Model Selection

Dan Lizotte

2017-09-25

Performance Evaluation vs. Model Selection

1. ***Performance:*** We would like to estimate the **generalization error** of our resulting predictor.
2. ***Model selection:*** We would like to choose the best model space (e.g. linear, quadratic, …) **for the data we have**

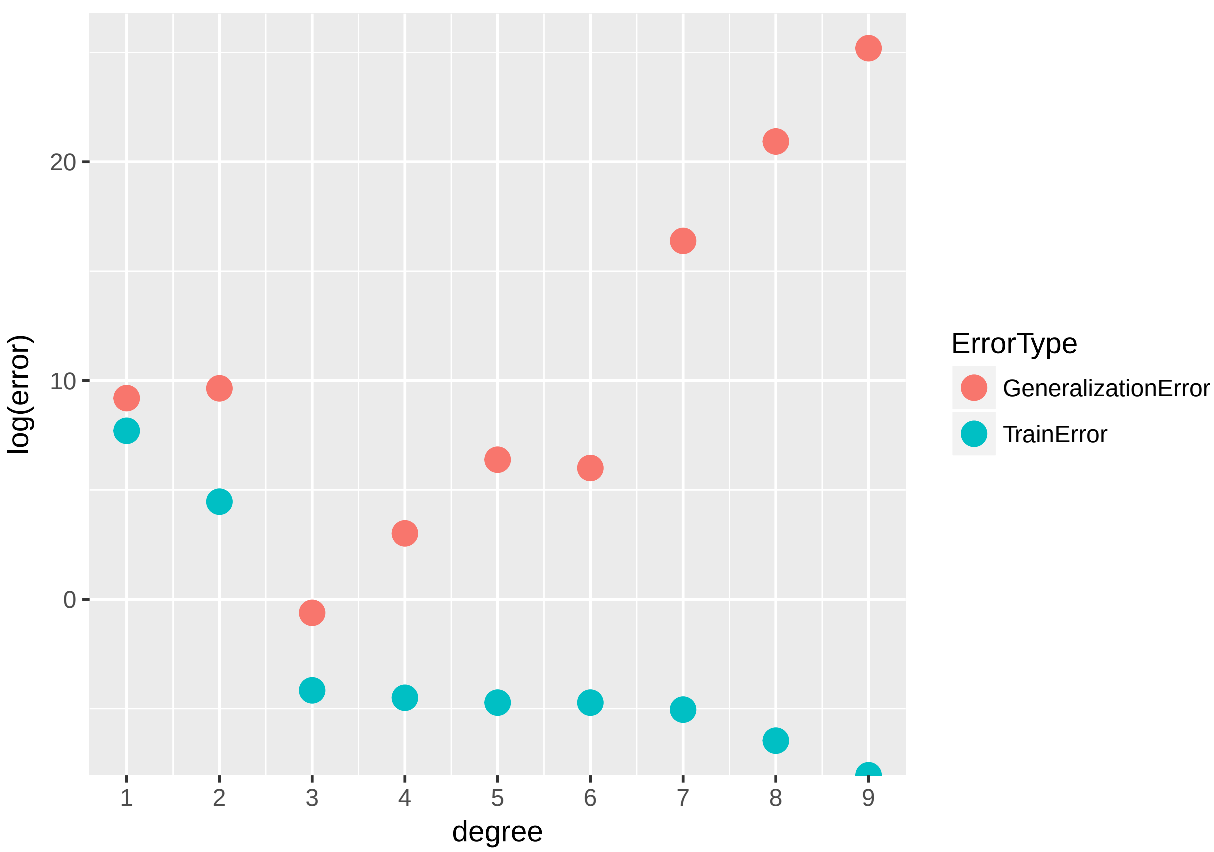
Supervised Learning Redux

1. Choose model class
2. Find the model in the class that gives the minimum training error.

But we saw previously that ***generalization error*** is what we really want to minimize.

And picking the wrong model class can be catastrophic.

Training Error, Generalization Error, Model Space



The best model space is not the simplest nor the most complex.

Overfitting

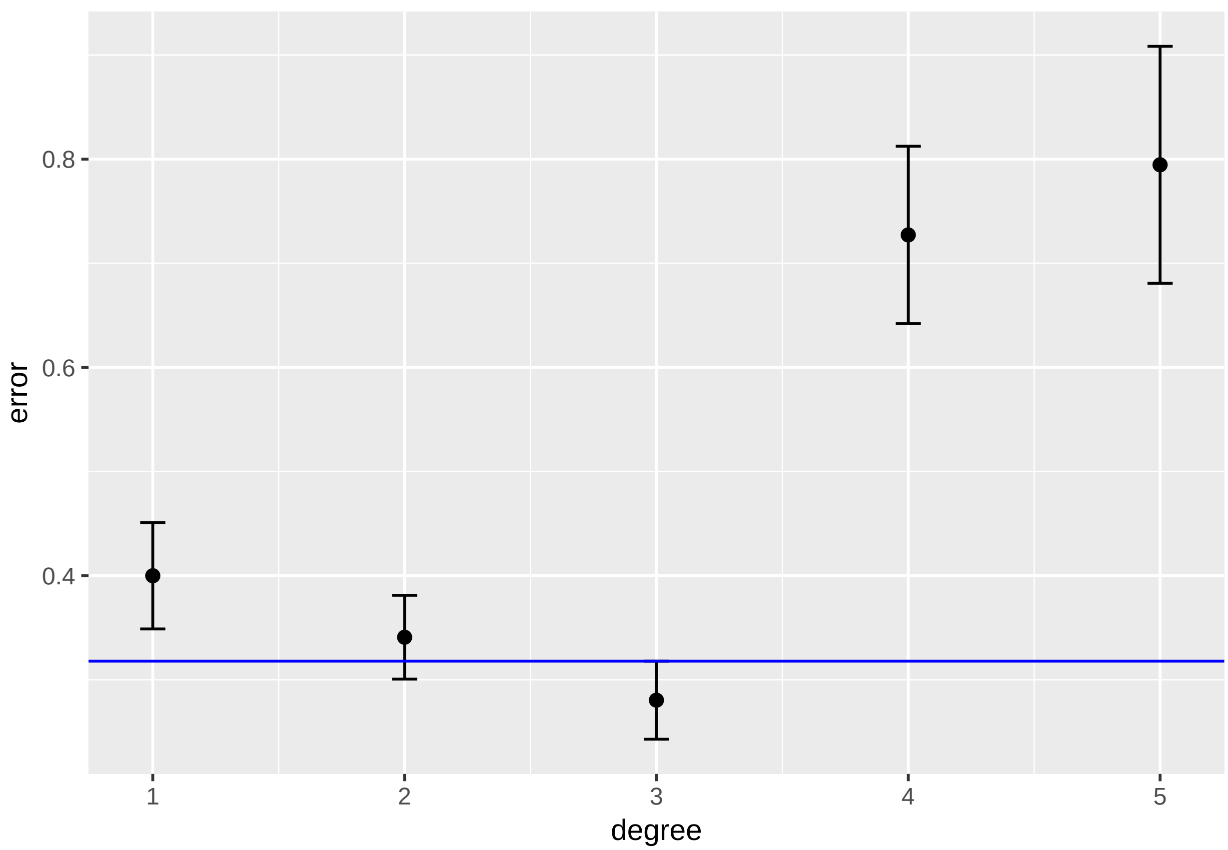
Larger model spaces *always* lead to lower training error.

- Suppose $\mathcal{H}_1$ is the space of all linear functions, $\mathcal{H}_2$ is the space of all quadratic functions. Note $\mathcal{H}_1 \subset \mathcal{H}_2$.
- Fix a data set.
- Let $h_1^* = \arg\min_{h' \in \mathcal{H}_1} \frac{1}{n} \sum_{i=1}^{n} L(h'(\mathbf{x}_i), y_i)$ and $h_2^* = \arg\min_{h' \in \mathcal{H}_2} \frac{1}{n} \sum_{i=1}^{n} L(h'(\mathbf{x}_i), y_i)$, both computed using the same dataset.
- It **must** be the case that $\min_{h' \in \mathcal{H}_2} \frac{1}{n} \sum_{i=1}^{n} L(h'(\mathbf{x}_i), y_i) \leq \min_{h' \in \mathcal{H}_1} \frac{1}{n} \sum_{i=1}^{n} L(h'(\mathbf{x}_i), y_i)$,
- Small training error, large generalization error is known as **overfitting**

Model Selection Strategy 1: A Validation Set

- A separate **validation set** can be used for model selection.
  - Train on the training set using each proposed model space
  - Evaluate each on the validation set, identify the one with lowest *validation* error
  - Choose the simplest model with performance $<$ 1 std. error worse than the best.

Validation Set Method



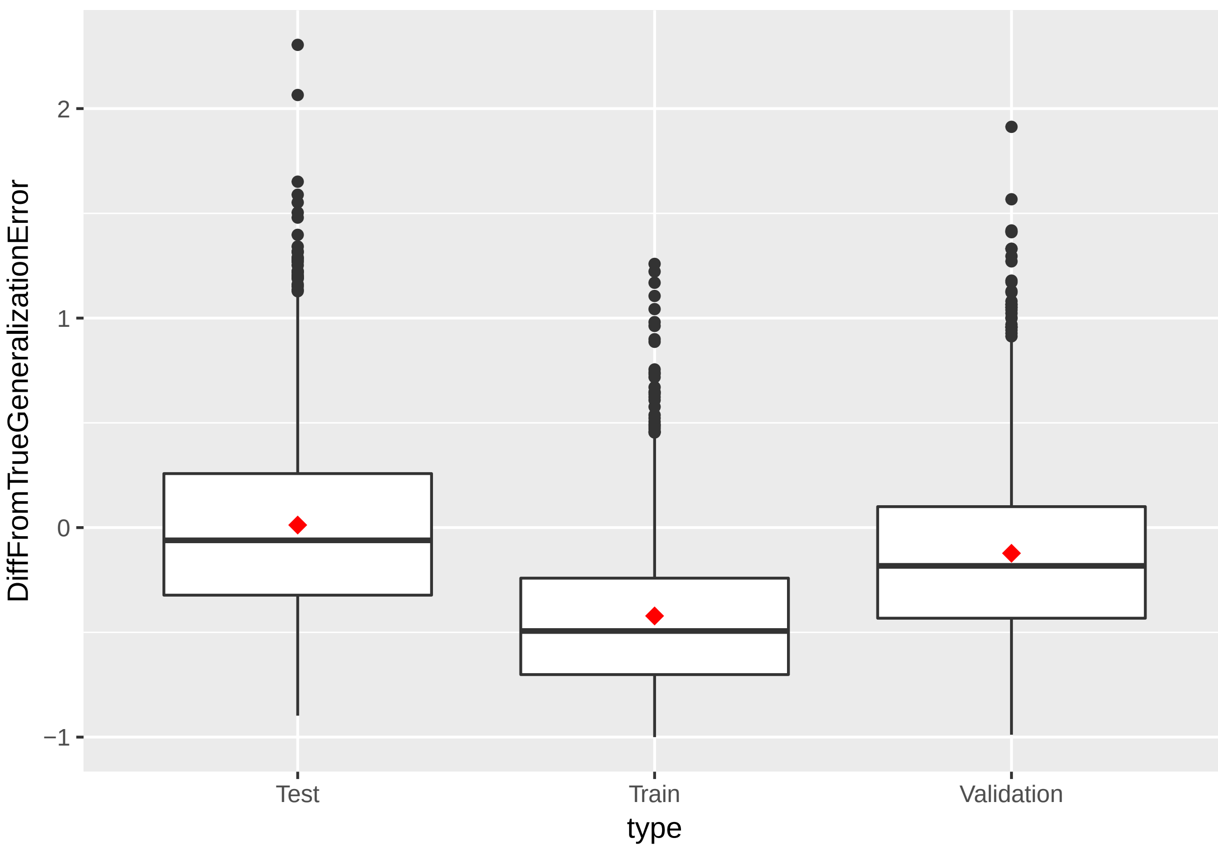
Validation sets for generalization error?

Experimental Scenario

- Generate training data, validation data
- Choose best model using validation data as per above
- Estimate performance of best model using validation data

Will this produce an unbiased estimate of generalization error?

Scenario



Training, Model Selection, and Performance Evaluation

- A general procedure for doing model selection and performance evaluation
- The data is randomly partitioned into three disjoint subsets:
  - A *training set* used only to find the parameters **w**
  - A *validation set* used to find the right model space (e.g., the degree of the polynomial)
  - A *test set* used to estimate the generalization error of the resulting model
- Can generate standard confidence intervals for the generalization error of the learned model

Problems with the Single-Partition Approach

- Pros:
  - Measures what we want: Performance of the actual learned model.
  - Simple
- Cons:
  - Smaller effective training sets make performance and performance estimates more variable.
  - Small validation sets can give poor model selection
  - Small test sets can give poor estimates of performance
  - For a test set of size 100, with 60 correct classifications, 95% C.I. for actual accuracy is $(0.497, 0.698)$.

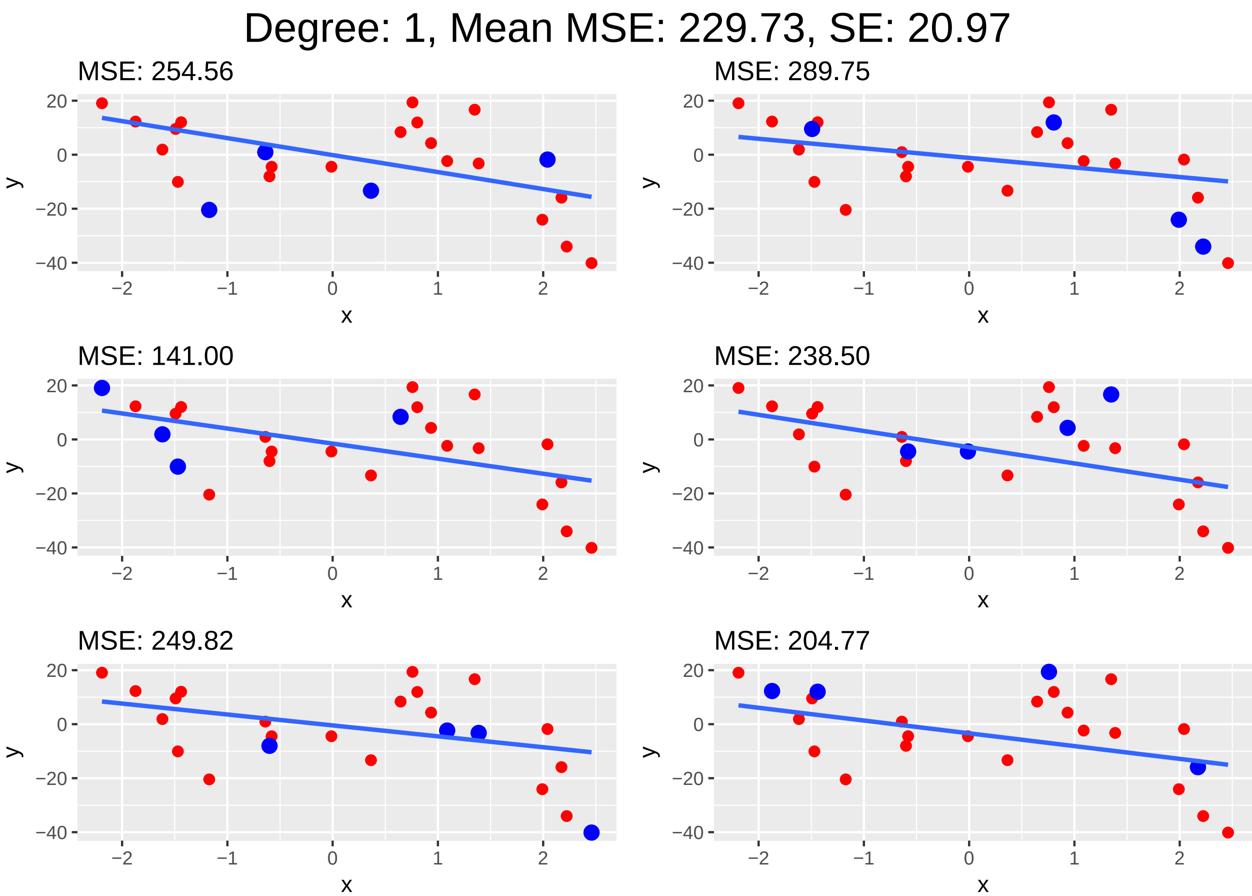
k-fold cross-validation (HTF 7.10, JWHT 5.1)

- Divide the instances into k disjoint partitions or folds of size n/k
- Loop through the partitions $i = 1...k$:
  - Partition i is for evaluation (i.e., estimating the performance of the algorithm after learning is done)
  - The rest are used for training (i.e., choosing the specific model within the space)
- *"Cross-Validation Error" is the average error on the evaluation partitions.* Has lower variance than error on one partition.
- This is the main CV **idea**; CV is used for different purposes though.

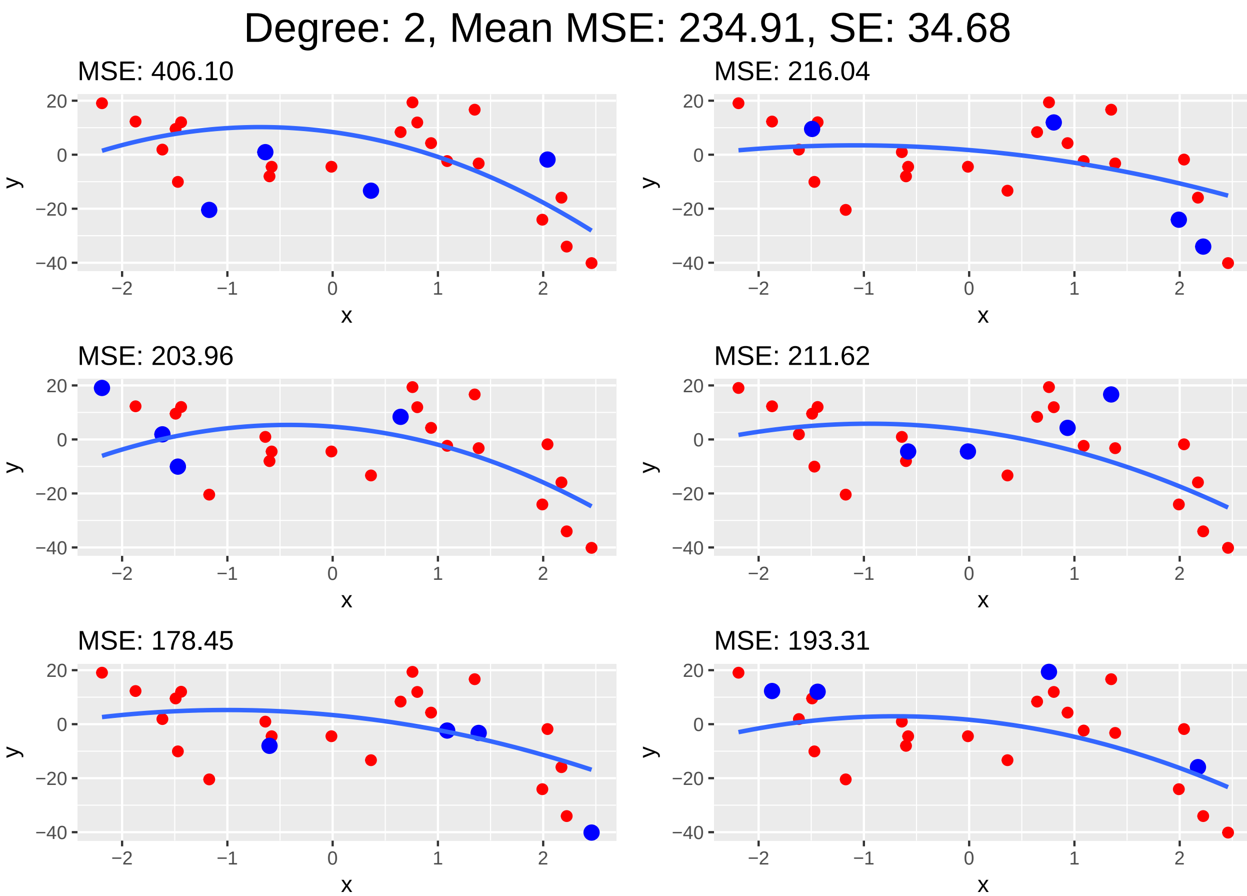
k-fold cross-validation model selection (HTF 7.10, JWHT 5.1)

- Divide the instances into k folds of size n/k.
- Loop over m model spaces $1...m$
  - Loop over the k folds $i = 1...k$:
    * Fold i is for validation (i.e., estimating the performance of the algorithm after learning is done)
    * The rest are used for training (i.e., choosing the specific model within the space)
- For each model space, report average error over folds, and standard error.

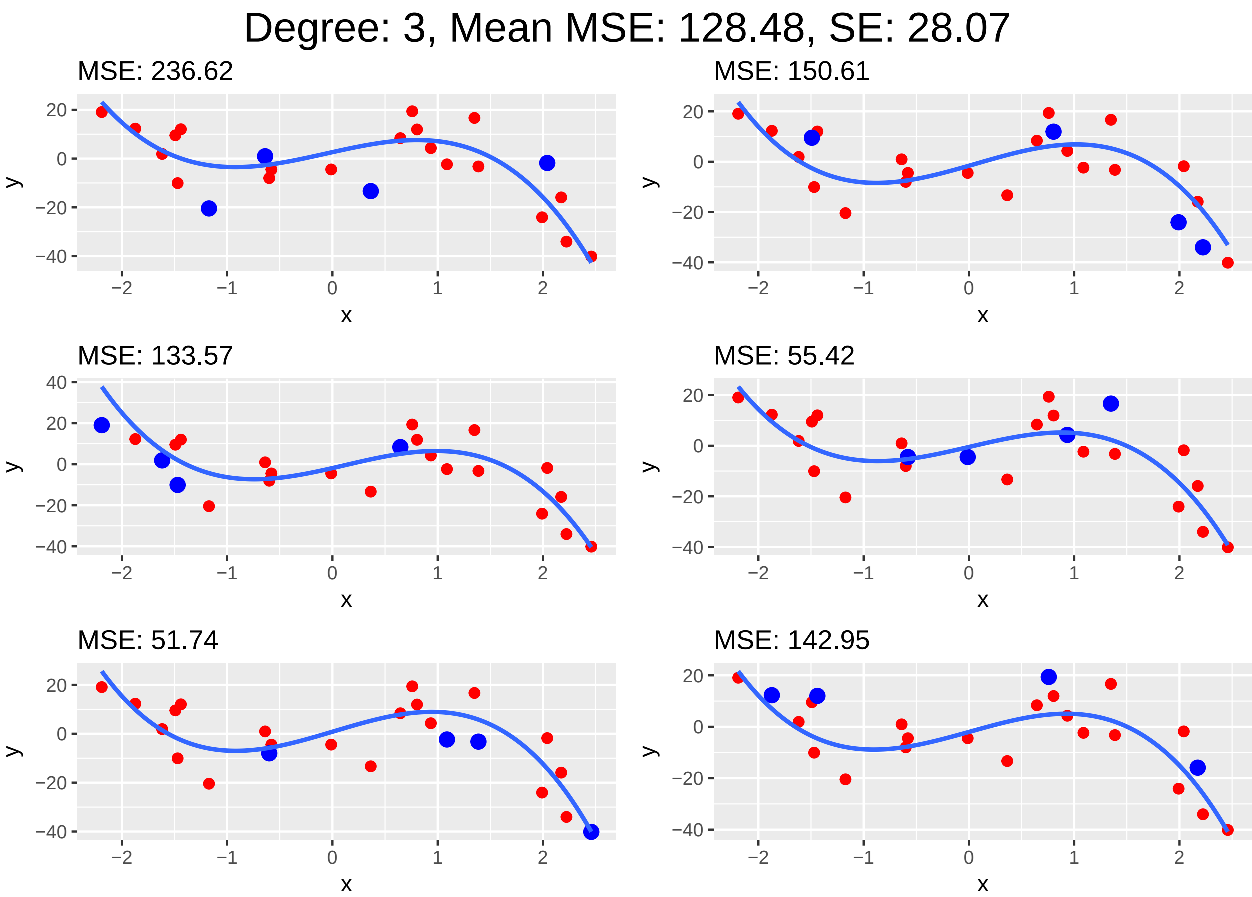
CV for Model Selection



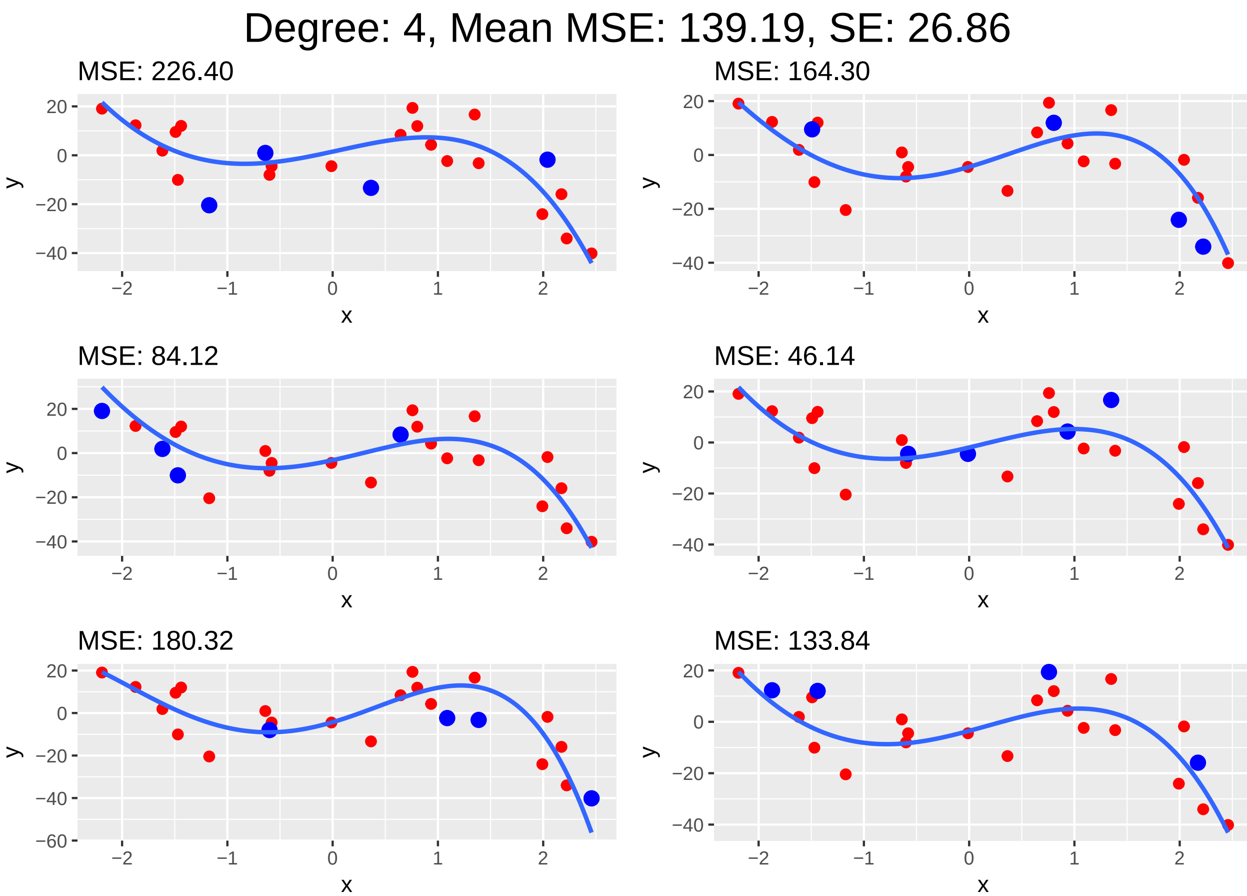
CV for Model Selection



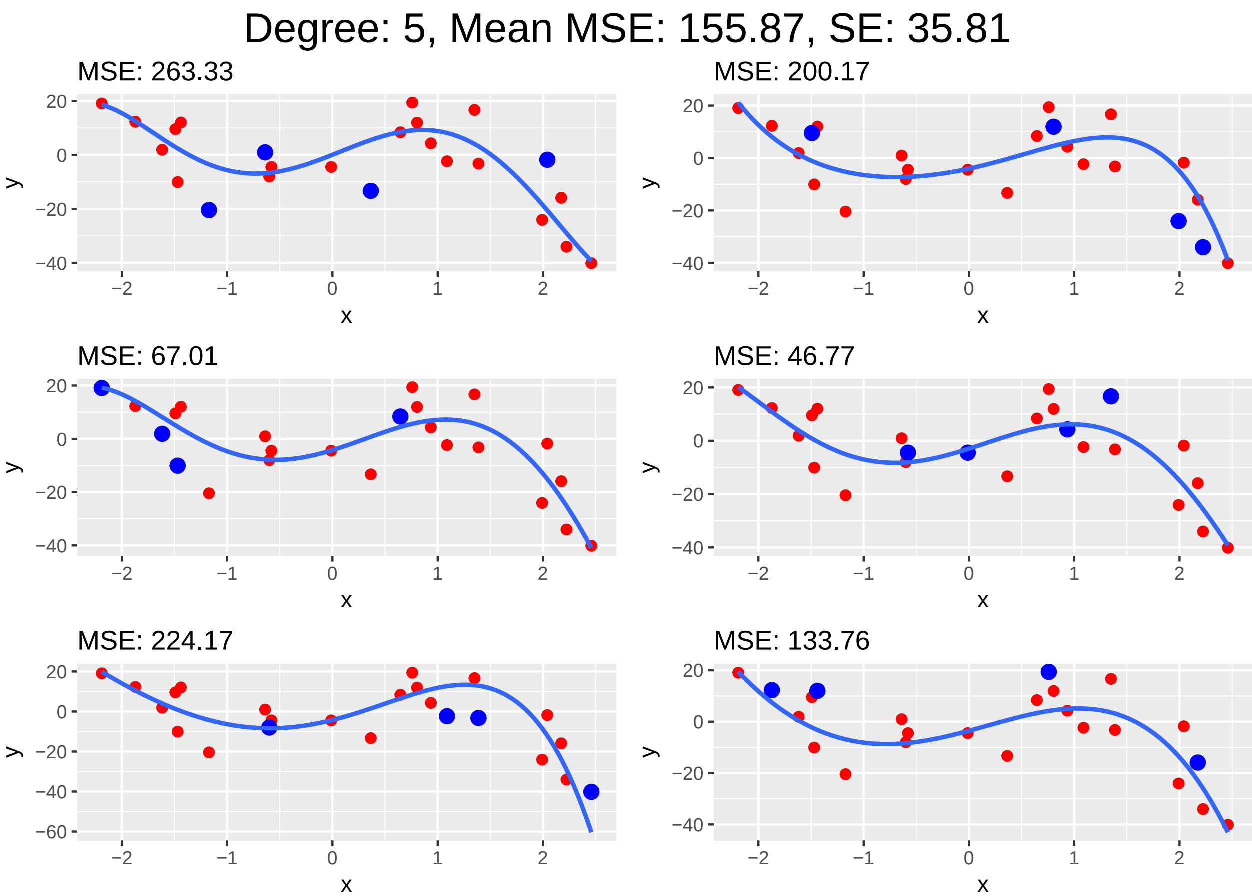
CV for Model Selection



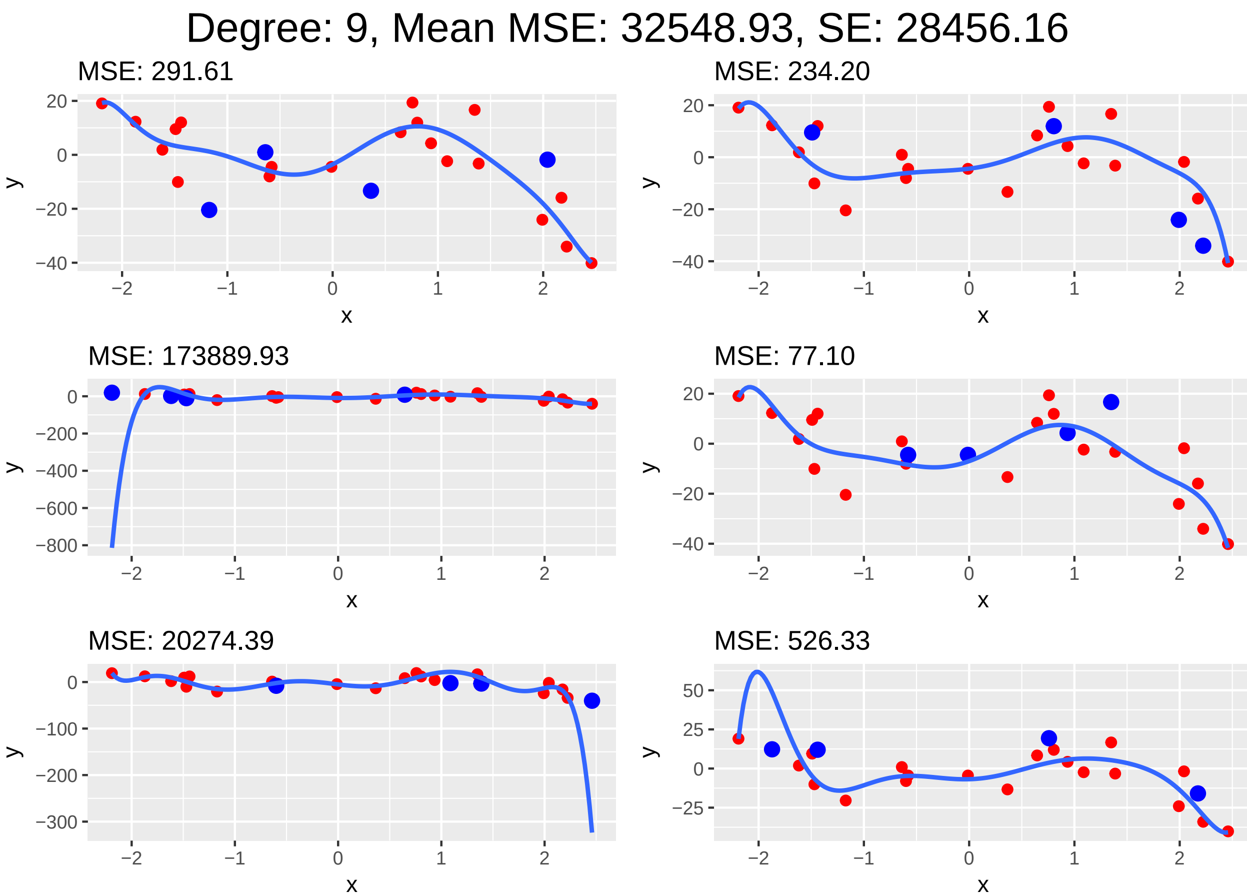
CV for Model Selection



CV for Model Selection



CV for Model Selection



CV for Model Selection

- As with a single validation set, select "most parsimonious model whose error is no more than one standard error above the error of the best model." (HTF, p.244)

Estimating "which is best" vs. "performance of best"

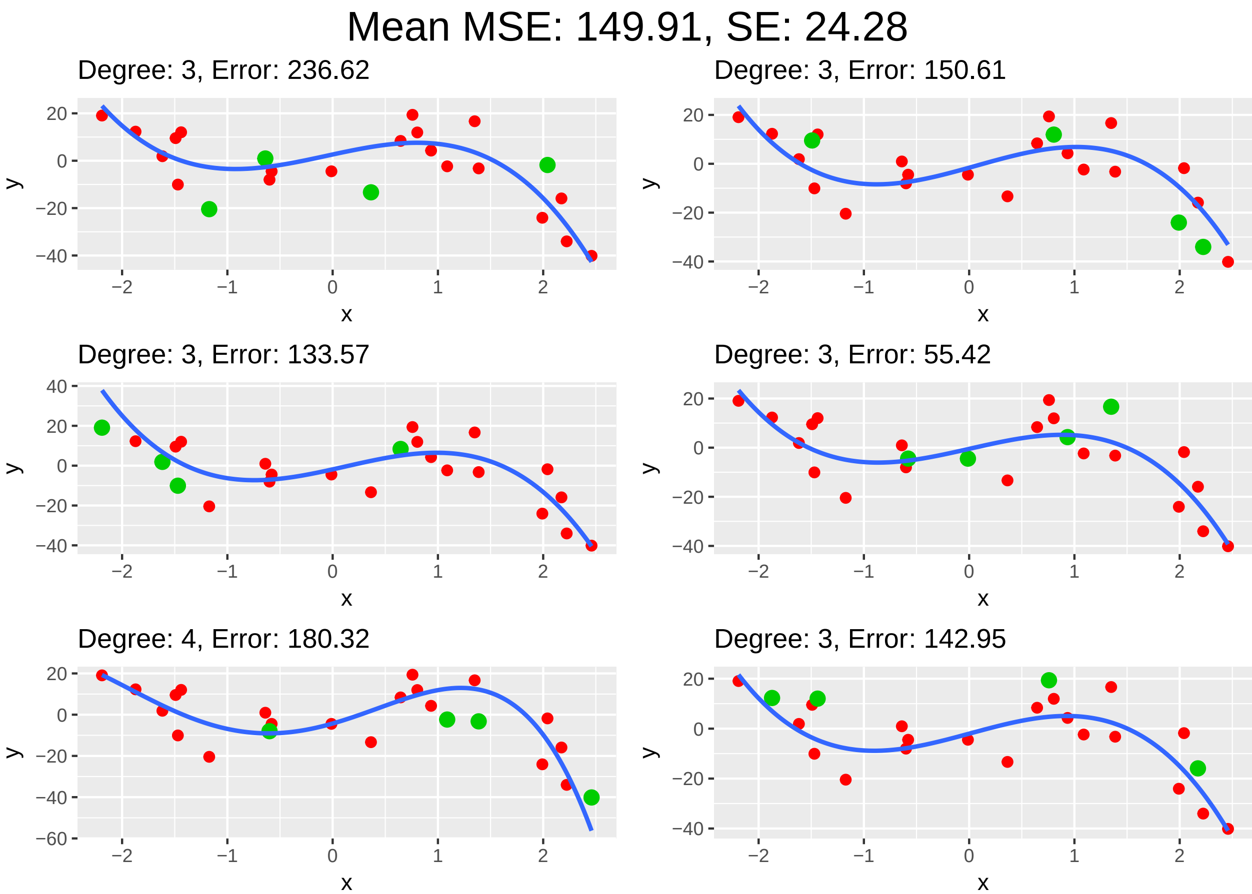
Estimated errors using 290 model spaces.

Nested CV for Model Evaluation

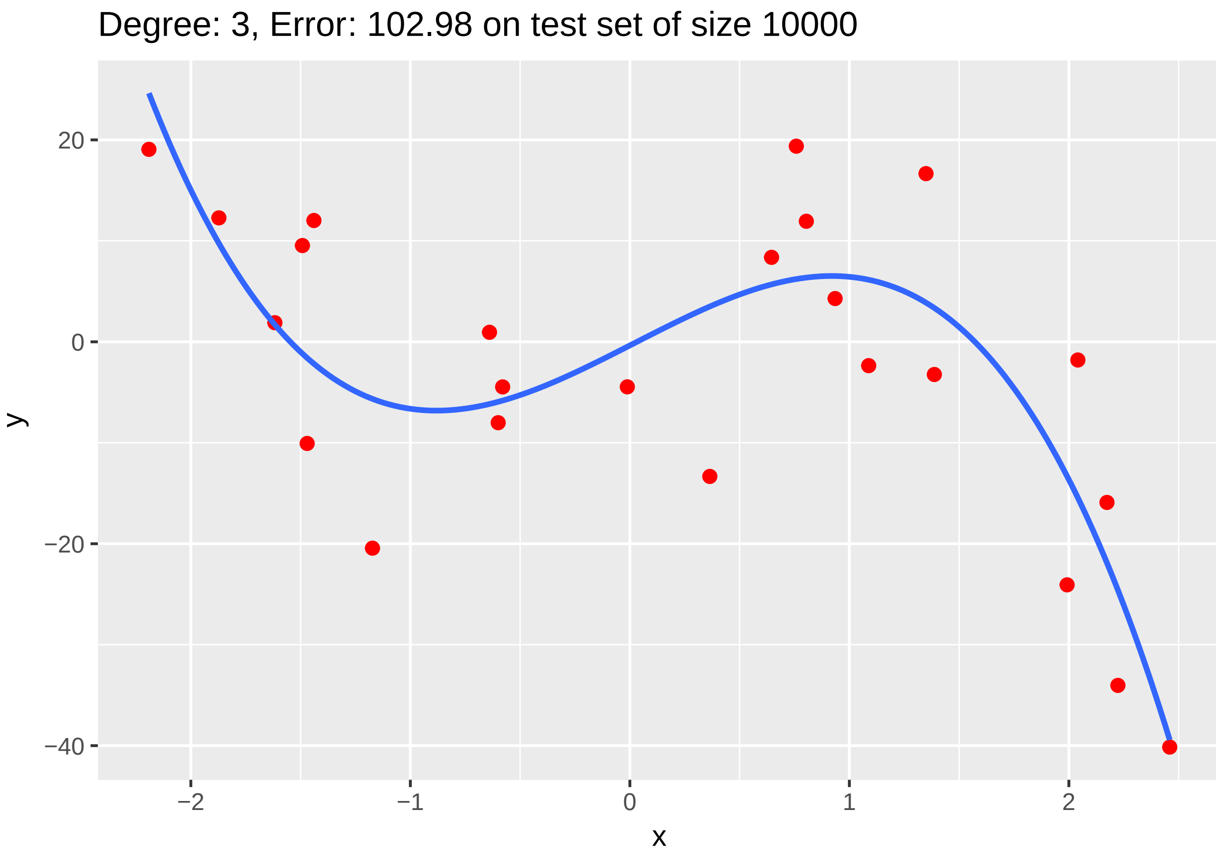
- Divide the instances into k "outer" folds of size n/k.
- Loop over the k outer folds $i = 1...k$:
  - Fold i is for testing; all others for training.
  - Divide the training instances into k' "inner" folds of size $(n - n/k)/k'$.
  - Loop over m model spaces $1...m$
    - Loop over the k' inner folds $j = 1...k'$:
      - Fold j is for validation

    - The rest are used for training
  * Use average error over folds and SE to choose model space.
  * Train on all inner folds.
- Test the model on outer test fold

Nested CV for Model Evaluation



Generalization Error for degree 3 model



Minimum-CV Estimate: 128.48, Nested CV Estimate: 149.91

Bias-correction for the CV Procedure

Cawley, Talbot. On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. JMLR v.11, 2010.

Tibshirani, Tibshirani. A bias correction for the minimum error rate in cross-validation. arXiv. 2009.

Ding et al. Bias correction for selecting the minimal-error classifier from many machine learning models. Bioinformatics 30 (22). 2014.

Summary

- The training error decreases with *the complexity (size) of the model space*
- Generalization error decreases at first, then starts increasing
- Set aside a *validation set* helps us find a good model space
- We then can report unbiased error estimate, using a *test set*, **untouched during both parameter training and validation**
- Cross-validation is a lower-variance but possibly biased version of this approach. It is standard.
- ***If you have lots of data, just use held-out validation and test sets.***