Foundations of Data Science
Lecture 4, Module 4
Fall 2022

Rumi Chunara, PhD

Model Selection

Goal of Model Selection

The goal of model selection is to **achieve the best generalization while avoiding overfitting**

Given a loss function L applied over predicted and ground-truth values, we can compute the following average errors:

$$R_{train} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{L}(f(x_i^{train}), y_i^{train})$$

Separate a percentage of the instances with ground-truth values to use as test (also known as holdout)!

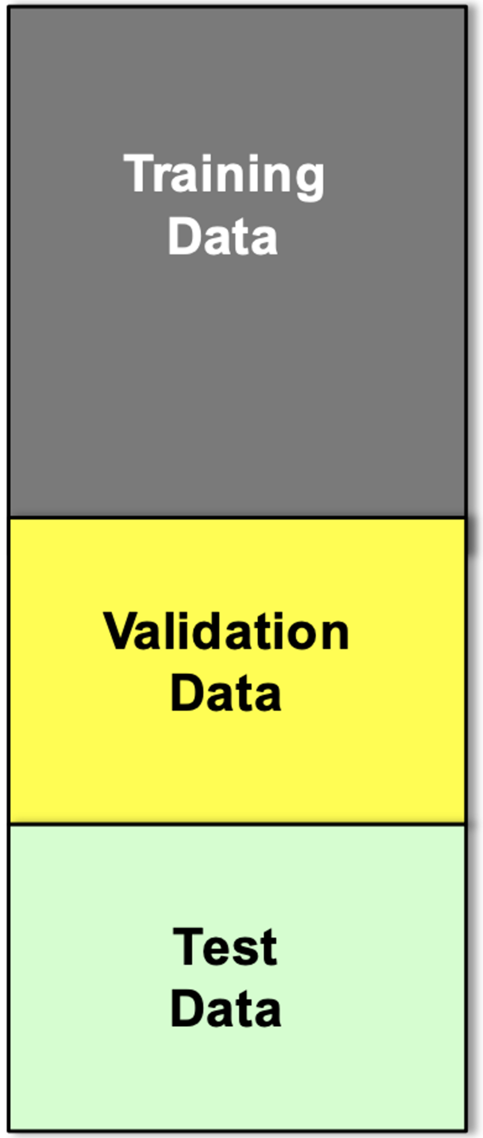
$$R_{test} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{L}(f(x_i^{test}), y_i^{test})$$

Model Selection

When picking a model for a supervised problem we often have many choices. We use test evaluation methodologies to optimize the following:

- Algorithm selection (e.g., Decision Tree vs. kNN vs. SVM)
- Feature selection (e.g., Pearson correlation, wrapper methods, dimensionality reduction)
- Hyper-parameter selection (e.g., *k* in kNN, or MinLeafSize in Decision Trees)

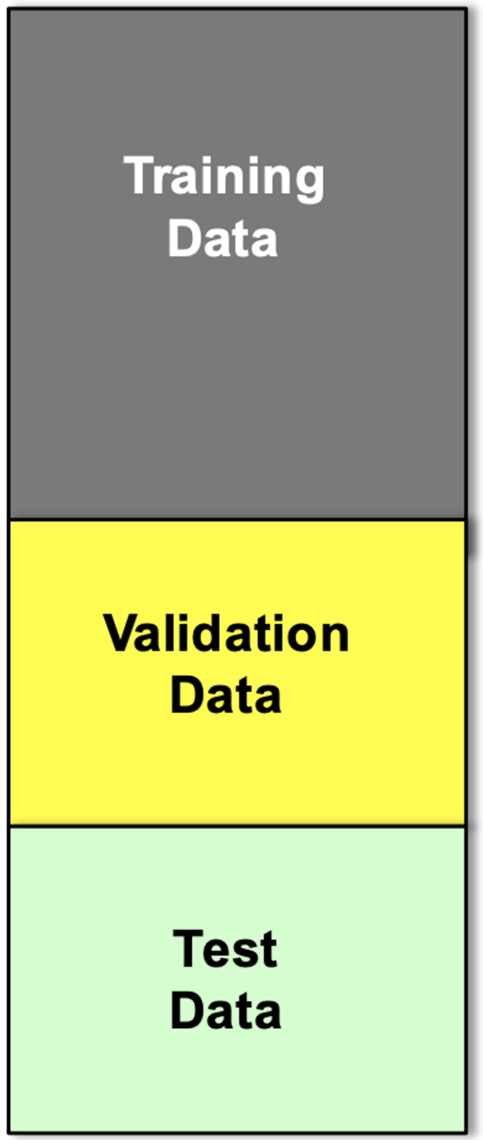
Basic Design for Test Evaluation



When doing any sort of model selection, one usually splits the data in 3 parts:

- **Training data** - used to find the optimal function given the model structure (i.e., fixed algorithm, feature set, hyper-parameters)
- **Validation data** - used to evaluate the loss for different model configurations. The configuration with the best loss is selected as the final model
- **Test data** - not used for any parameter or model selection. It is only used to measure generalization

Basic Design for Test Evaluation



Common ratios used:

- 70% train, 15% val, 15% test
- 80% train, 10% val, 10% test
- 60% train, 20% val, 20% test

Model Selection in Practice

Define: Training Data
Define: Validation Data
Define: Test Data

1. For each configuration c in the set C=[Algorithm x Feature Set x Hyper-parameter]:

- Find the function $\hat{f}_c$ such that: $\hat{f}_c = \underset{f \in \mathbb{F}}{\operatorname{argmin}} R^{train}$
- With $\hat{f}_c$ estimated, get the validation loss: $R_c^{val} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{L}(\hat{f}_c(x_i^{val}), y_i^{val})$

2. Choose the optimal configuration c_{opt} such that: $c_{opt} = \underset{c \in C}{\operatorname{argmin}(\max)}\ R_c^{val}$

3. Define: NewTrain = Train + Validation data

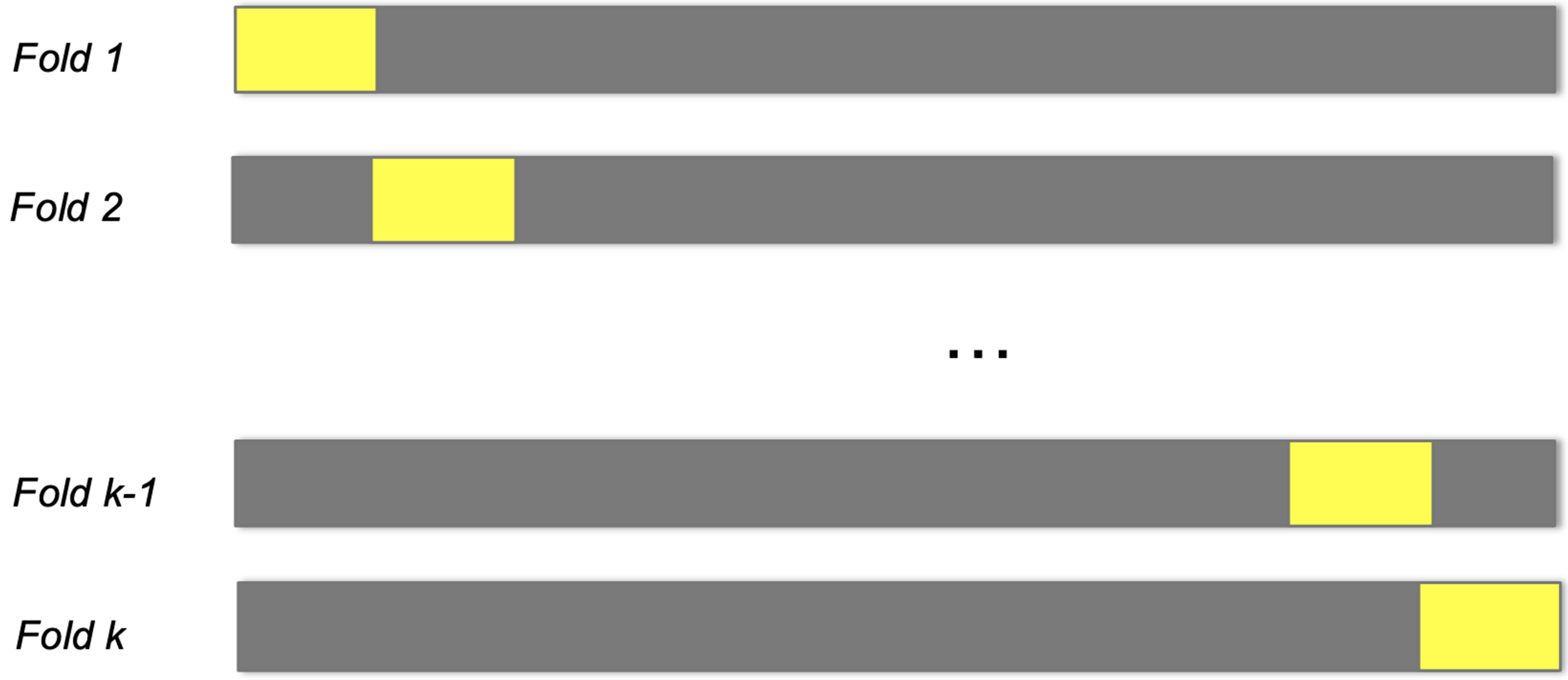
4. Find the function $\hat{f}$ such that: $\hat{f} = \underset{f \in \mathbb{F}}{\operatorname{argmin}} R^{NewTrain}$

5. Estimate the test loss as: $R^{test} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{L}(\hat{f}(x_i^{test}), y_i^{test})$

https://github.com/briandalessandro/DataScienceCourse/blob/master/ipython/Lectures/2020%20Intro%20to%20DS%20Module%2013%20-%20ModelSelection.pdf

Cross Validation

If there is not enough data to create train/validation/test datasets that are large enough for a proper understanding of generalization, "recycle" data by using **k-fold cross validation**.



Here, we only work with training (gray) and test (yellow) data!

Cross Validation - Properties

- Each sample in the datasets participates in both training and testing, **but never at the same time**

- The use of multiple folds allows for the computation of variance statistics and confidence intervals
    - As a consequence, one can use this for statistical inference (data analysis that is not only descriptive!)

Cross Validation - Properties

- Each sample in the datasets participates in both training and testing, **but never at the same time**

- The use of multiple folds allows for the computation of variance statistics and confidence intervals
  - As a consequence, one can use this for statistical inference (data analysis that is not only descriptive!)

Bias/Variance Tradeoff

Bias

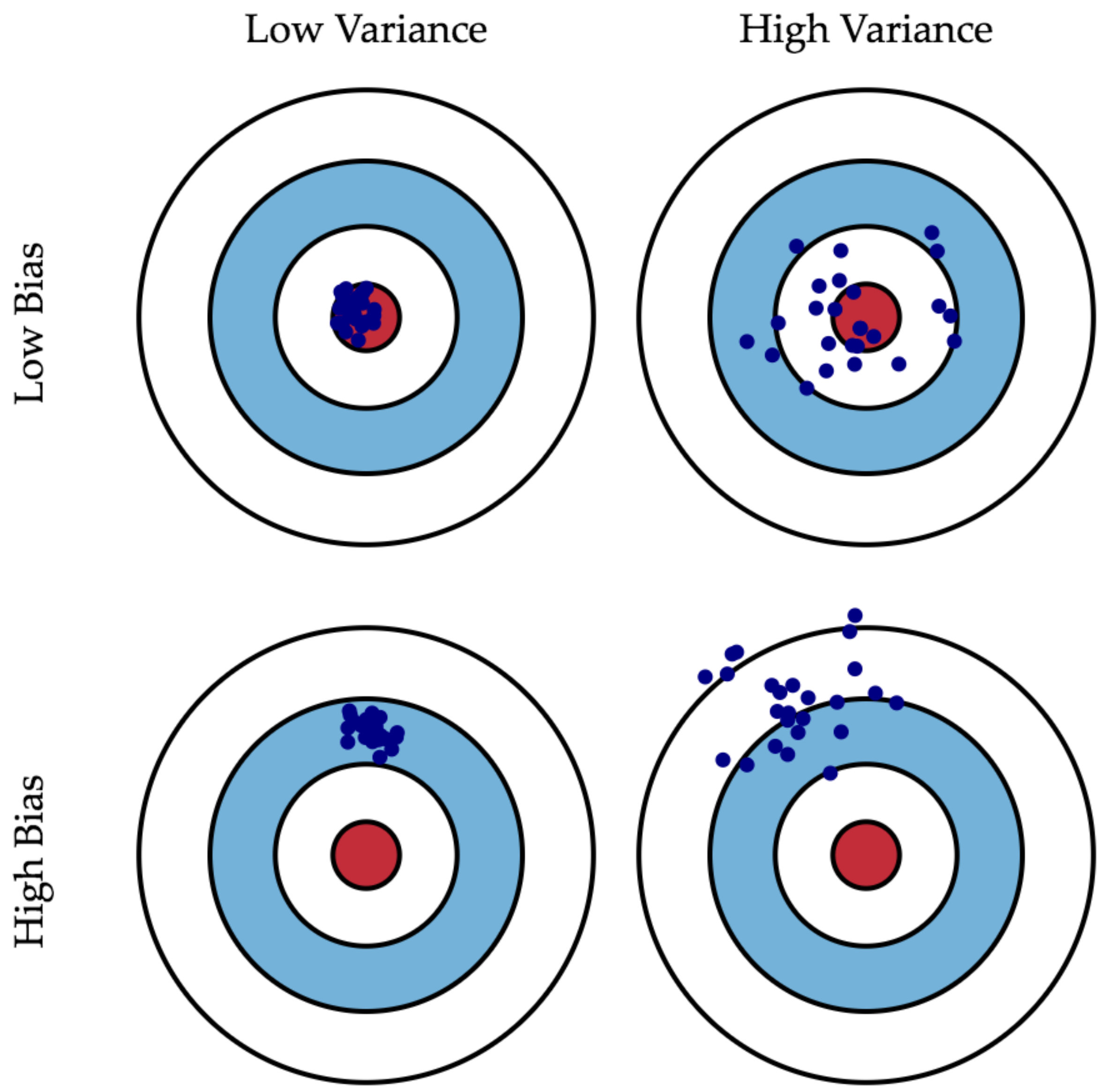
- Low bias
  - linear regression applied to linear data
  - 2nd degree polynomial applied to quadratic data
- High bias
  - constant function
  - linear regression applied to non-linear data

Variance

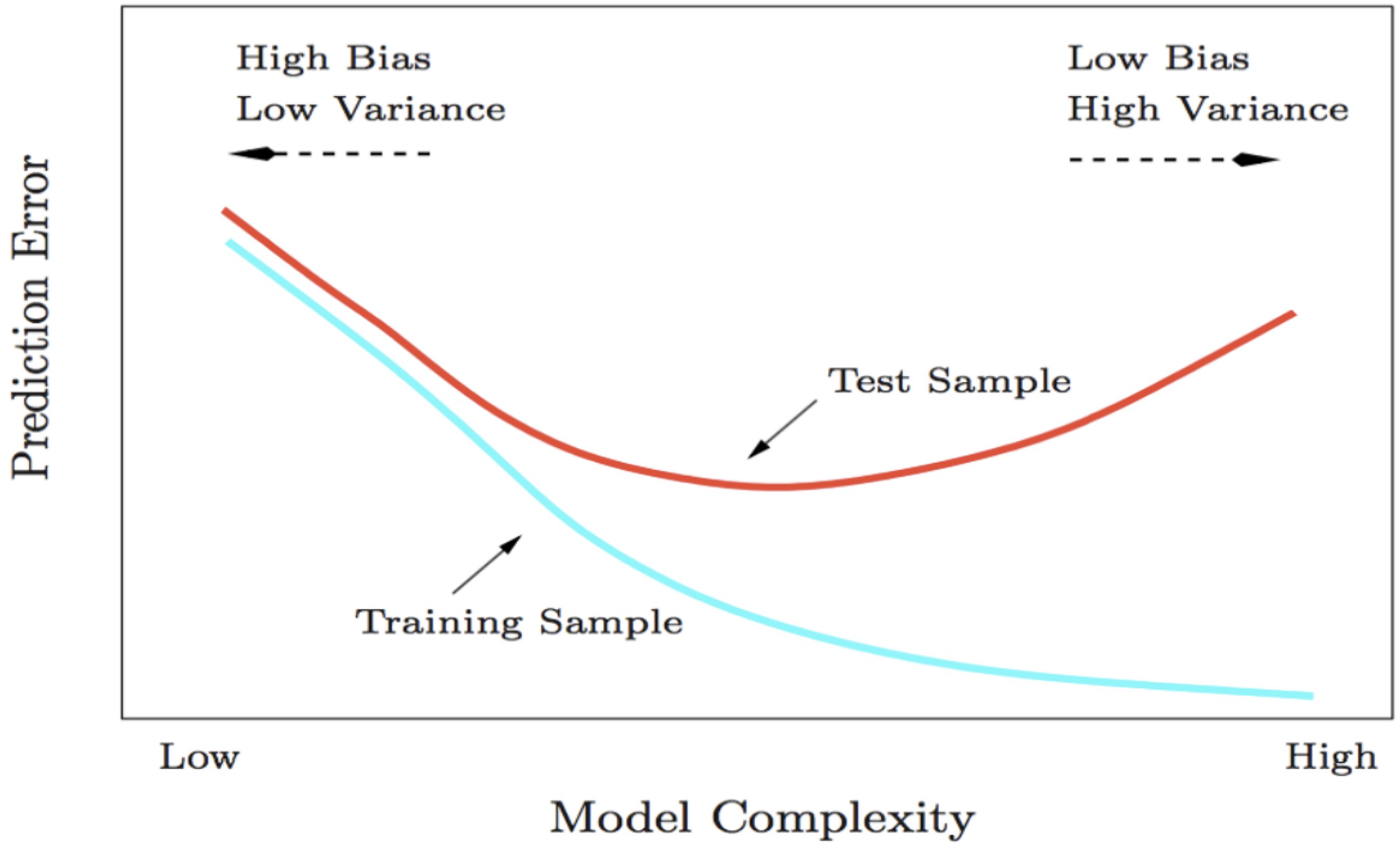
- Low variance
  - constant function
  - model independent of training data
  - model depends on stable measures of data
    - mean
    - median
- High variance
  - high degree polynomial

Sources of Variation

- noise in targets or input attributes
- bias (model mismatch)
- training sample
- randomness in learning algorithm
- randomized subsetting of train set:
  - cross validation, train and early stopping set


Low Variance
High Variance
Low Bias
High Bias

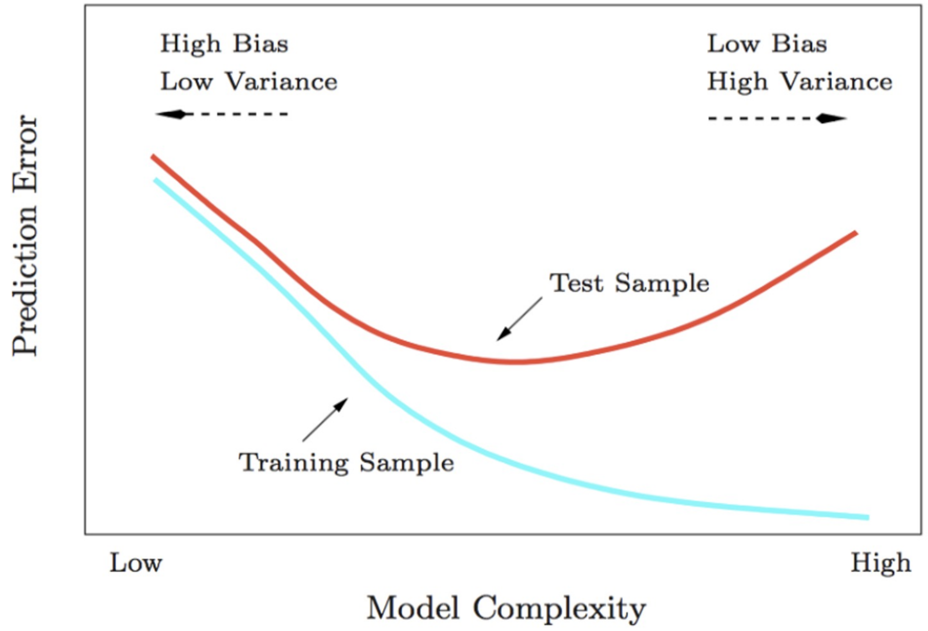
Performance Tradeoffs



Choosing Hyperparameters

Software will do most of the model training work, but the scientist still needs to instruct the software on which hyper-parameters to explore.

Grid search parameter values!



You have 2 main choices to make:
1) the range of values to explore, 2) the actual (and number of) values to explore

Using your best judgment and knowledge of the algorithm and problem, you want hyper-parameters that span the range of low to high model complexity

$\underbrace{\hspace{20em}}$

Range should start here

And end here

https://github.com/briandalessandro/DataScienceCourse/blob/master/ipython/Lectures/2020%20Intro%20to%20DS%20Module%2013%20-%20ModelSelection.pdf

Reduce Variance Without Increasing Bias

- Averaging reduces variance:

$$Var(\overline{X}) = \frac{Var(X)}{N}$$

- Average models to reduce model variance
- One problem:
  - only one train set
  - where do multiple models come from?

Expectation, Bias & Variance

Expectation of model: The expectation of the model is the average of the collection of models estimated over many training datasets.

$$\mathbb{E}[\hat{f}_{\mathcal{D}}(x)]$$

Bias of model: The model bias describes how much the expectation of the model deviates from the true value of the function we are trying to estimate. Low bias signifies that our model does a good job of approximating our function, and high bias signifies otherwise. The bias measures the ***average accuracy*** of the model.

$$\mathbb{E}\left[\hat{f}_{\mathcal{D}}(x)\right] - f(x)$$

Variance of model: The model variance is the expectation of the squared differences between a particular model and the expectation of the collection of models estimated over many datasets. It captures how much the model fits vary across different datasets, so it measures the ***average consistency*** of the model. A learning algorithm with high variance indicates that the models vary a lot across datasets, while low variance indicates that models are quite similar across datasets.

$$\mathbb{E}[(\hat{f}_{\mathcal{D}}(x) - E[\hat{f}_{\mathcal{D}}(x)])^2]$$

Bias/Variance in Generalization

- Squared loss: $L(a,b) = (a-b)^2$
- Consider one data point (x,y)
- Notation:
  - $y = f_{\mathcal{D}}$ (true value)
  - $\hat{y} = \hat{f}_{\mathcal{D}}$ (prediction)
  - $S = (y - \hat{y})^2$ (squared error)

Decomposing Expected Squared Error

$$E(\hat{y}-y)^2 = \mathrm{E}[(\hat{y}-\mathrm{E}[\hat{y}]+\mathrm{E}[\hat{y}]-y)^2]$$

$$=\mathrm{E}[(\hat{y}-\mathrm{E}[\hat{y}])^2 + 2((\hat{y}-\mathrm{E}[\hat{y}]))(\mathrm{E}[\hat{y}]-\hat{y})) + (\mathrm{E}[\hat{y}]-y)^2]$$

$$=\mathrm{E}[(\hat{y}-\mathrm{E}[\hat{y}])^2 + 2\mathrm{E}[(\hat{y}-\mathrm{E}[\hat{y}]))(\mathrm{E}[\hat{y}]-\hat{y})] + \mathrm{E}[(\mathrm{E}[\hat{y}]-y)^2]$$

$$=\mathrm{E}[(\hat{y}-\mathrm{E}[\hat{y}])^2 + 2(\mathrm{E}(\hat{y})-y)\,\mathrm{E}(\hat{y}-\mathrm{E}[\hat{y}]) + \mathrm{E}\,[(\mathrm{E}[\hat{y}]-y)^2]$$

$$\mathrm{E}(\hat{y}-\mathrm{E}[\hat{y}]) = \mathrm{E}(\hat{y})-\mathrm{E}[\hat{y}] = 0$$

$$=\mathrm{E}[(\hat{y}-\mathrm{E}[\hat{y}])^2 + \mathrm{E}\,[(\mathrm{E}[\hat{y}]-y)^2]$$

= Var + Bias^2

Generalization Error (Notation)

- **"True" distribution:** P(x,y)
  - Unknown to us
- **Train:** $\hat{y}$ = y
  - Using training data $\mathcal{D}$ = $\{(x_1,y_1),\ldots,(x_n,y_n)\}$
  - Sampled from P(x,y)
- **Generalization Error:**
  - $\mathcal{L}(h) = E_{(x,y)\sim P(x,y)}[\, f(\hat{y},y)\,]$
  - E.g., $f(a,b) = (a-b)^2$

How do you know that you have a good classifier?

- Is a feature contributing to overall performance?
- Is classifier A better than classifier B?
- Internal Evaluation/Validity:
  - Measure the performance of the classifier.
- External Evaluation/Validity:
  - Measure the performance on a downstream task