Model Selection

Héctor Corrada Bravo

University of Maryland, College Park, USA

Fannie Mae: 2017-07-14

Model Selection

Let's revisit our discussion about model evaluation based on expected predicted error.

Model Selection

Let's revisit our discussion about model evaluation based on expected predicted error.

How do we measure our models' ability to predict unseen data, when we only have access to training data?

Cross-validation

The most common method to evaluate model **generalization** performance is cross-validation.

It is used in two essential data analysis phases: Model Selection and Model Assessment.

Cross-validation

Model Selection

Decide what kind, and how complex of a model we should fit.

Cross-validation

Model Selection

Decide what kind, and how complex of a model we should fit.

Consider a regression example: I will fit a linear regression model, what predictors should be included?, interactions?, data transformations?

Cross-validation

Model Selection

Decide what kind, and how complex of a model we should fit.

Consider a regression example: I will fit a linear regression model, what predictors should be included?, interactions?, data transformations?

Another example is what classification tree depth to use.

Cross-validation

Model Selection

Decide what kind, and how complex of a model we should fit.

Consider a regression example: I will fit a linear regression model, what predictors should be included?, interactions?, data transformations?

Another example is what classification tree depth to use.

Which kind of algorithm to use, linear regression vs. decision tree vs. random forest

Cross-validation

Model Assessment

Determine how well does our selected model performs as a **general** model.

Cross-validation

Model Assessment

Determine how well does our selected model performs as a **general** model.

Ex. I've built a linear regression model with a specific set predictors. How well will it perform on unseen data?

Cross-validation

Model Assessment

Determine how well does our selected model performs as a **general** model.

Ex. I've built a linear regression model with a specific set predictors. How well will it perform on unseen data?

The same question can be asked of a classification tree of specific depth.

Cross-validation

Cross-validation is a resampling method to obtain estimates of **expected prediction error rate** (or any other performance measure on unseen data).

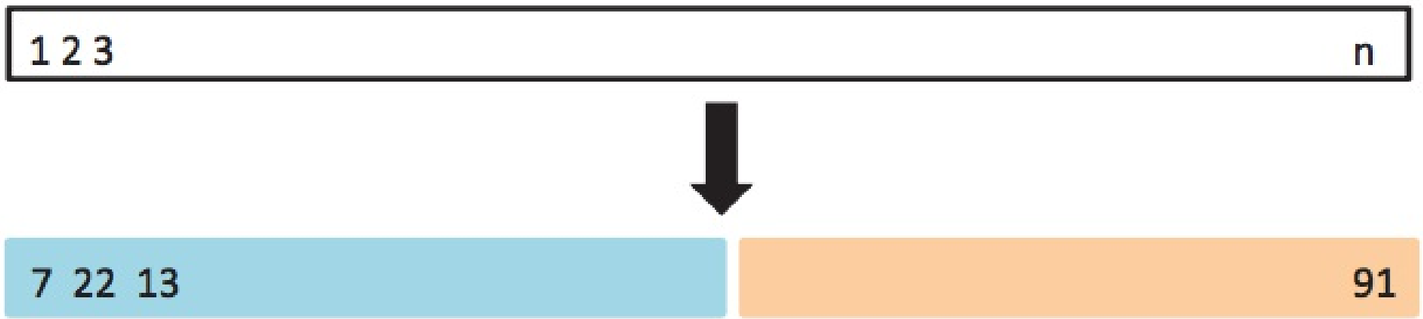
In some instances, you will have a large predefined test dataset **that you should never use when training**.

In the absence of access to this kind of dataset, cross validation can be used.

Validation Set

The simplest option to use cross-validation is to create a validation set, where our dataset is **randomly** divided into training and validation sets.

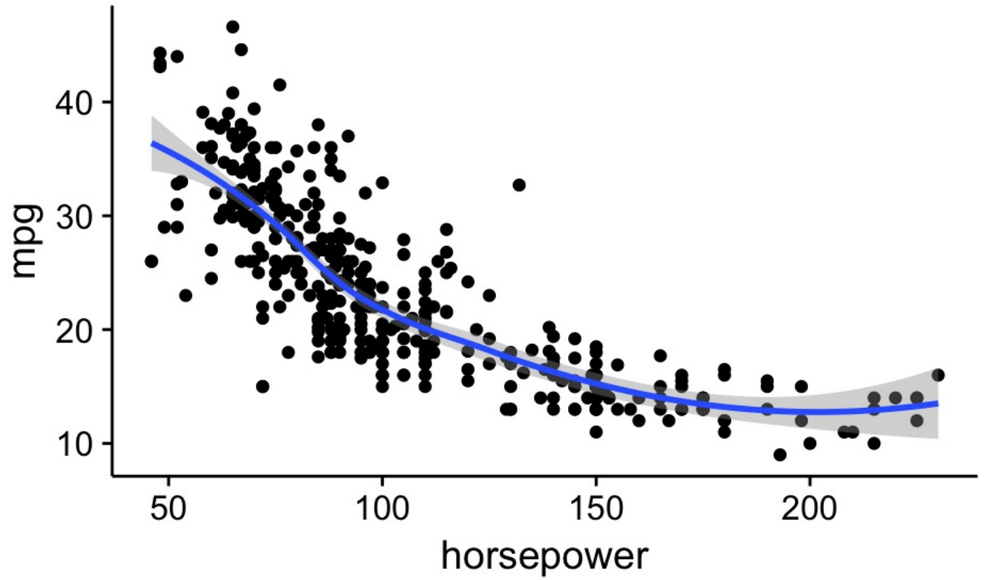
Then the validation is set aside, and not used at until until we are ready to compute **test error rate** (once, don't go back and check if you can improve it).



Validation Set

Let's look at our running example using automobile data, where we want to build a regression model to predict miles per gallon given other auto attributes.

A linear regression model was not appropriate for this dataset. Use polynomial regression as an illustrative example.



Validation Set

For polynomial regression, our regression model (for a single predictor X) is given as a d degree polynomial.

$$\mathbb{E}[Y|X = x] = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_d x^d$$

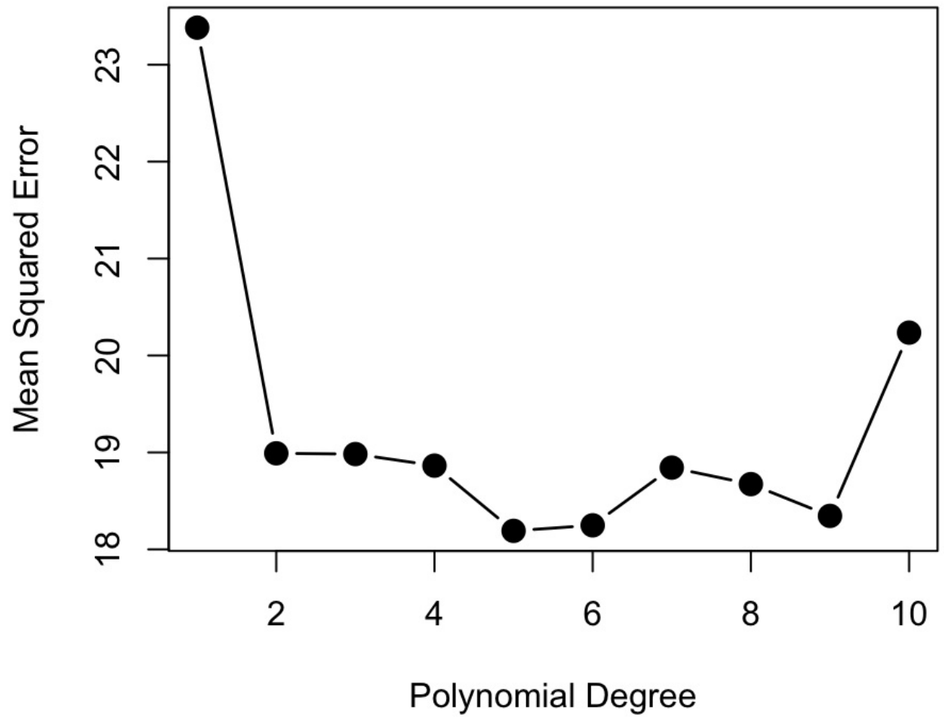
For model selection, we want to decide what degree d we should use to model this data.

Validation Set

Using the validation set method, split our data into a training set,

fit the regression model with different polynomial degrees d on the training set,

measure test error on the validation set.



Resampled validation set

The validation set approach can be prone to sampling issues.

It can be highly variable as error rate is a random quantity and depends on observations in training and validation sets.

Resampled validation set

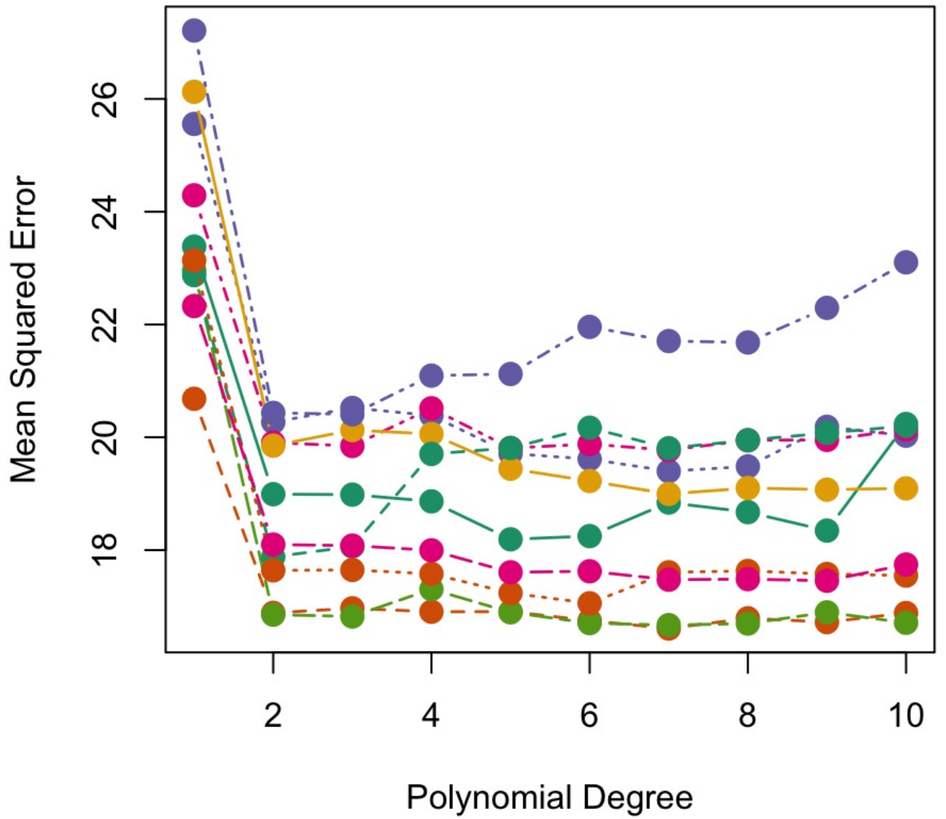
The validation set approach can be prone to sampling issues.

It can be highly variable as error rate is a random quantity and depends on observations in training and validation sets.

We can improve our estimate of test error by averaging multiple measurements of it (remember the law of large numbers).

Resampled validation set

Resample validation set 10 times (yielding different validation and training sets) and averaging the resulting test errors.



Leave-one-out Cross-Validation

This approach still has some issues.

Each of the training sets in our validation approach only uses 50% of data to train, which leads to models that may not perform as well as models trained with the full dataset and thus we can overestimate error.

Leave-one-out Cross-Validation

This approach still has some issues.

Each of the training sets in our validation approach only uses 50% of data to train, which leads to models that may not perform as well as models trained with the full dataset and thus we can overestimate error.

To alleviate this situation, we can extend our approach to the extreme: Make each single training point it's own validation set.

Leave-one-out Cross-Validation

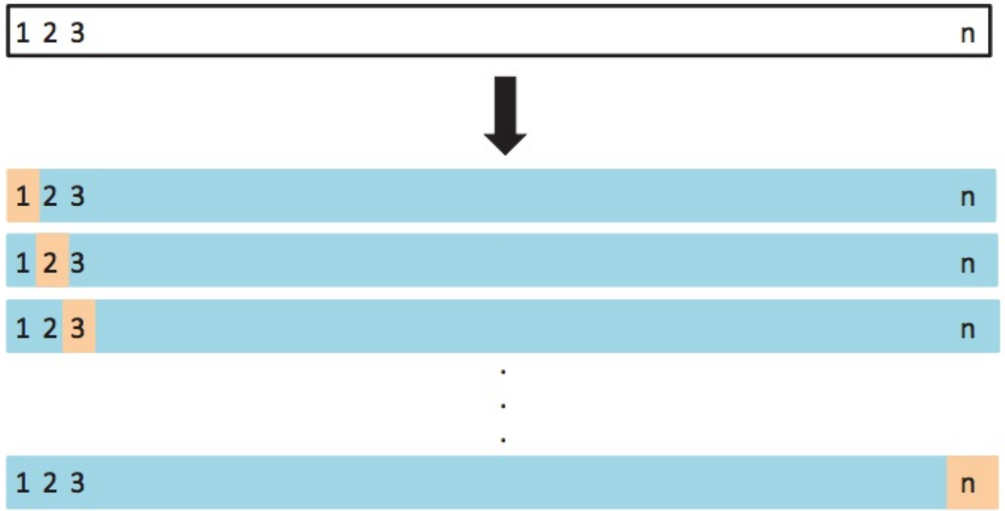
Procedure:

For each observation i in data set:

a. Train model on all but i-th observation

b. Predict response for i-th observation

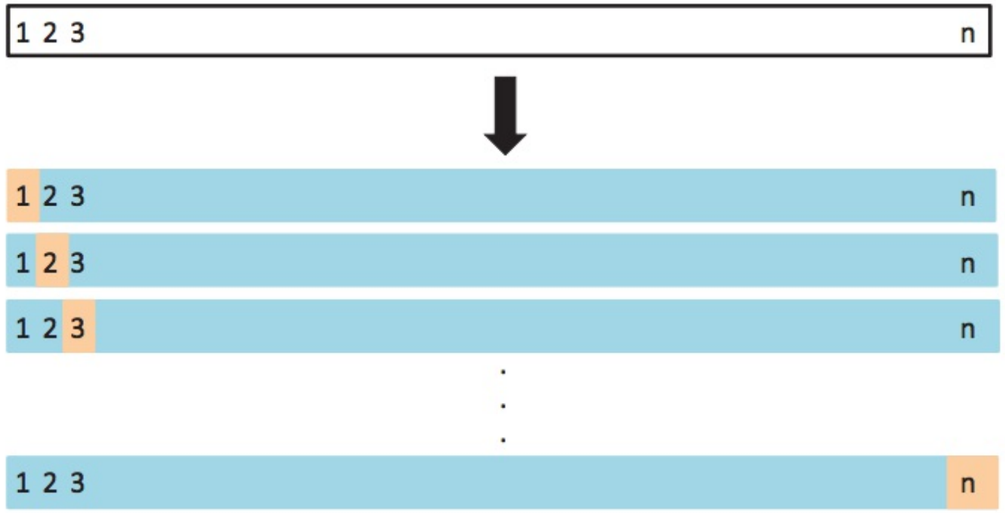
c. Calculate prediction error



Leave-one-out Cross-Validation

This gives us the following cross-validation estimate of error.

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



Leave-one-out Cross-Validation

Advantages:

- use $n-1$ observations to train each model
- no sampling effects introduced since error is estimated on each sample

Leave-one-out Cross-Validation

Advantages:

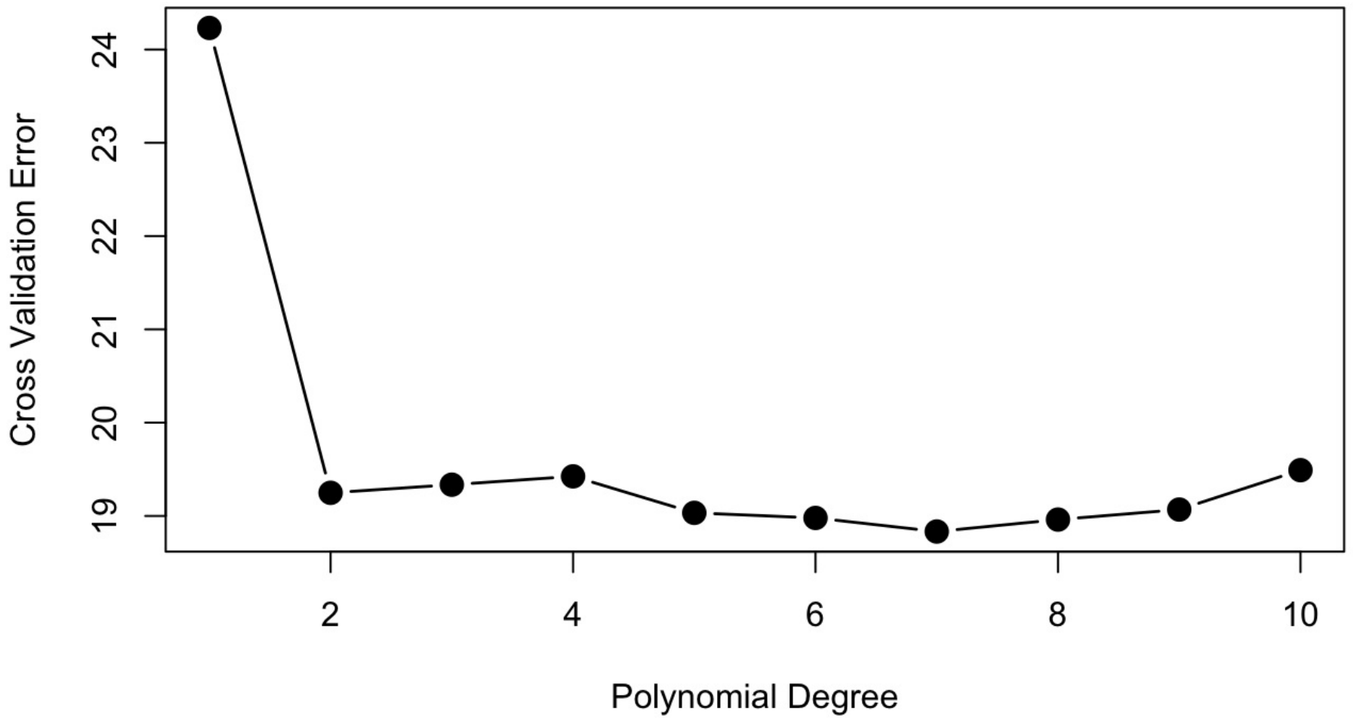
- use $n-1$ observations to train each model
- no sampling effects introduced since error is estimated on each sample

Disadvantages:

- Depending on the models we are trying to fit, it can be very costly to train $n-1$ models.
- Error estimate for each model is highly variable (since it comes from a single datapoint).

Leave-one-out Cross-Validation

On our running example



Leave-one-out Cross-validation

For linear models (and some non-linear models) there is a nice trick that allows one to compute (exactly or approximately) LOOCV from the full data model fit which we will not get into here.

See notes and slides for chapter 2

k-fold Cross-Validation

This discussion leads us to the most commonly used cross-validation approach k-fold Cross-Validation.

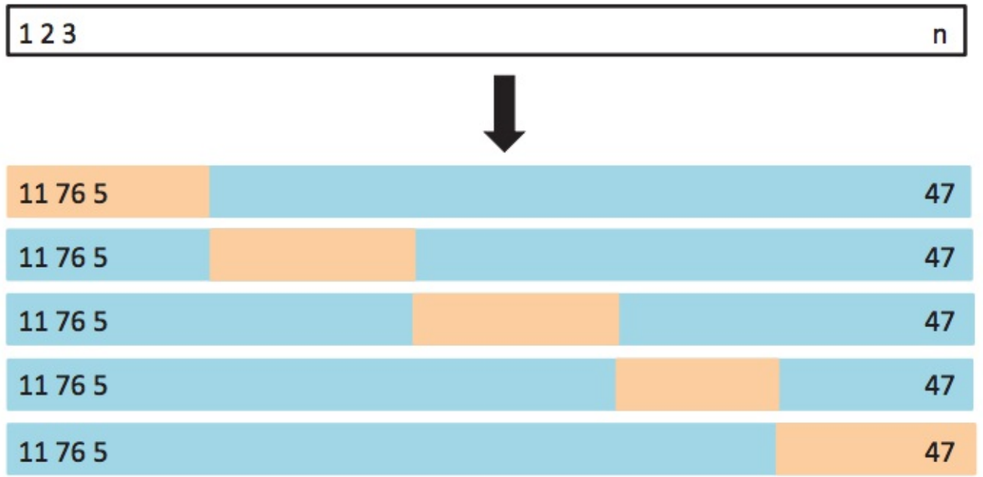
k-fold Cross-Validation

Procedure:

Partition observations randomly into k groups (folds).

For each of the k groups of observations:

- Train model on observations in the other $k-1$ folds
- Estimate test-set error (e.g., Mean Squared Error) on this fold



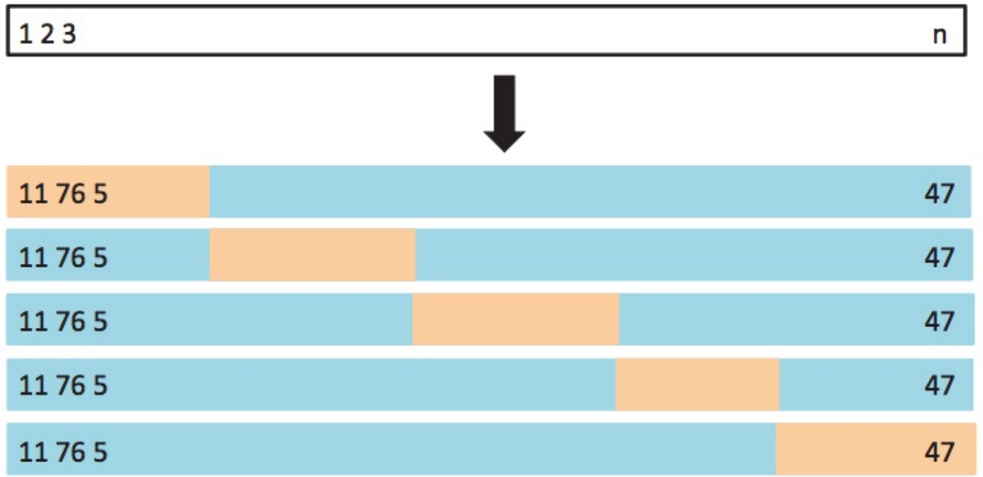
k-fold Cross-Validation

Procedure:

Compute average error across k folds

$$CV_{(k)} = \frac{1}{k} \sum_i MSE_i$$

where MSE_i is mean squared error estimated on the i-th fold



k-fold Cross-Validation

- Fewer models to fit (only k of them)
- Less variance in each of the computed test error estimates in each fold.

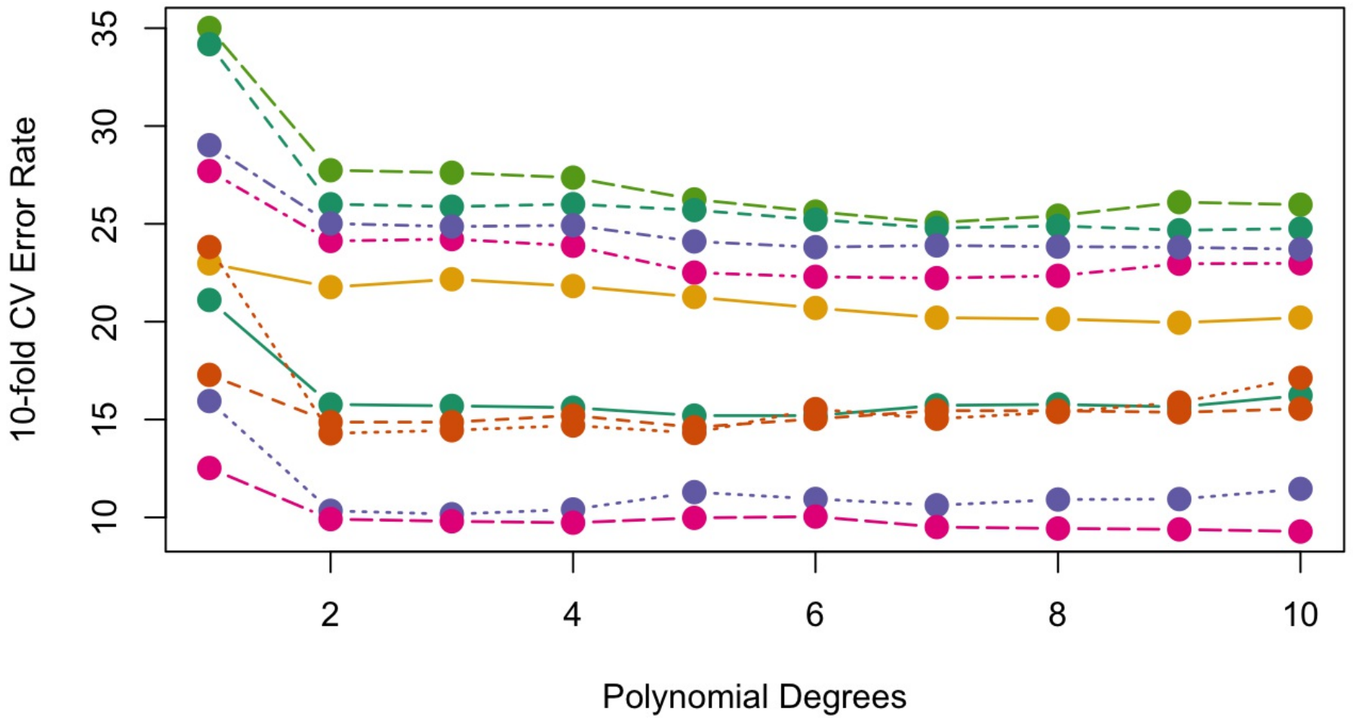
k-fold Cross-Validation

- Fewer models to fit (only k of them)
- Less variance in each of the computed test error estimates in each fold.

It can be shown that there is a slight bias (over estimating usually) in error estimate obtained from this procedure.

k-fold Cross-Validation

Running Example



Cross-Validation in Classification

Each of these procedures can be used for classification as well.

In this case we would substitute MSE with performance metric of choice.
E.g., error rate, accuracy, TPR, FPR, AUROC.

Cross-Validation in Classification

Each of these procedures can be used for classification as well.

In this case we would substitute MSE with performance metric of choice. E.g., error rate, accuracy, TPR, FPR, AUROC.

Note however that not all of these work with LOOCV (e.g. AUROC since it can't be defined over single data points).

Evaluating Classification

The AUROC statistic

The AUROC statistic is related to the Mann-Whitney non-parametric statistical test for distributional differences.

Null hypothesis: for randomly drawn pair of samples from two populations, it is equally likely that sample from first population is greater than sample from second population.

Evaluating Classification

The AUROC statistic

The AUROC statistic is related to the Mann-Whitney non-parametric statistical test for distributional differences.

Null hypothesis: for randomly drawn pair of samples from two populations, it is equally likely that sample from first population is greater than sample from second population.

Specifically, if x_A and x_B are drawn randomly from populations A and B respectively, $P(x_A < x_B) = P(x_A > x_B)$.

Evaluating Classification

The AUROC statistic

Consider a classifier C trained to distinguish between two classes, using a training set containing n_A and n_B instances for each of the two classes respectively.

Evaluating Classification

The AUROC statistic

Consider a classifier C trained to distinguish between two classes, using a training set containing n_A and n_B instances for each of the two classes respectively.

Denote as C_i the score given by classifier i with higher C_i indicating predictions for class A.

Evaluating Classification

The AUROC statistic

Use the Mann-Whitney test to verify that scores for class A are greater than scores for class B

Evaluating Classification

The AUROC statistic

Use the Mann-Whitney test to verify that scores for class A are greater than scores for class B

Null hypothesis: $P(C_i < C_j) = P(C_j < C_i)$ for randomly drawn pairs C_i from class A and C_j from class B.

Evaluating Classification

The AUROC statistic

The Mann-Whitney test uses the U statistic to perform this test:

$$U = \sum_{i=1}^{n_A} \sum_{j=1}^{n_B} \frac{I\{C_i > C_j\}}{n_A n_B}$$

This is an empirical estimate of $P(C_i > C_j)$, which under the null hypothesis of the Mann-Whitney test is 0.5.

Evaluating Classification

The AUROC statistic

The Mann-Whitney test uses the U statistic to perform this test:

$$U = \sum_{i=1}^{n_A} \sum_{j=1}^{n_B} \frac{I\{C_i > C_j\}}{n_A n_B}$$

This is an empirical estimate of $P(C_i > C_j)$, which under the null hypothesis of the Mann-Whitney test is 0.5.

It can be shown that U is exactly the AUCROC.

Evaluating Classification

The AUROC statistic

Note that the U statistic, and thus AUROC, is only dependent on the rank of scores c_i not on their magnitude.

Evaluating Classification

The AUROC statistic

Note that the U statistic, and thus AUROC, is only dependent on the rank of scores c_i not on their magnitude.

This implies that we can compare AUCROC for classifiers that produce scores in different scales, e.g., probabilities or not.

Evaluating Classification

The AUROC statistic

The relationship to the Mann-Whitney test also permits to use its inferential tools on AUCROC statistics.

See http://papers.nips.cc/paper/2645-confidence-intervals-for-the-area-under-the-roc-curve.pdf

Evaluating Classification

The AUROC statistic

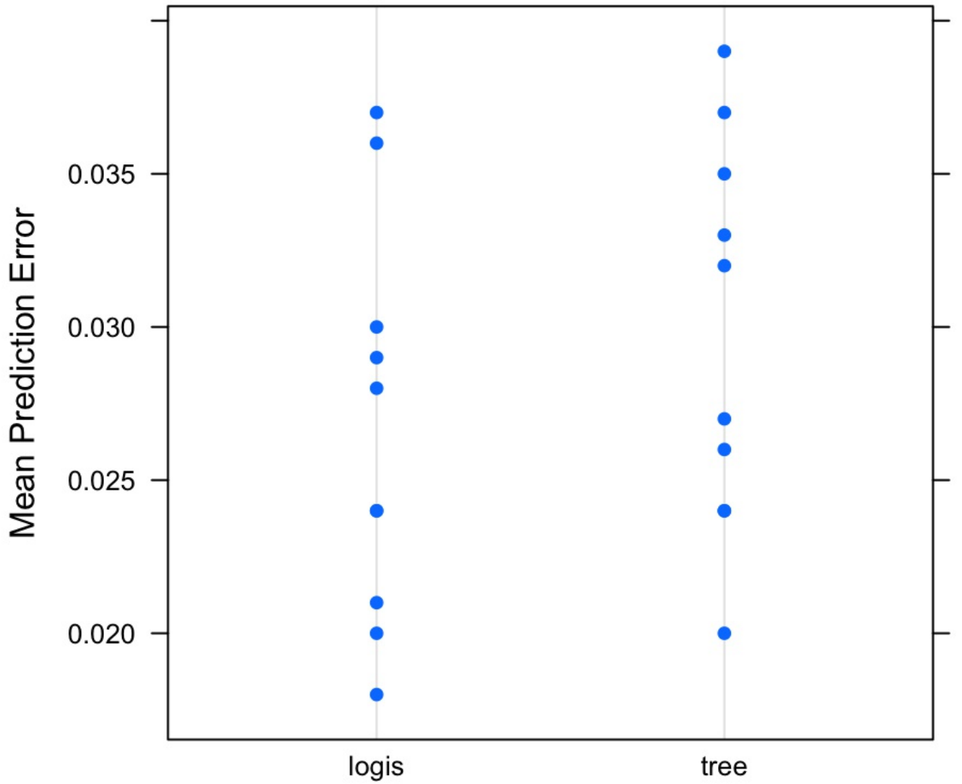
The relationship to the Mann-Whitney test also permits to use its inferential tools on AUCROC statistics.

See http://papers.nips.cc/paper/2645-confidence-intervals-for-the-area-under-the-roc-curve.pdf

There are methods to compare AUCROC statistics from multiple classifiers. See http://ieeexplore.ieee.org/document/6851192/ for the most practical.

Comparing models using cross-validation

Suppose you want to compare two classification models (logistic regression vs. a decision tree) on the `Default` dataset. We can use Cross-Validation to determine if one model is better than the other, using a t-test for example.



Comparing models using cross-validation

Using hypothesis testing:

term	estimate	std.error	statistic	p.value
(Intercept)	0.0267	0.0020306	13.148828	0.0000000
methodtree	0.0030	0.0028717	1.044677	0.3099998

In this case, we do not observe any significant difference between these two classification methods.

Summary

Model selection and assessment are critical steps of data analysis.

Resampling methods are general tools used for this purpose.

k-fold cross-validation can be used to provide larger training sets to algorithms while stabilizing empirical estimates of expected prediction error