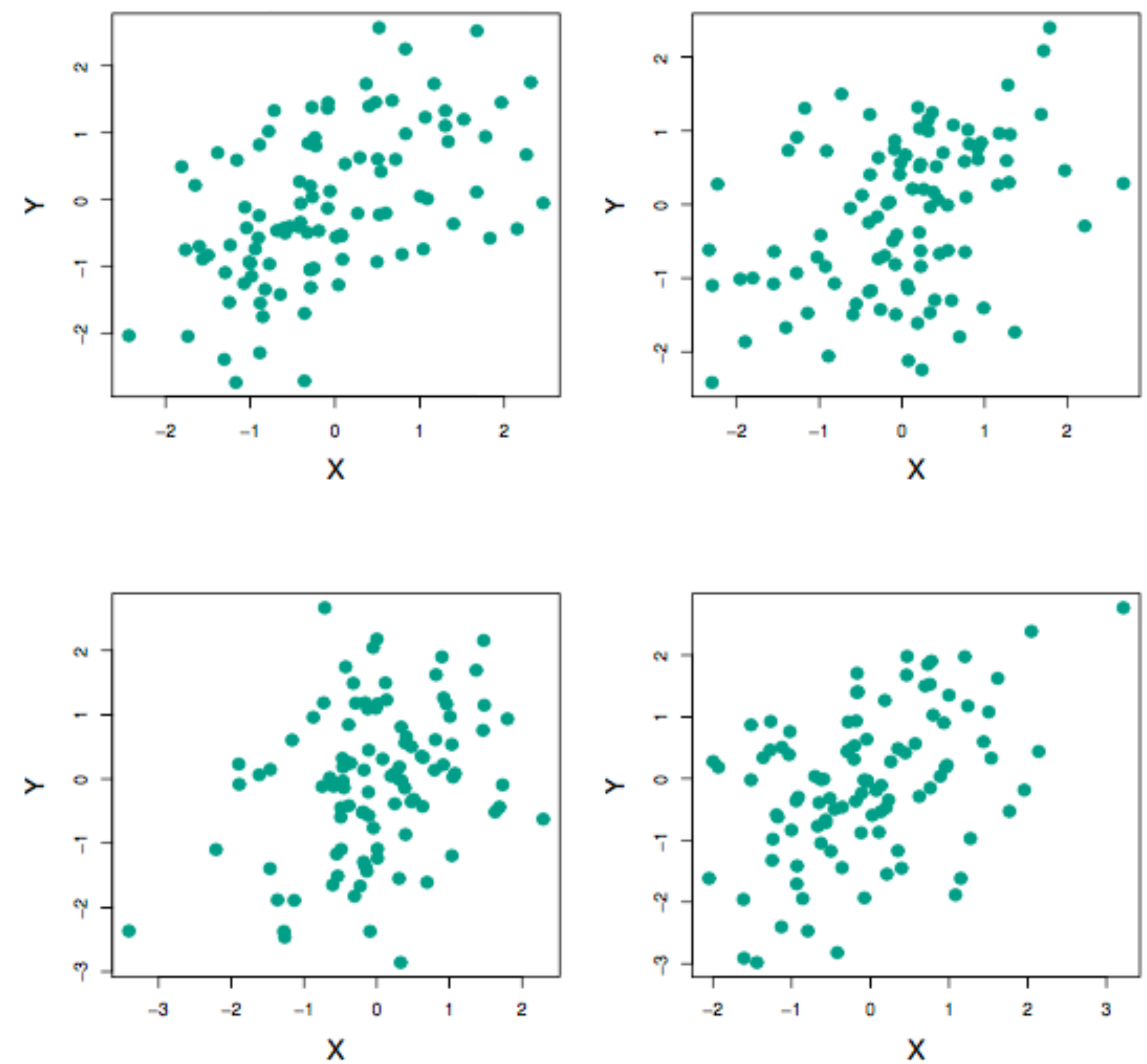
- We wish to choose the fraction to minimize the total risk (variance) of our investment

$$\mathrm{Var}(\alpha X + (1 - \alpha)Y)$$

Simple Example: Bootstrap

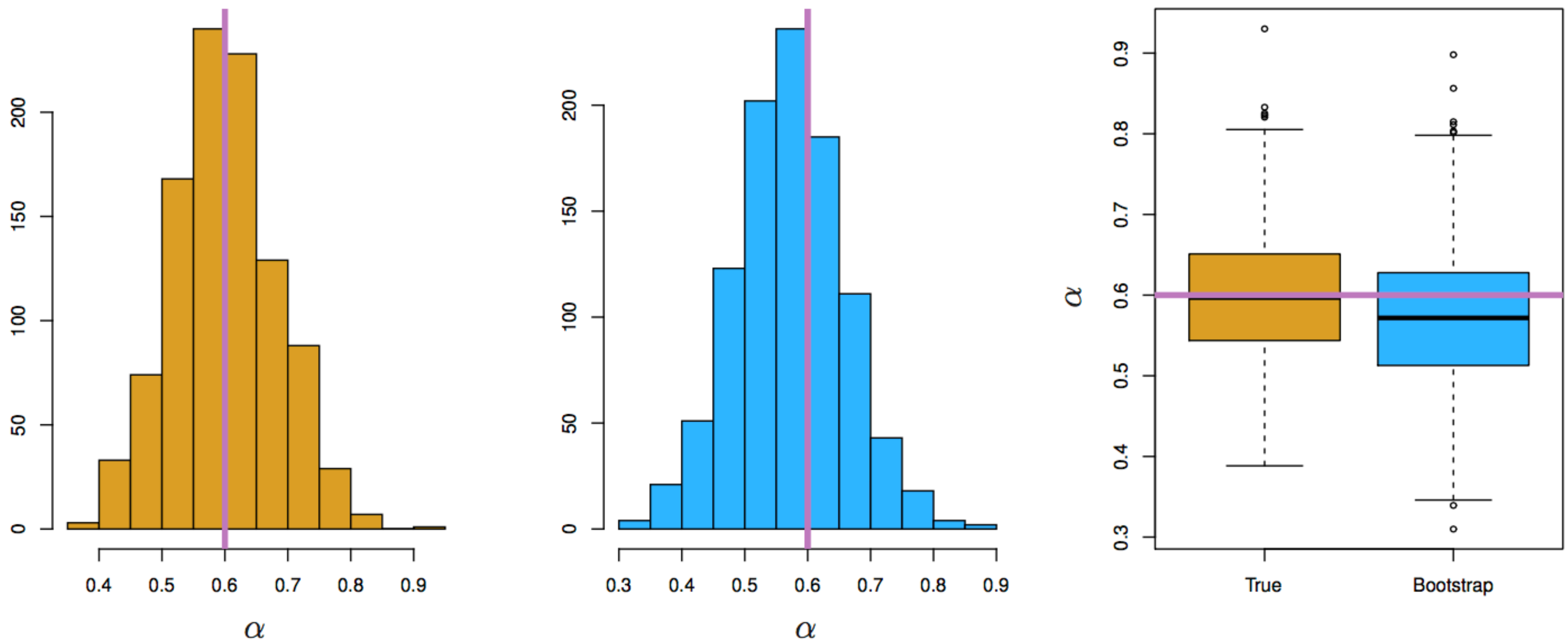
- Estimate variance and covariance for X, Y
- Estimated value that minimizes the variance of our investment

$$\hat{\alpha} = \frac{\hat{\sigma}_y^2 - \hat{\sigma}_{xy}}{\hat{\sigma}_x^2 + \hat{\sigma}_y^2 - 2\hat{\sigma}_{xy}}$$



https://lagunita.stanford.edu/c4x/HumanitiesandScience/StatLearning/asset/cv_boot.pdf

Simple Example: Bootstrap



https://lagunita.stanford.edu/c4x/HumanitiesandScience/StatLearning/asset/cv_boot.pdf

Example: Bootstrap Splines

Estimated

Bootstrap

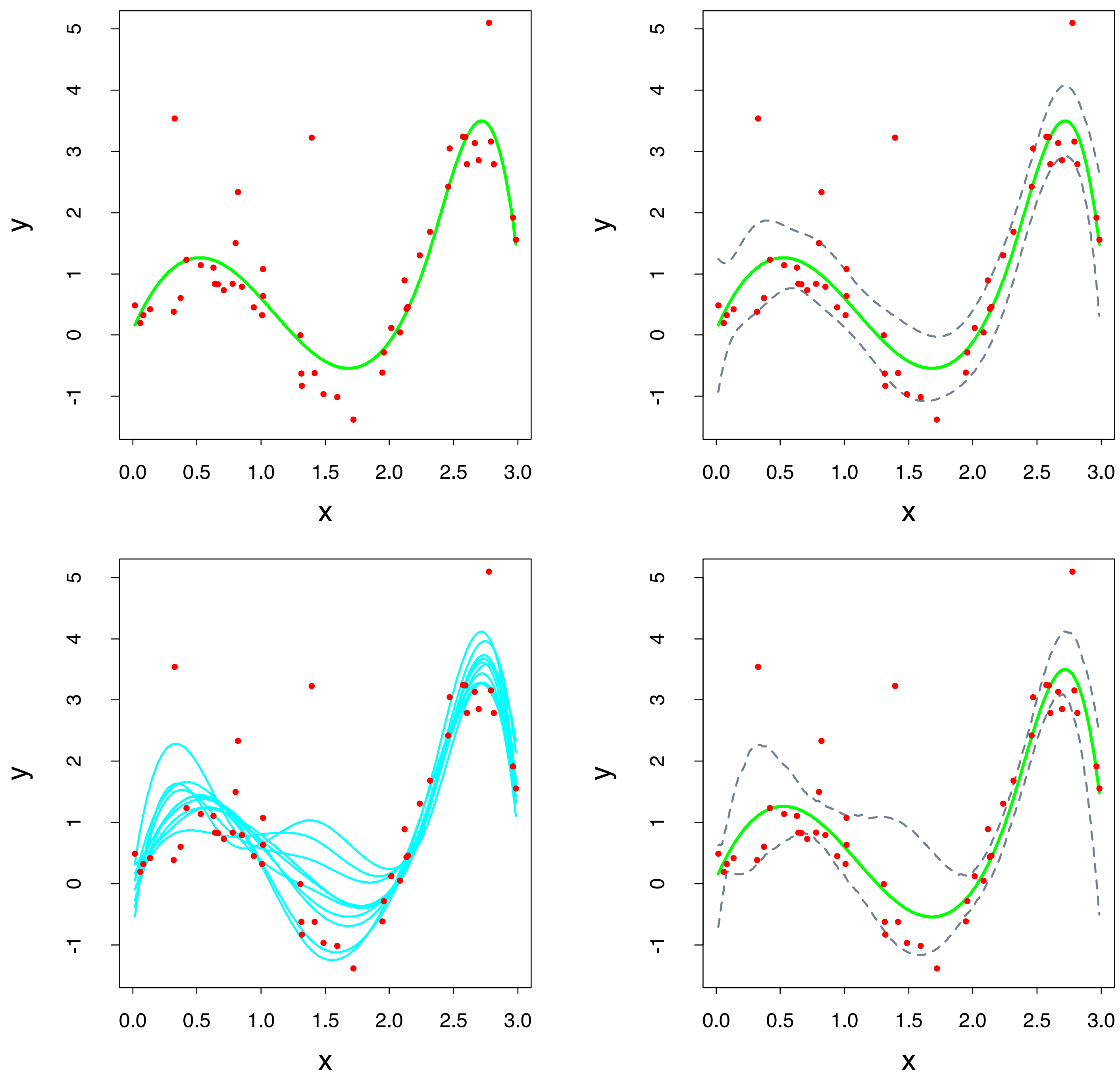
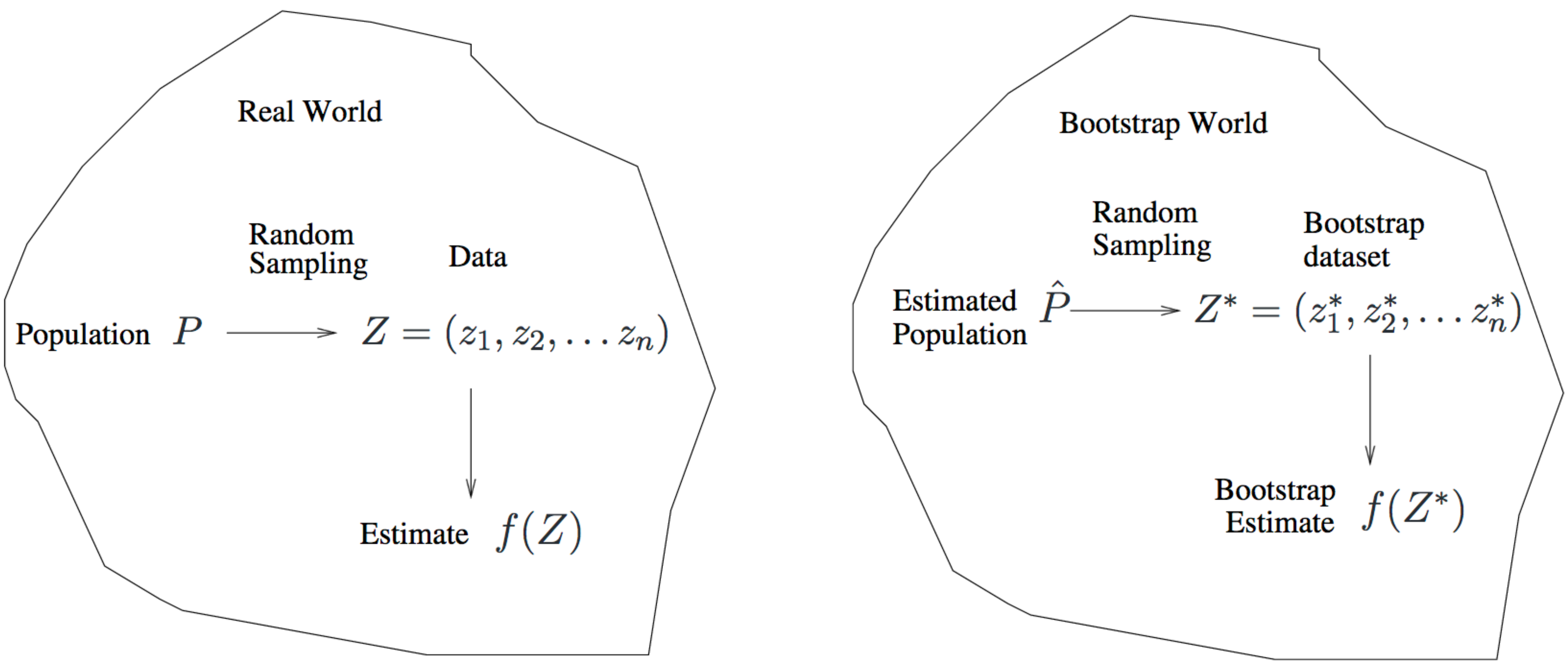


Figure 8.2 (Hastie et al.)

Bootstrap: General



https://lagunita.stanford.edu/c4x/HumanitiesandScience/StatLearning/asset/cv_boot.pdf

Bootstrap Properties

- Simple and straightforward to derive estimates of standard errors and confidence intervals for complex estimators

- Asymptotically consistent (under certain conditions)

- In more complex data situations, bootstrapping may not be easy

    - Example: time series data — how to deal with sampling with replacement?

Bootstrap for Prediction Error

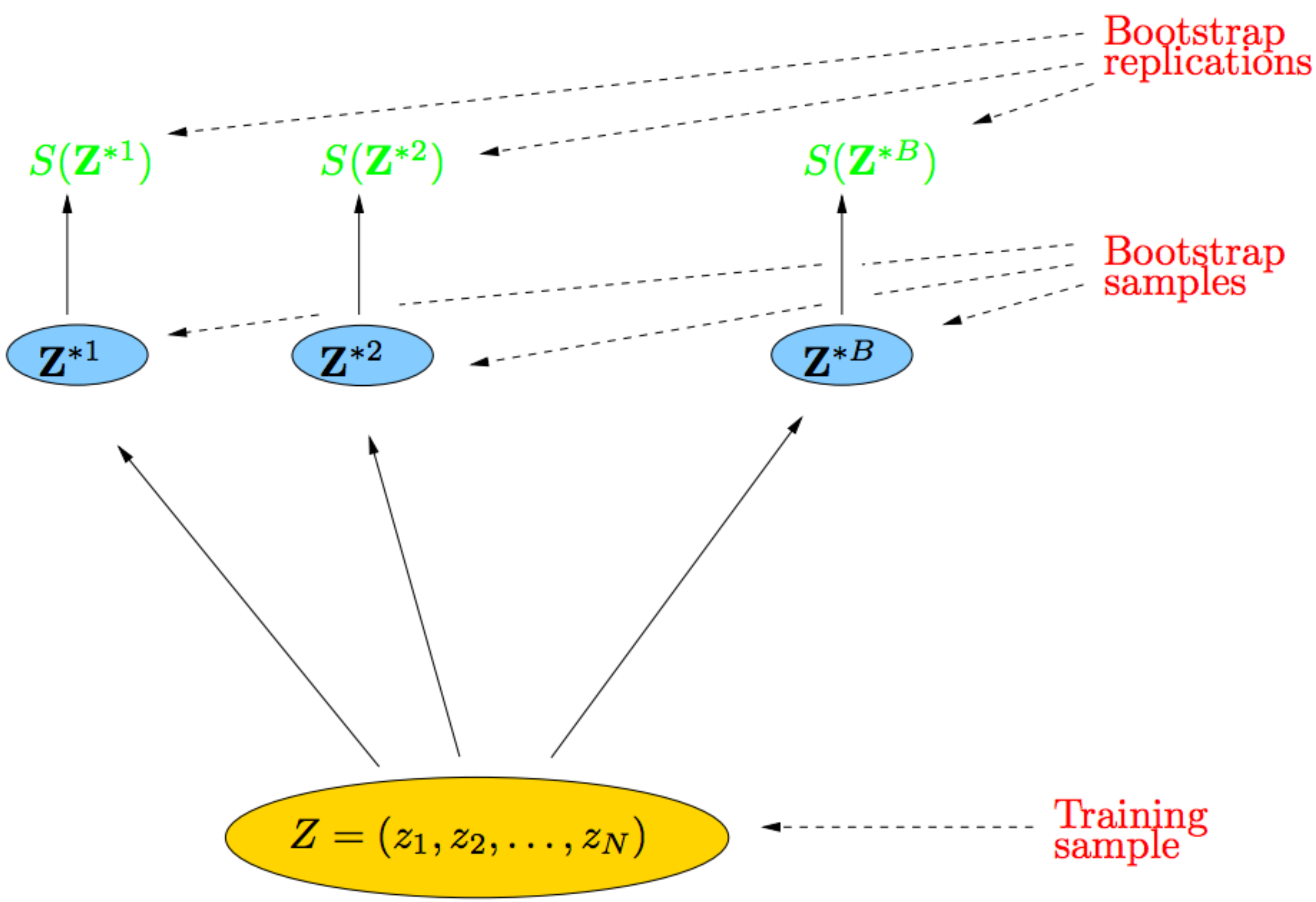


Figure 7.12 (Hastie et al.)

Bootstrapping for Prediction Error

- Fit model in question on a set of bootstrap samples
- Keep track of how well it predicts on the original training set
- Estimate of in-sample error

$$\overline{\mathrm{Err}}_{\mathrm{boot}} = \frac{1}{B}\frac{1}{N}\sum_{b}\sum_{i} L(y_i, \hat{f}^{*b}(\mathbf{x}_i))$$

Anything wrong with this?

Leave-one-out Bootstrap

https://sebastianraschka.com/blog/2016/model-evaluation-selection-part2.html

Leave-one-out Bootstrap

- For each observation, keep track of predictions from bootstrap samples not containing that observation

$$\overline{\mathrm{Err}}^{(1)} = \frac{1}{N} \sum_{i} \frac{1}{|C^{-i}|} \sum_{b \in C^{-i}} L(y_i, \hat{f}^{*b}(\mathbf{x}_i))$$

- Solves overfitting problem from before

- What is downside? (Hint: how many samples)

“0.632 Estimator”

- Corrects the bias of LOO bootstrap error

$$\overline{\text{Err}}^{(.632)} = 0.368\text{TrainErr} + 0.632\overline{\text{Err}}^{(1)}$$

- Works well in “light” (under) fitting scenarios

- Account for the overfitting by taking into account “no-information error rate” — when inputs and class labels are independent

“0.632+ Estimator”

- No-information error rate

$$\gamma = \sum_{\ell} \hat{p}_\ell (1 - \hat{q}_\ell)$$

- Relative overfitting rate

$$\hat{R} = \frac{\overline{\mathrm{Err}}^{(1)} - \mathrm{TrainErr}}{\hat{\gamma} - \mathrm{TrainErr}}$$

- New estimator

$$\overline{\mathrm{Err}}^{(.632+)} = (1 - \hat{w})\mathrm{TrainErr} + \hat{w}\overline{\mathrm{Err}}^{(1)}, \ \hat{w} = \frac{0.632}{1 - 0.368\hat{R}}$$

Bootstrap for Prediction Error

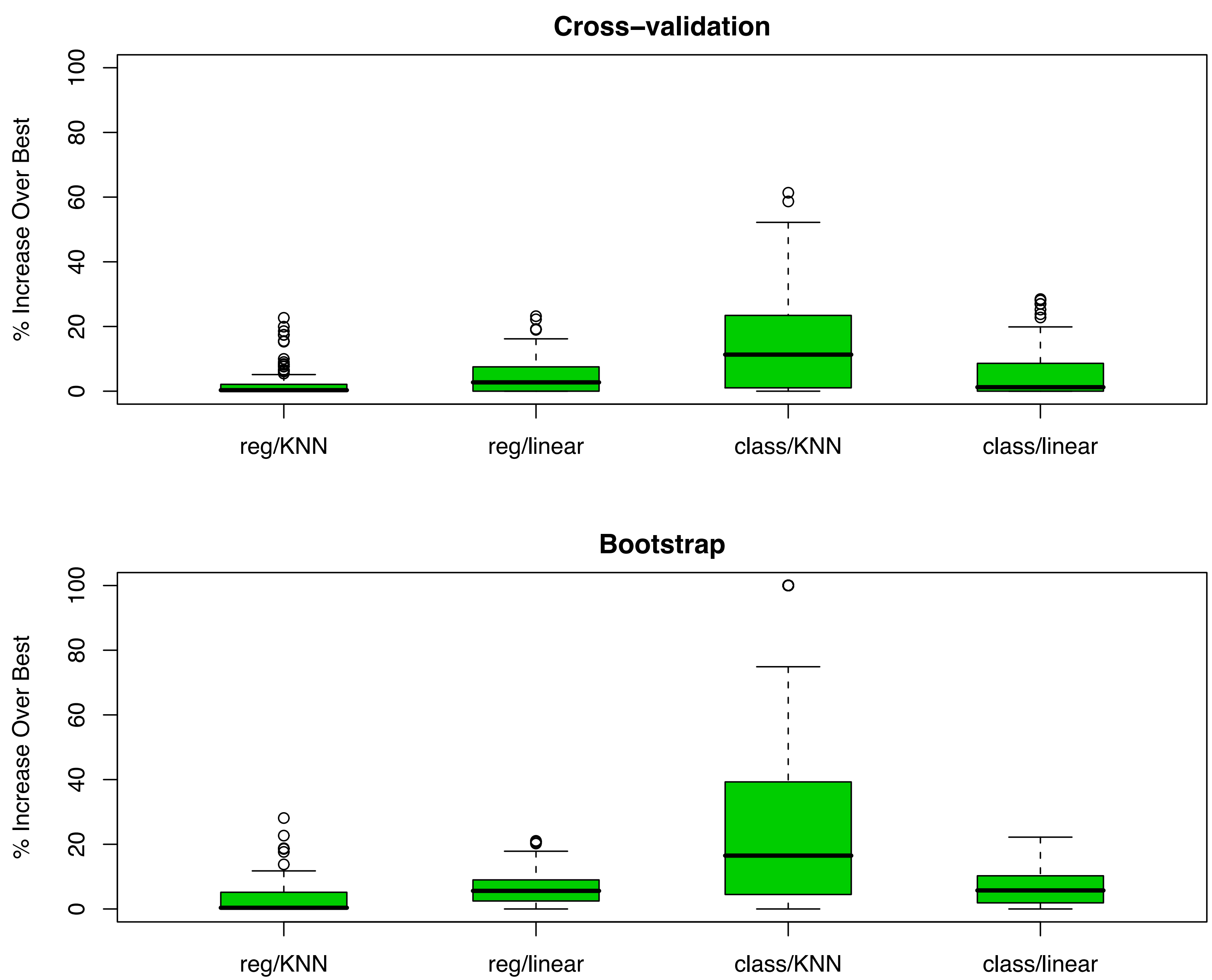


Figure 7.13 (Hastie et al.)

Bootstrap vs Cross-Validation

- Cross validation sacrifices dataset size to estimate error
- Bootstrapping approaches error estimation by resampling our dataset to its original size
- Average over performance in these resampled datasets to estimate performance on future unseen data