Kris Sankaran

Published Sep 23, 2019

Cross Validation and Model Selection

IFT6758, Fall 2019

Reading: ISLR section 5.1 and PDS pg. 359 - 375

Daily Choices for Data Scientists

Knowing how to fit models is not enough, if you want to solve a real-world problem.

- How should you select between model families?
- Which parameters are best within a model family?

- Should you be trying to improve the data?
  - More samples? Richer features?
  - Less missingness, fewer outliers, ...

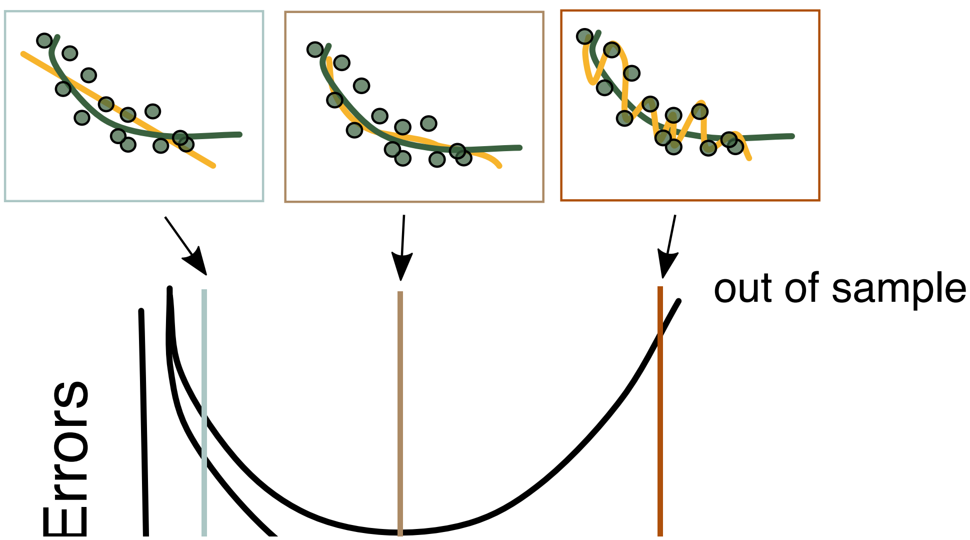
?

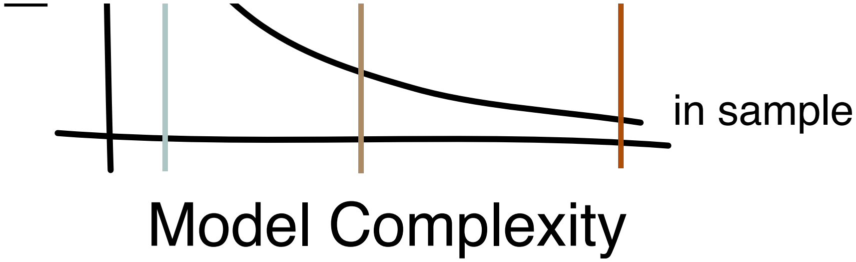
Transitioning to Inference

- We'll be more introspective, trying to understand properties of our algorithms
- The heart of inference: Being critical of the processes people use to learn from data

Reminder: Bias-Variance Tradeoff

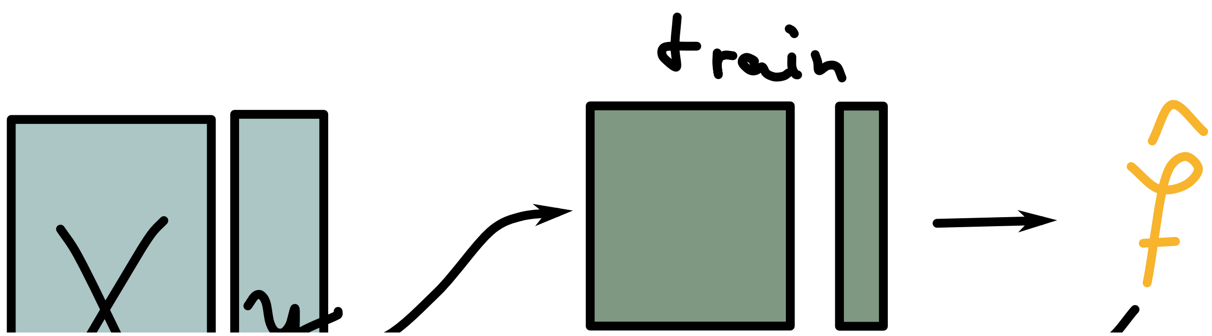
- Ultimately, you want your model to perform well on out-of-sample data
- If you only evaluate on in-sample data, you will underestimate the out-of-sample error

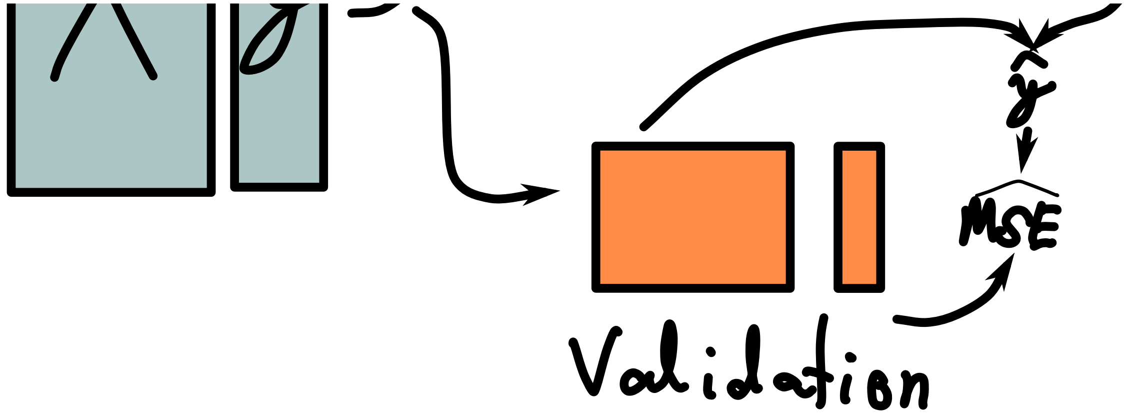




Validation Sets

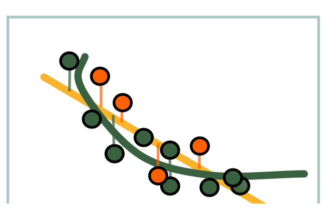
- To approximate the out-of-sample error, we can use a validation set.
- Randomly divide your sample into two pieces, one to train and another to validate



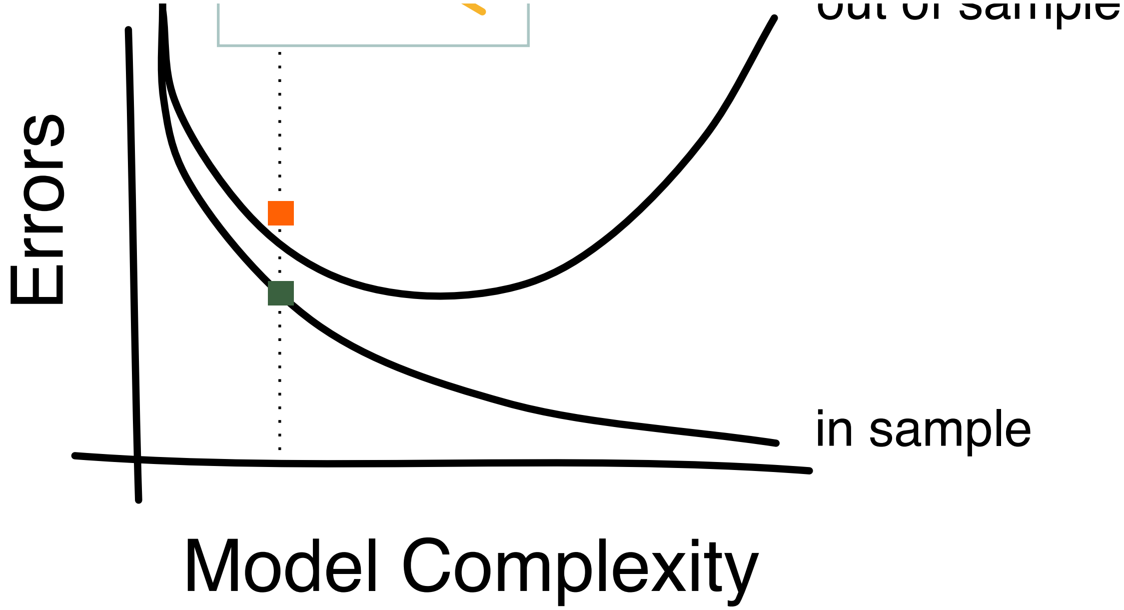


Validation Sets

If you run this over models with different degrees of complexity, you can see the bias-variance tradeoff.

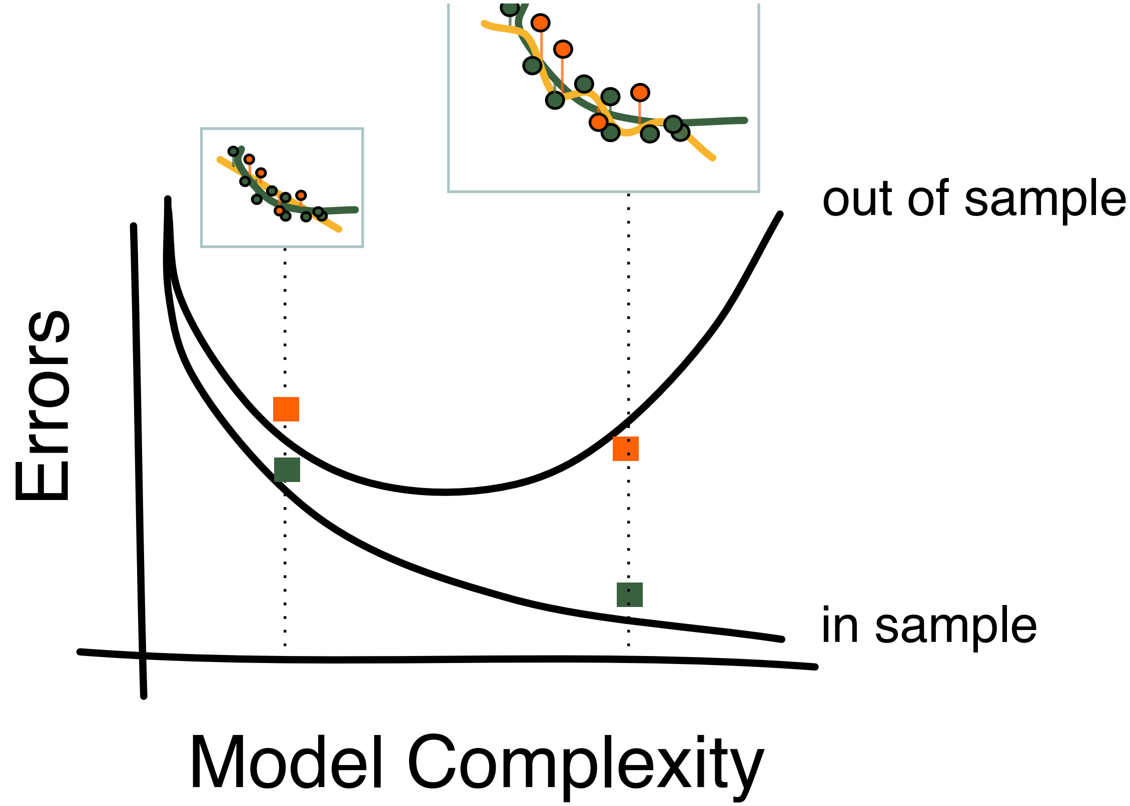


out of sample



Validation Sets

If you run this over models with different degrees of complexity, you can see the bias-variance tradeoff.

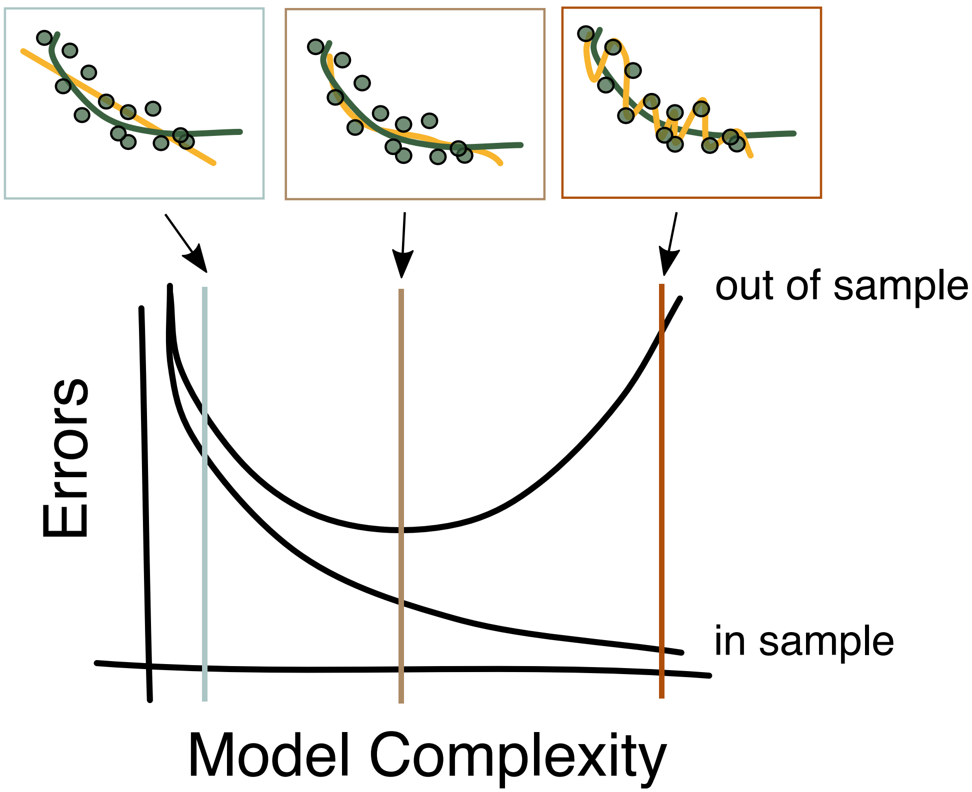


Complexity Regimes

Even if you only evaluate the train / validation error for a model of a given complexity, you get useful information.

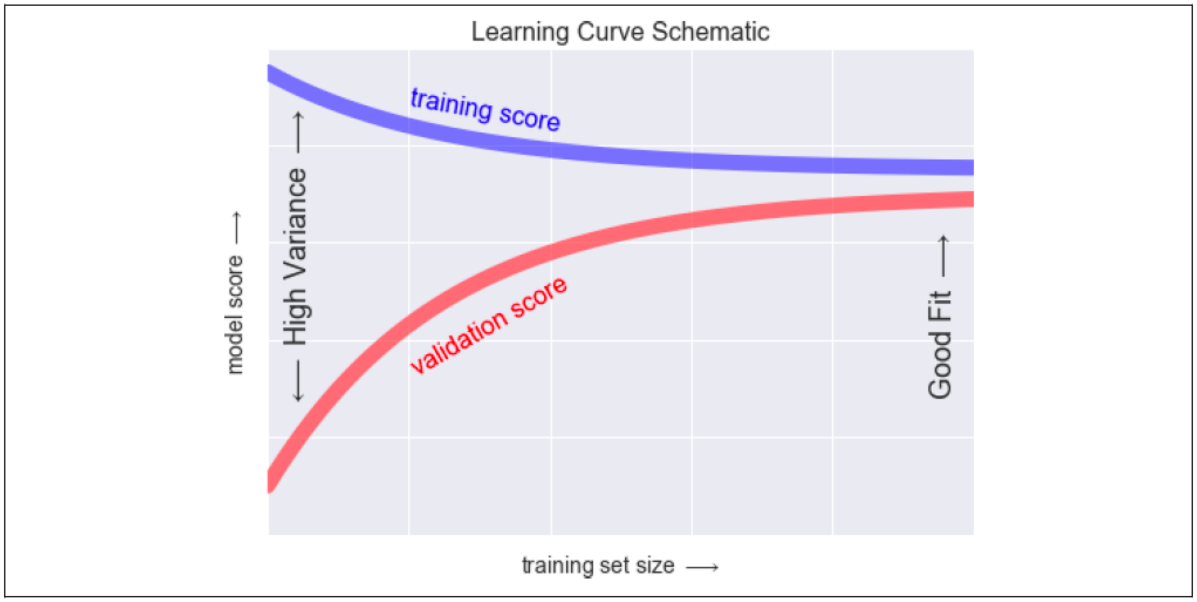
- Training $\ll$ validation error $\rightarrow$ Model is overfit
- Training $\approx$ validation error $\rightarrow$ Model is underfit (or OK)

Common heuristic: Overfit the data first, then regularize.



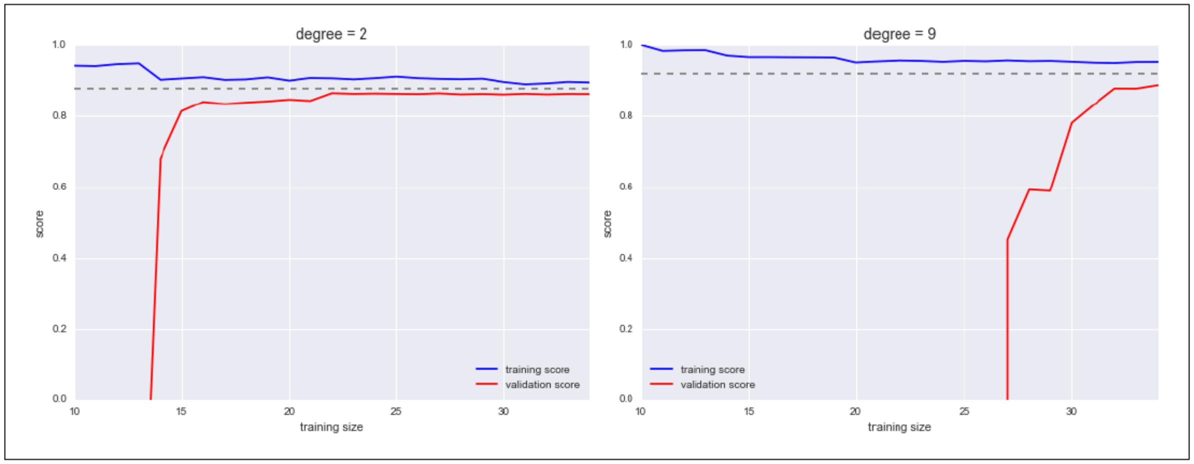
Learning Curves

- As you gather more data, how much better do your models get?
- This can guide the decision to collect more data.



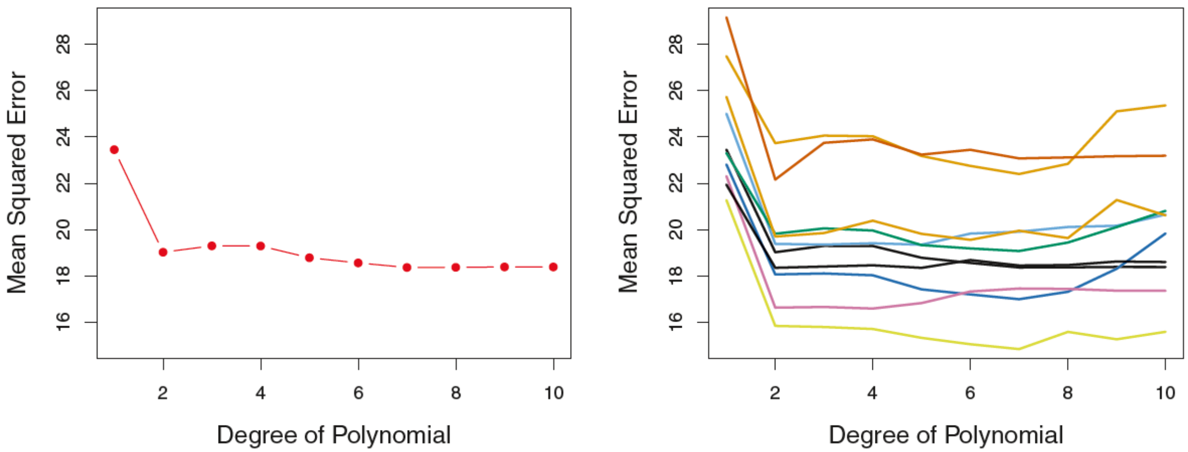
Learning Curves

- Models of different complexities have different learning curves
- Larger models don't saturate as quickly. They are,
  - worse than small models on small datasets
  - better than small models on large datasets



Evaluation and Randomness

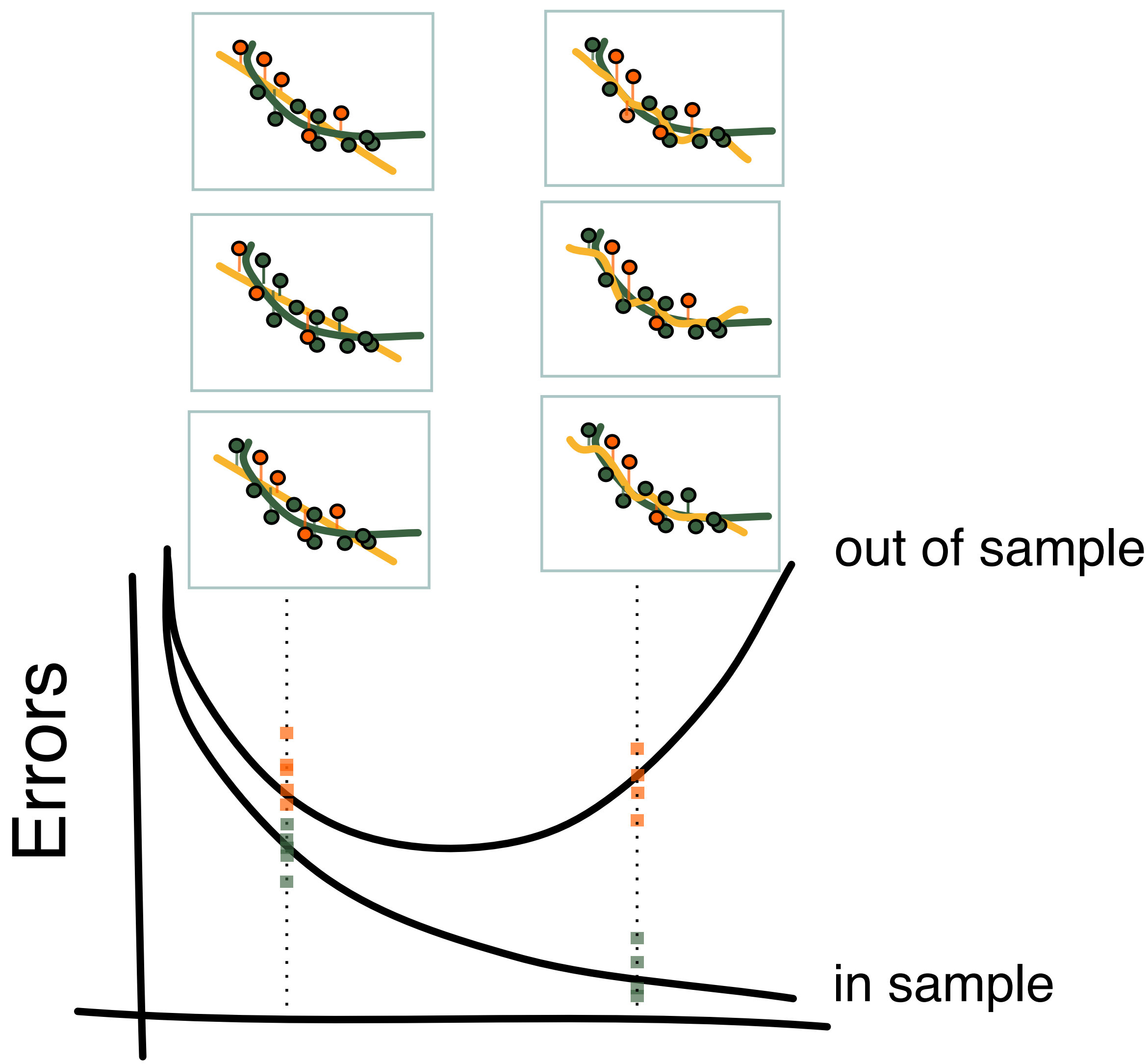
- We are only *estimating* out-of-sample error
- These estimates might be good or bad
  - Have randomness from choice of validation set
  - Have randomness from dataset collection



Evaluation and Randomness

Evaluation and Randomness

- We are only *estimating* out-of-sample error
- These estimates might be good or bad
  - Have randomness from choice of validation set
  - Have randomness from dataset collection



Model Complexity

Bias and Variance in Validation Error

- Variance: Different validation sets give different estimates
- Bias: Training on subset leads to worse expected performance (remember learning curves)
- Bias: You might overfit to the test set

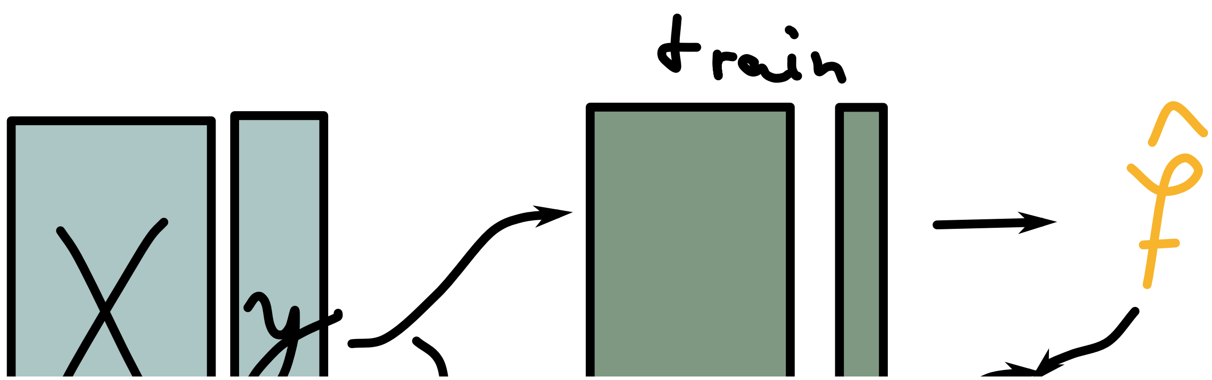
There are a few alternatives to validation sets. We'll talk about,

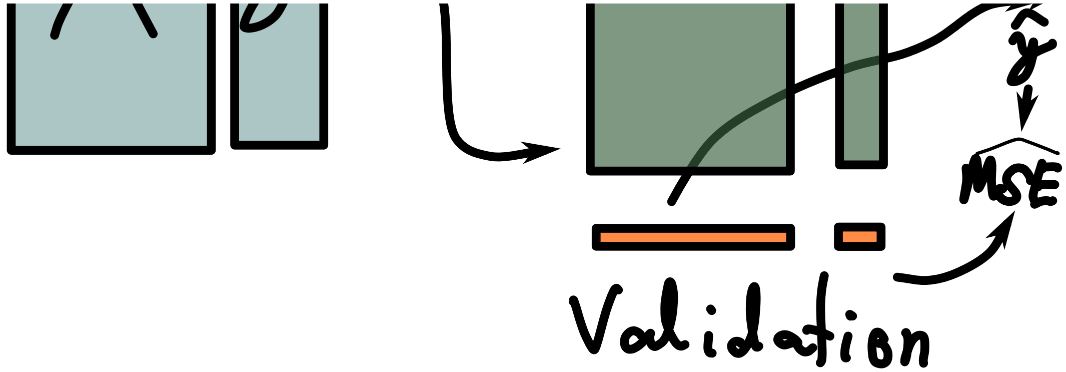
- Leave-One-Out Cross Validation [LOOCV]
- K-Fold Cross Validation.

Alternatives: LOOCV

1. Fit your model without sample (x_i, y_i). Call the fit $\hat{f}_{-i}$.
2. Compute holdout $\widehat{MSE}_i := \left(y_i - \hat{f}_{-i}(x_i)\right)^2$
3. Estimate the out-of-sample error by averaging this over all possible holdouts coming from (1) and (2),

$$\widehat{MSE} = \frac{1}{n}\sum_{i=1}^{n}\widehat{MSE}_i$$



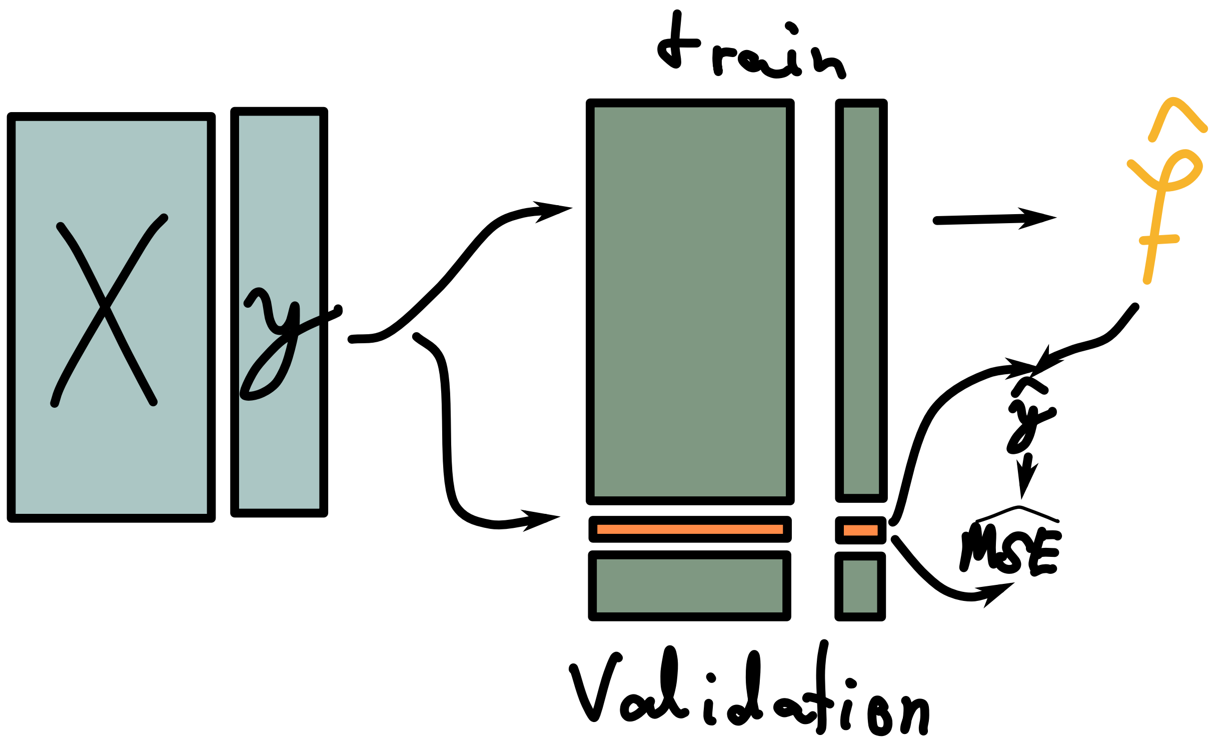


Alternatives: LOOCV

1. Fit your model without sample (x_i, y_i). Call the fit $\hat{f}_{-i}$.
2. Compute holdout $\widehat{MSE}_i := \left(y_i - \hat{f}_{-i}(x_i)\right)^2$
3. Estimate the out-of-sample error by averaging this over all possible holdouts coming from (1) and (2),

$$\widehat{MSE} = \frac{1}{n}\sum^{n}\widehat{MSE}_i$$

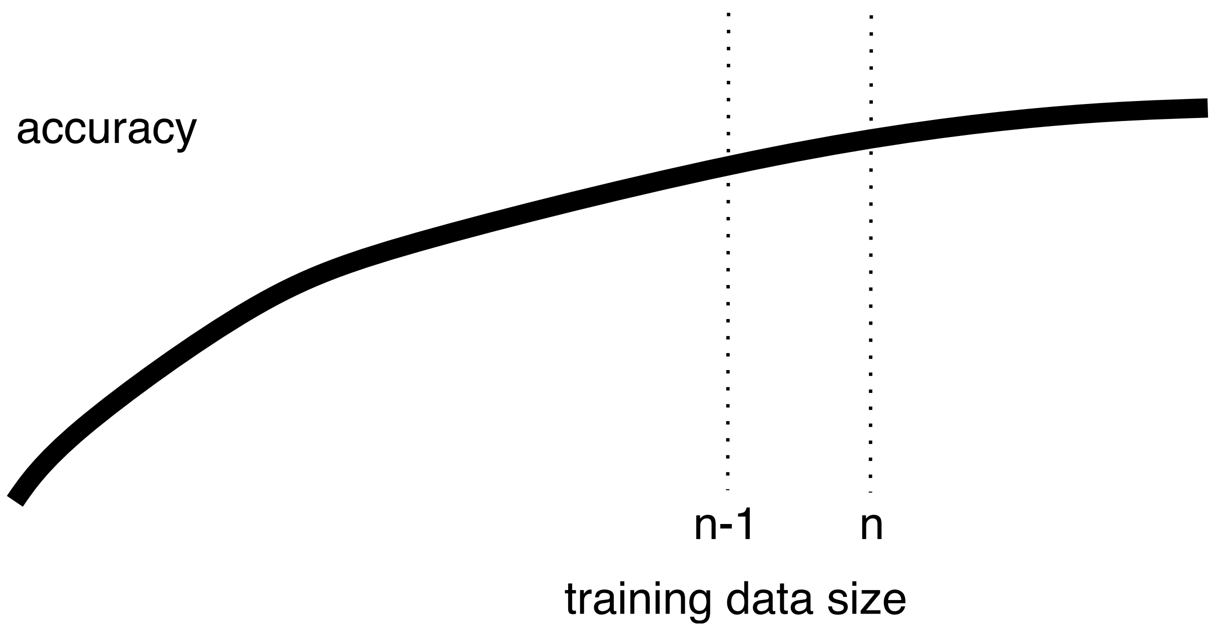
$$n \quad \sum_{i=1}$$



LOOCV

Advantages

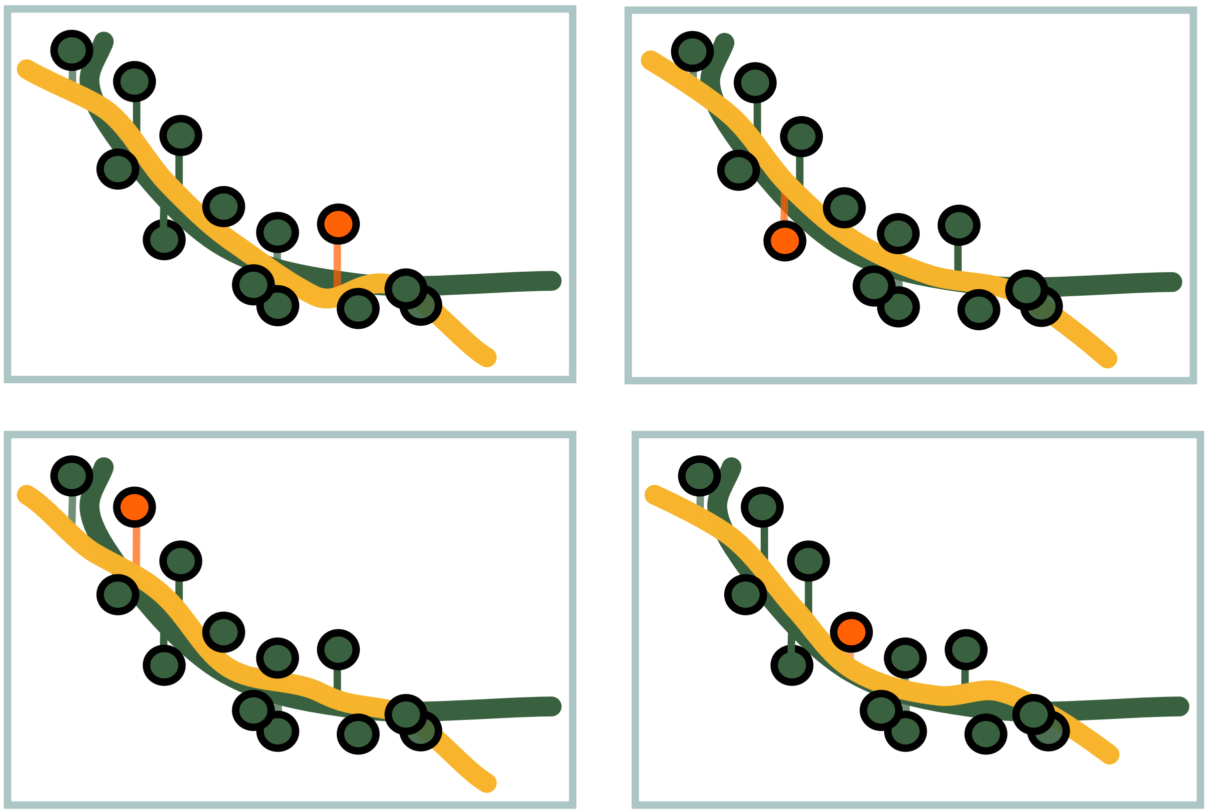
- Lower bias. We use almost all the training data, so we don't underestimate performance.



LOOCV

Disdvantages

- High computational complexity (except linear regression)
- The trained models are correlated
  - The $\widehat{MSE}_i$ are correlated
  - The average of correlated variables has larger variance than the average of independent ones
  - The out-of-sample estimate has higher variance



Alternatives: K-Fold CV

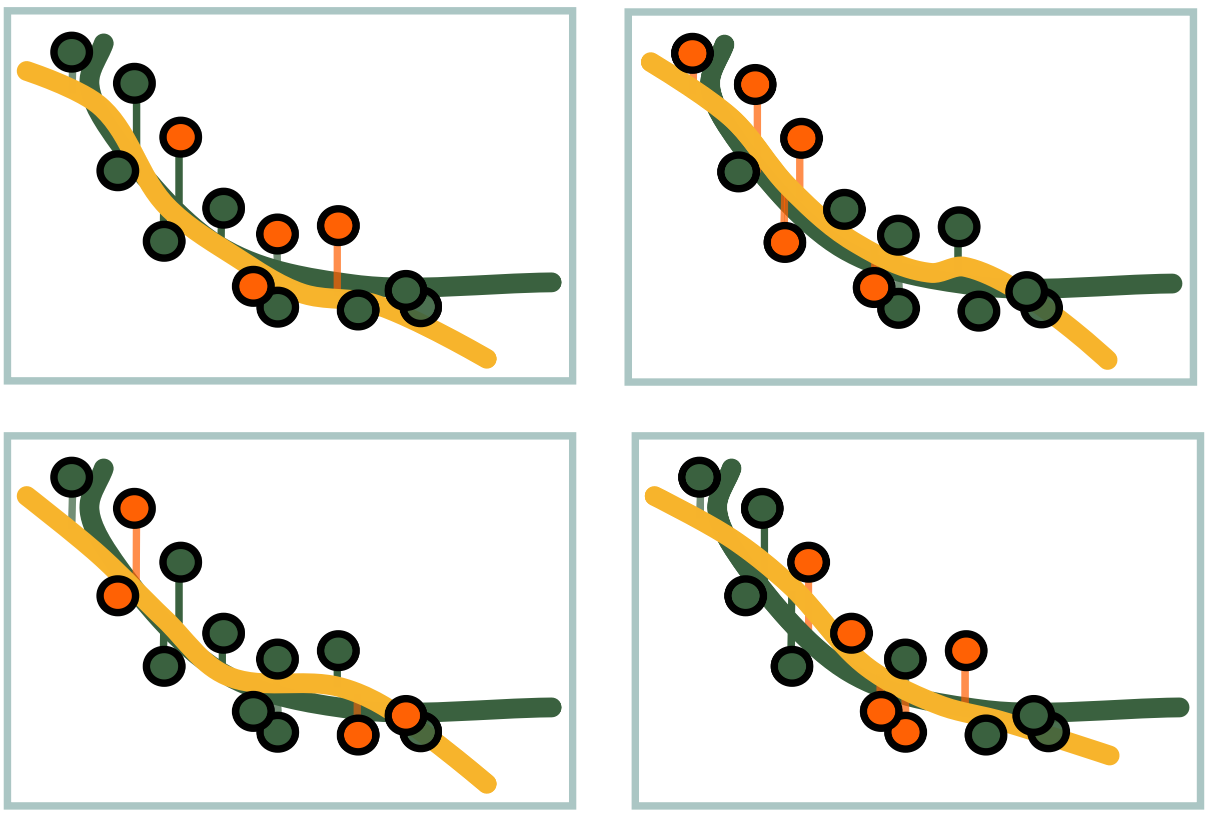
1. Randomly partition samples into one of K folds, $\{S_1, \ldots, S_K\}$.
2. Fit your model without fold S_k. Call the fit $\hat{f}_{-k}$.
3. Compute holdout $\widehat{MSE}_k := \sum_{i \in S_k} \left(y_i - \hat{f}_{-k}\left(x_i\right)\right)^2$
4. Estimate the out-of-sample error by averaging over folds,

$$\widehat{MSE} = \frac{1}{K} \sum_{k=1}^{K} \widehat{MSE}_k$$

K-Fold CV

Advantages

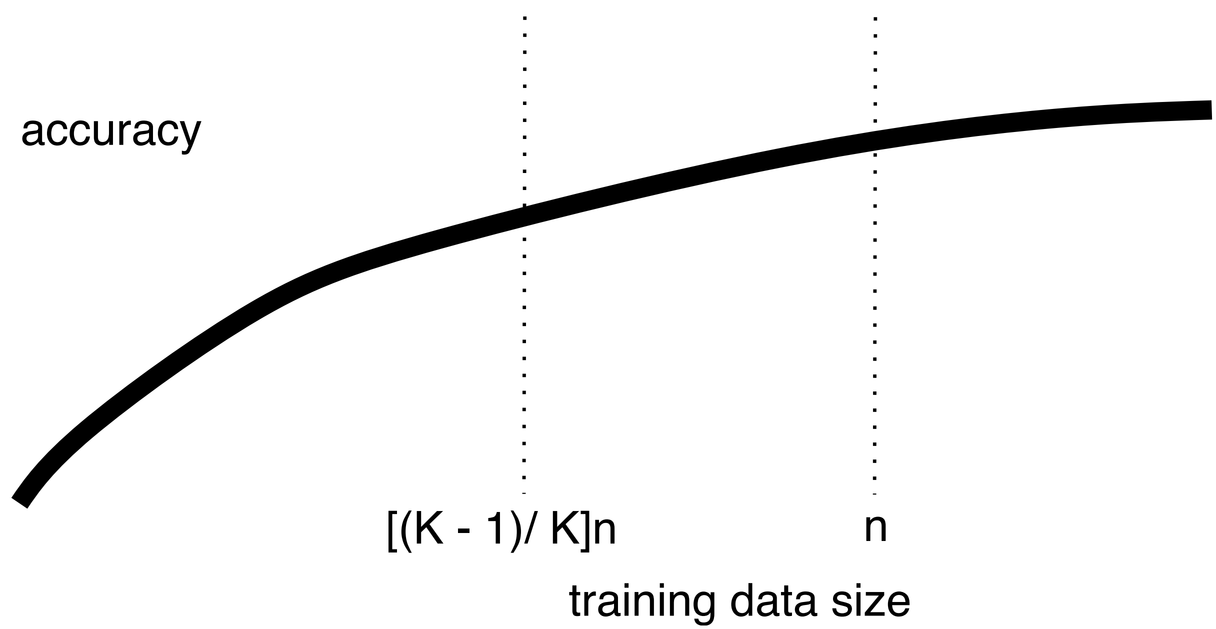
- More computationally tractable
- Learns less correlated models
  - The estimates $\widehat{MSE}_k$ are less correlated
  - The estimate $\widehat{MSE}$ has lower variance



K-Fold CV

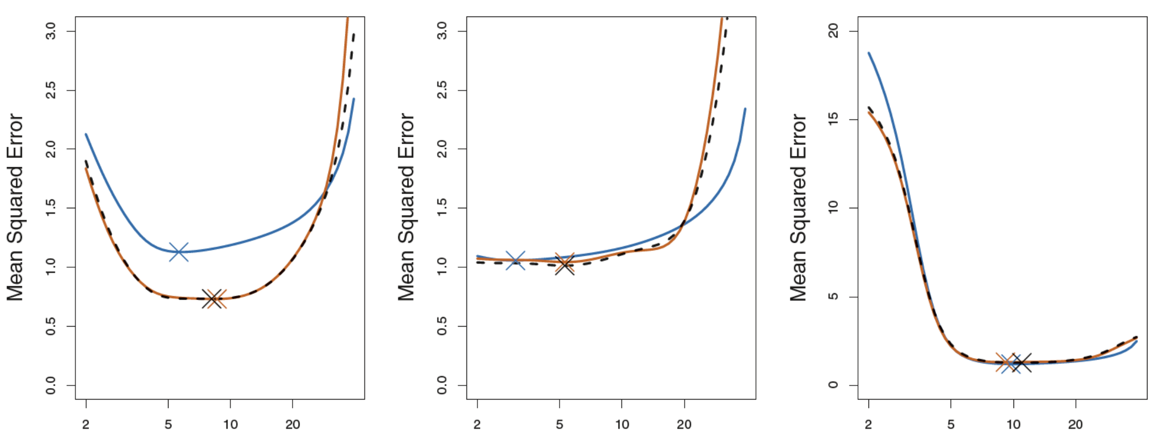
Disadvantages

- We don't train using the full training set
- We bias our estimates upwards
  - Model on full dataset is actually better than estimated



Estimation Quality: LOOCV and K-Fold

- The blue curves are known out-of-sample MSE's from a simulation experiment
- Black and orange are LOOCV and K-Fold estimates, respectively
- Note: Even when estimates of out-of-sample MSE is poor, the estimate of the minimum might be good



Flexibility Flexibility Flexibility

Hyperparameter Search

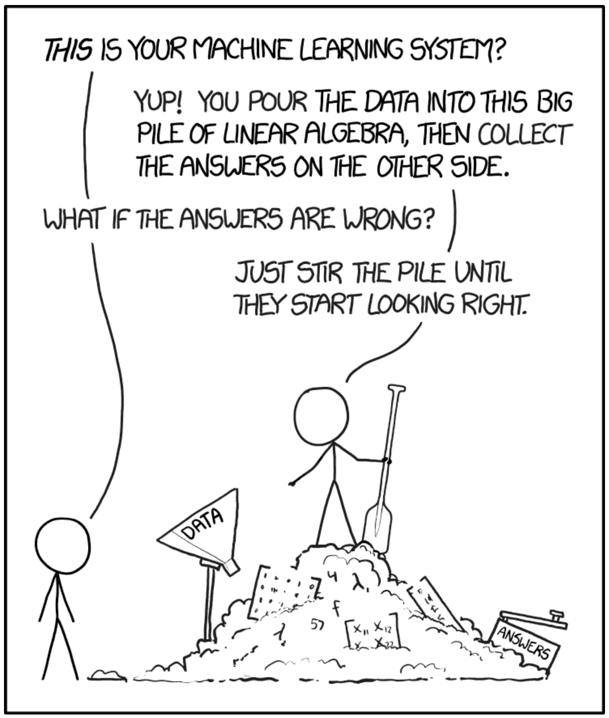
- We will often have many parameters to tune simultaneously
  - Model parameters: Polynomial degree, # trees, ...
  - Training parameters: Learning rate, subsampling, ...
  - Preprocessing: Normalization, outlier removal, ...
- No single "model complexity" parameter

Search Options

- Manual search
- Grid search
- Random search
- Combinations of these

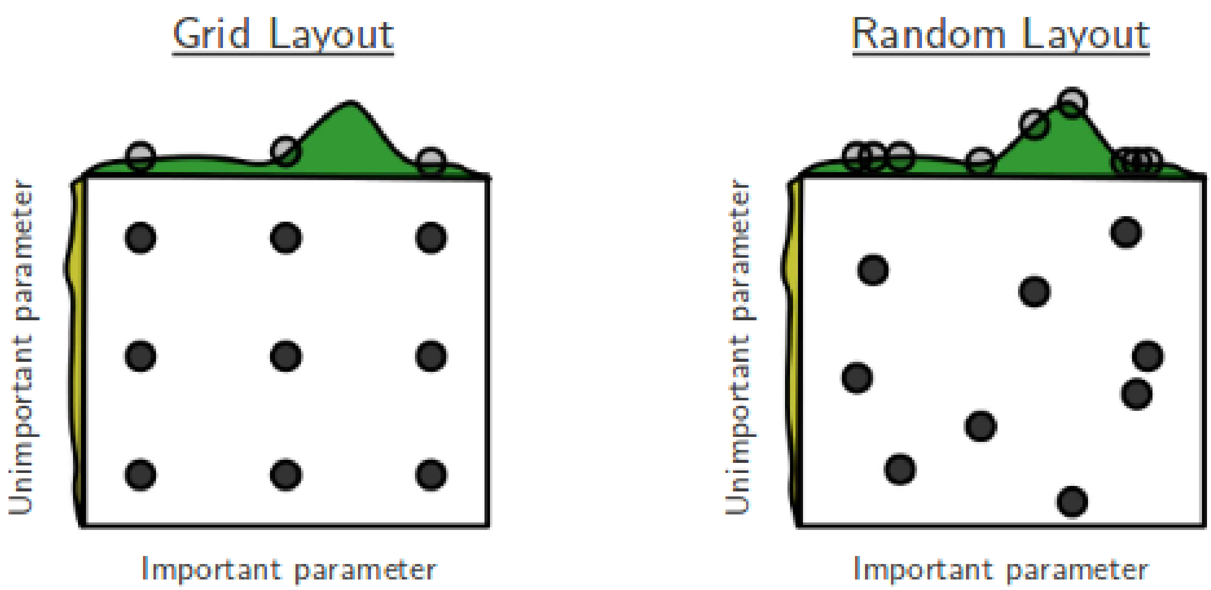
Manual Search

- Relate all the parameters to overall model complexity
  - e.g., more iterations $\rightarrow$ higher complexity
- Guide your choice of parameters by which regime (over vs. underfitting) you are in
- Advantage: Uses bias-variance tradeoff information
- Disadvantage: Tedious, not fully reproducible



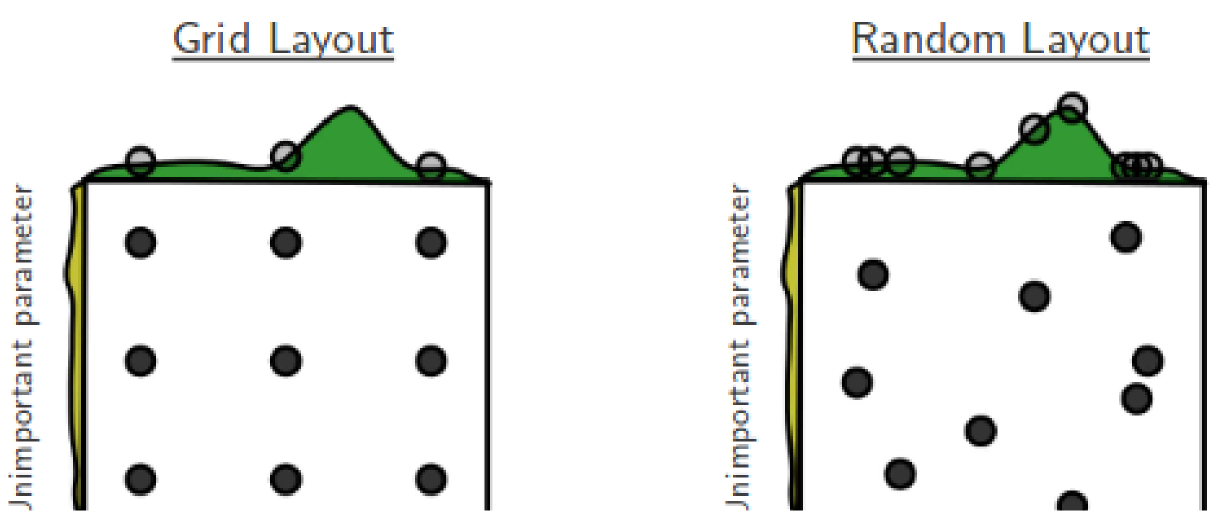
Grid Search

- Compute out-of-sample error on all combinations of parameters
- Advantage: Automatic, easy to implement
- Disadvantage: Exponentially many parameter configurations



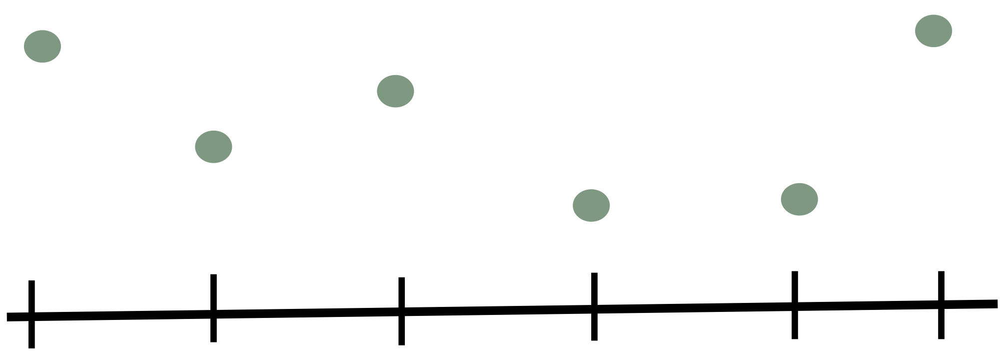
Random Search

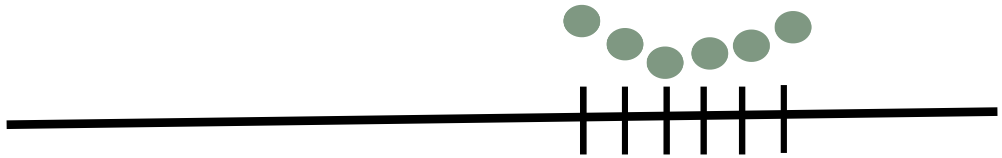
- Compute out-of-sample error on random samples of parameters
- Advantage: Automatic, easy to implement. Relevant parameters become clear quickly.
- Disadvantage: Still suffers when very many parameters.



Combinations

- Can fix a few parameters manually, and use random search for others
- Can use "multiscale" search. Automatically search over predefined grids, but manually set the grids to more promising regions.





```
import {slide} from @mbostock/slide
```

```
<style>
```

```
import {mtex} from @krisrs1128/function-fitting
```

```
import {mtex_block} from @krisrs1128/function-fitting
```