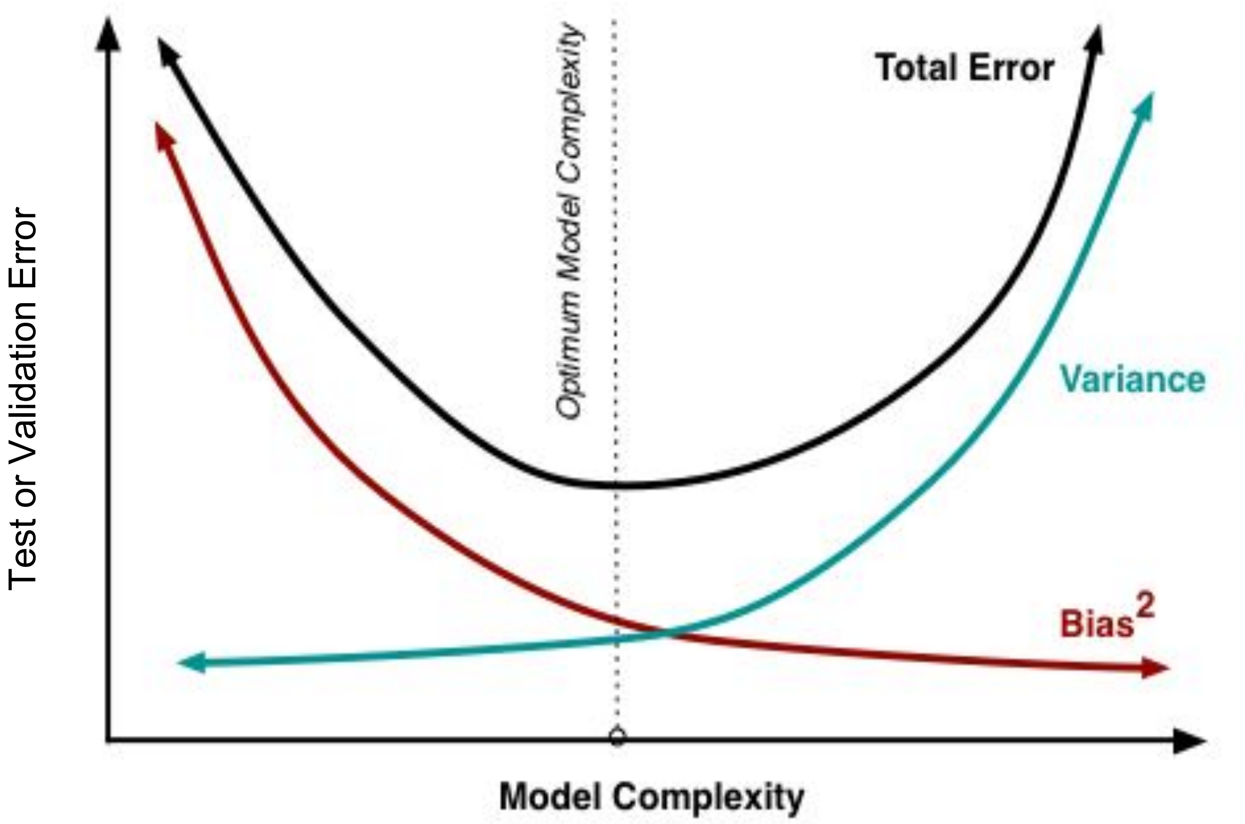
Cross Validation

Learning Objectives

- Motivate the use of cross validation
- Describe how to do it (especially k-fold cross validation)
- In the individual assignment, code it from scratch
- Contrast cross validation with alternative (complementary) procedures for model comparison

Recall the Bias-Variance Tradeoff



How to Mitigate Overfitting

1. **Get more data…**

2. **Subset Selection...** keep only a subset of model features (i.e, dimensions)

3. **Regularization...** restrict magnitude of model features (i.e. restrict parameter space)

4. **Dimensionality Reduction…** project data into a lower dimensional space (later in course)

5. **Cross Validation…** train many models on random subsets of data and choose best parameters

Subset Selection

Best subset: Try every model. Every possible combination of *p* predictors

- Computationally intensive. 2^p possible subsets of *p* predictors
- High chance of finding a "good" model by random chance.

… A sort-of monkeys-Shakespeare situation …

Stepwise: Iteratively pick predictors to be in/out of the final model.

- Forward, backward, forward-backward strategies
    - Forward: starting with just one and adding more features, one-by-one
    - Backward: starting with them all, and removing one-by-one
- Sklearn features only backward recursive elimination.

Model Selection with Complexity Penalty

$$C_p = \frac{1}{n}(RSS + 2\underline{p}\hat{\sigma}^2)$$

Mallow's Cp

p is the total # of parameters

$\hat{\sigma}^2$ is an estimate of the variance of the error, ε

$$AIC = -2logL + 2 \cdot \underline{p}$$

L is the maximized value of the likelihood function for the model estimated

$$BIC = \frac{1}{n}(RSS + log(n)\underline{p}\hat{\sigma}^2)$$

This is Cp, except 2 is replaced by log(n). log(n) > 2 for n>7, so BIC generally exacts a heavier penalty for more variables

$$Adjusted\ R^2 = 1 - \frac{RSS/(n-\underline{p}-1)}{TSS/(n-1)}$$

Similar to R^2, but pays price for more variables

Side Note: Can show AIC and Mallow's Cp are equivalent for linear case

Model Selection with Complexity Penalty

$$C_p = \frac{1}{n}(RSS + 2\underline{p}\hat{\sigma}^2)$$

Mallow's Cp

p is the total # of parameters

$\hat{\sigma}^2$ is an estimate of the variance of the error, ε

$$AIC = -2logL + 2 \cdot \underline{p}$$

L is the maximized value of the likelihood function for the model estimated

$$BIC = \frac{1}{n}(RSS + log(n)\underline{p}\hat{\sigma}^2)$$

This is Cp, except 2 is replaced by log(n). log(n) > 2 for n>7, so BIC generally exacts a heavier penalty for more variables

$$Adjusted\ R^2 = 1 - \frac{RSS/(n-\underline{p}-1)}{TSS/(n-1)}$$

Similar to R^2, but pays price for more variables

Side Note: Can show AIC and Mallow's Cp are equivalent for linear case

Cross Validation

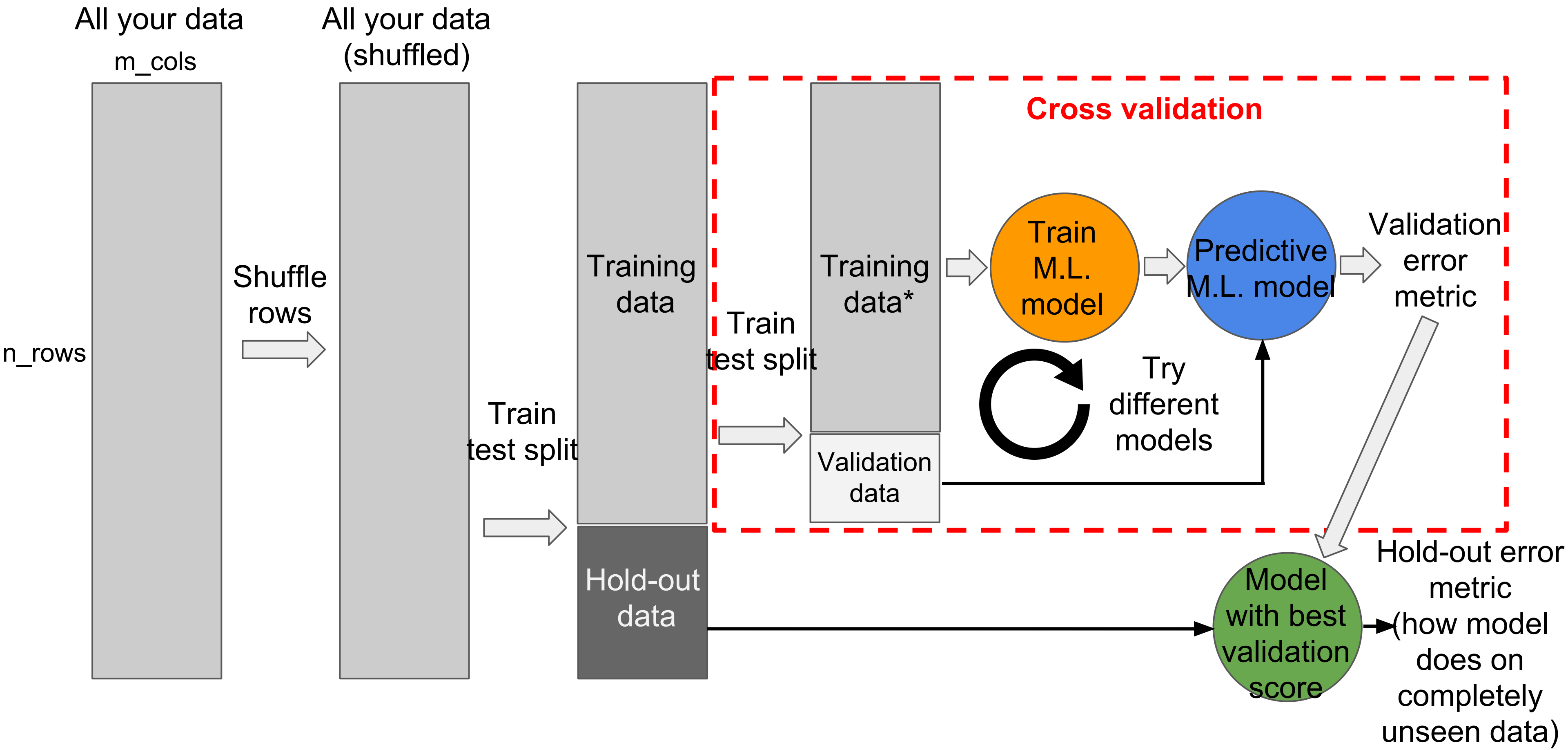
“Cross-validation ... is a model validation technique for assessing how the results of a statistical analysis will generalize to an independent data set.”
- Wikipedia

We use cross-validation for two things:

1) Attempting to quantify how well a model (of some given complexity) will predict on an unseen data set
2) Tuning hyperparameters of models to get best predictions.

Scikit-learn tangent (hyperparameters?)

Cross Validation - illustrated



Cross Validation - enumerated

1. Split your data (after splitting out hold-out set) into training/validation sets.

   70/30, 80/20 or 90/10 splits are commonly used

2. Use the training set to train several models of varying complexity.

   e.g. linear regression (w/ and w/out interaction features), neural nets, decision trees, etc.

3. Evaluate each model using the validation set.

   calculate R^2, MSE, accuracy, or whatever you think is best

4. Keep the model that performs best over the **validation** set.

People use different terms for the splits.

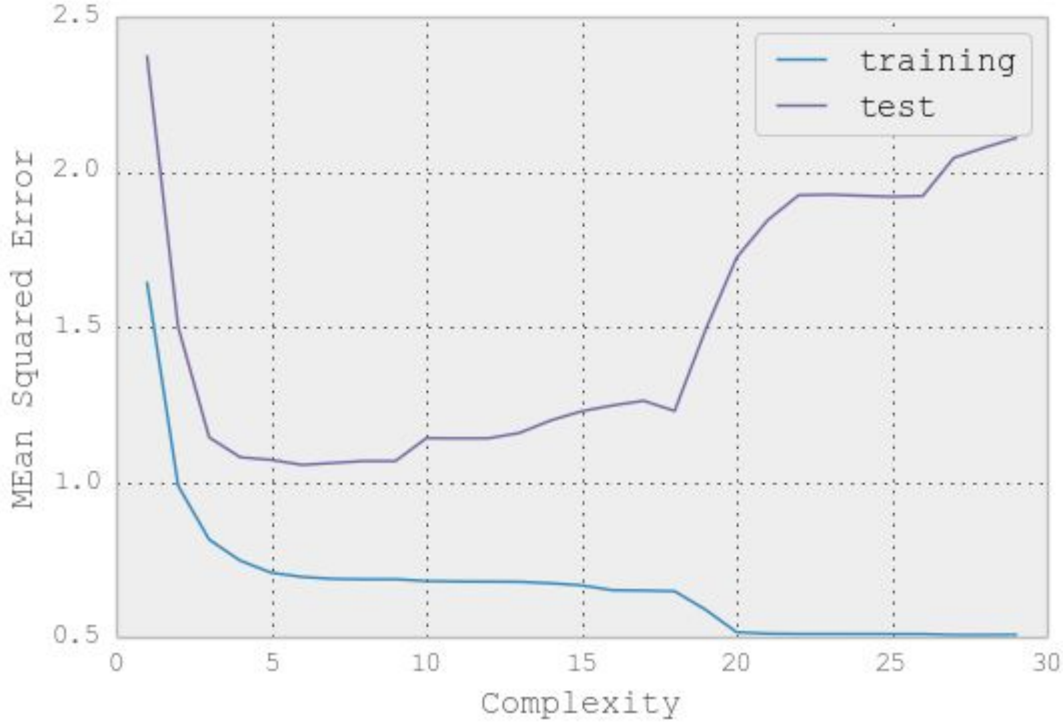
All data -> Train, Test, Hold-out

All data -> Train, Validate, Test

All data -> Train, Validate, Hold-out

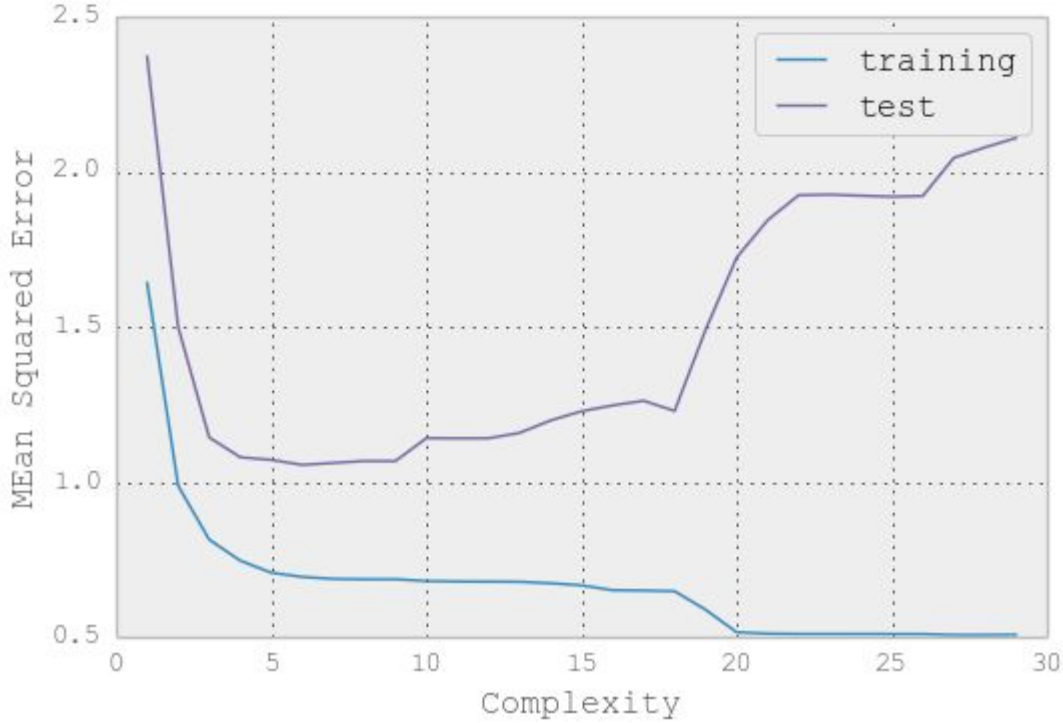
All the same idea.

Cross Validation - visualized



Number of features, interaction between features, order of features

Cross Validation - visualized



Number of features, interaction between features, order of features

Which error is most important to minimize? Why?

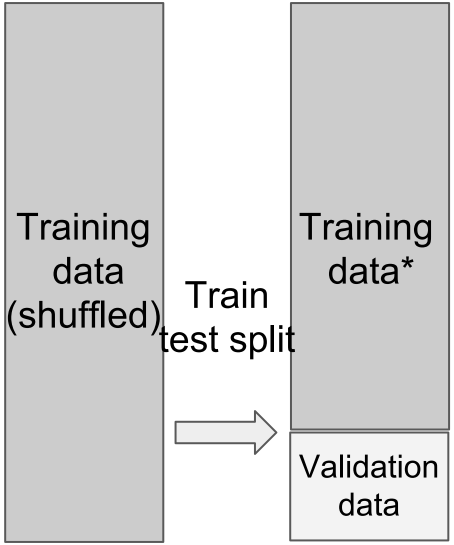
What model complexity is best? How did you decide?

At the optimum model complexity, what is the bias? What is the variance?

Is it possible for the test error to be lower than the training error?

Cross Validation - options

A single train-test split



Training data (shuffled)

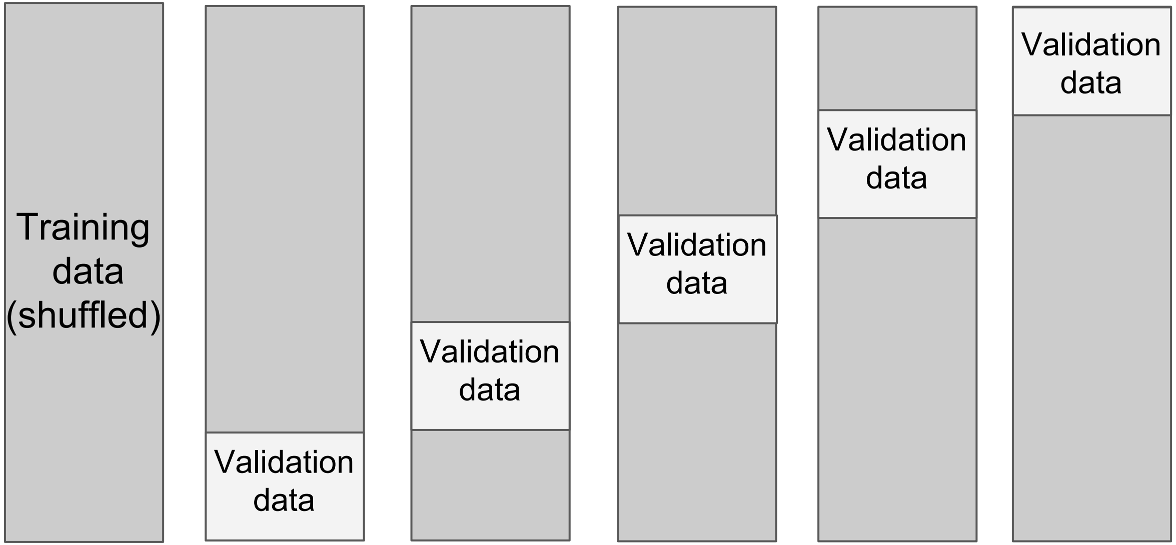
Train test split

Training data*

Validation data

After training get only 1 estimate of validation error (what if validation data very different from training data by chance?!?)

k-fold (showing k=5)



Training data (shuffled)

Validation data

Validation data

Validation data

Validation data

Validation data

After training get k estimates of validation error from the same model complexity, so calculate the mean validation error from those five estimates. This gives a more robust, less variable estimate.

Special case of k-fold: k = n (Leave one out CV).
Models are highly correlated in LOOCV.

Recap Learning Objectives

- Describe why cross validation is used
- Describe how to do it (especially k-fold cross validation)
- In the individual assignment, code it from scratch
- Aside: introduce feature selection/elimination
- Contrast cross validation with other statistical model comparison methods