Computational Data Analysis

Machine Learning

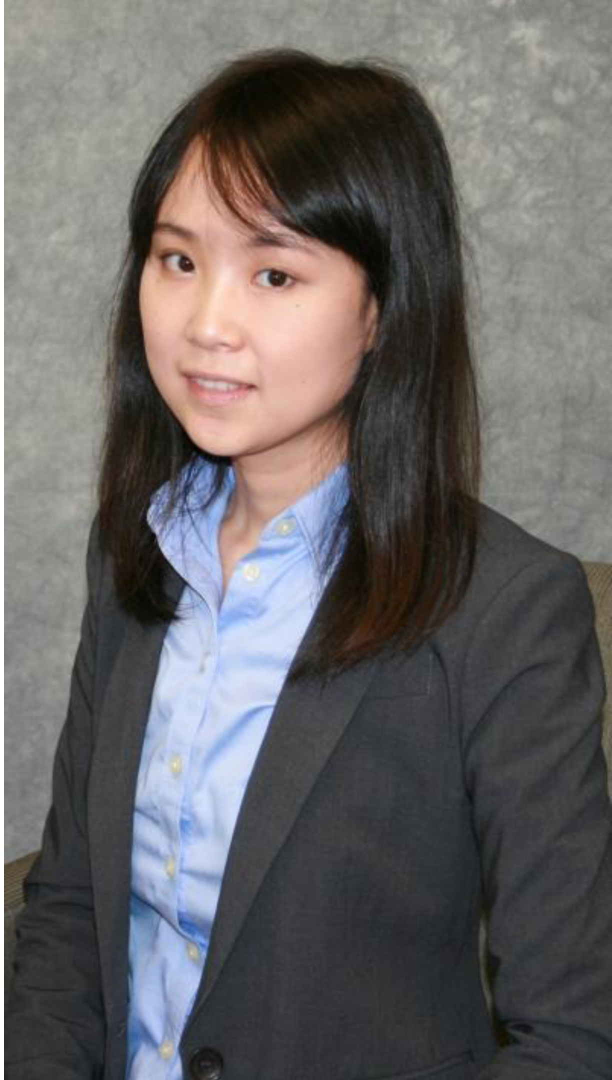
Yao Xie, Ph.D.

Associate Professor

Harold R. and Mary Anne Nash Early Career Professor

H. Milton Stewart School of Industrial and Systems Engineering

Bias-Variance Trade-off and Cross Validation



Outline

- Bias-variance tradeoff: illustration using polynomial regression
- Cross validation
- Bias-variance tradeoff: theory

Fitting polynomial function

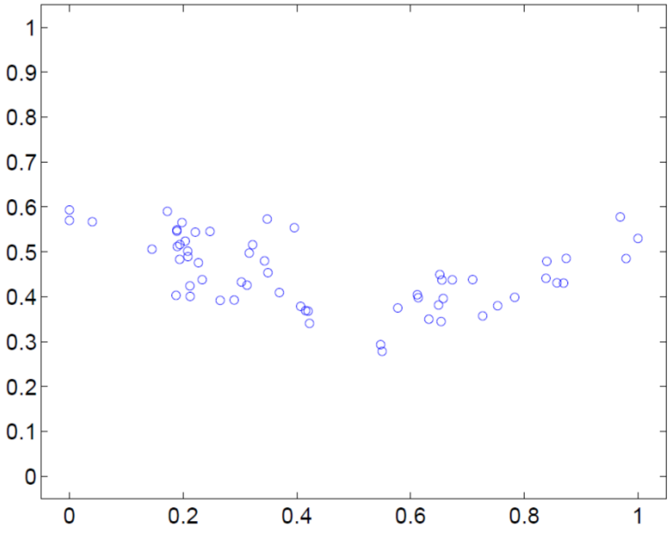
Want to fit a polynomial regression model

$$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_n x^n + \epsilon$$

Let

$$\tilde{x} = (1, x, x^2, \dots, x^n)^\top$$
$$\theta = (\theta_0, \theta_1, \theta_2, \dots, \theta_n)^\top$$
ϵ: noise

$$\Rightarrow \quad y = \theta^\top \tilde{x} + \epsilon$$



Solving using least-square

Given m data points, find θ that minimizes the mean square error

$$L(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(y^i - \theta^\top \tilde{x}^i\right)^2$$

Set gradient to 0 and find parameter

$$\frac{\partial L(\theta)}{\partial \theta} = \frac{1}{m}\sum_{i}^{m}\left(-2y^i\tilde{x}^i + 2\tilde{x}^i{\tilde{x}^i}^T\theta\right) = 0$$

Matrix vector representation

Define $\tilde{X} = (\tilde{x}^1, \tilde{x}^2, \dots \tilde{x}^m)$,

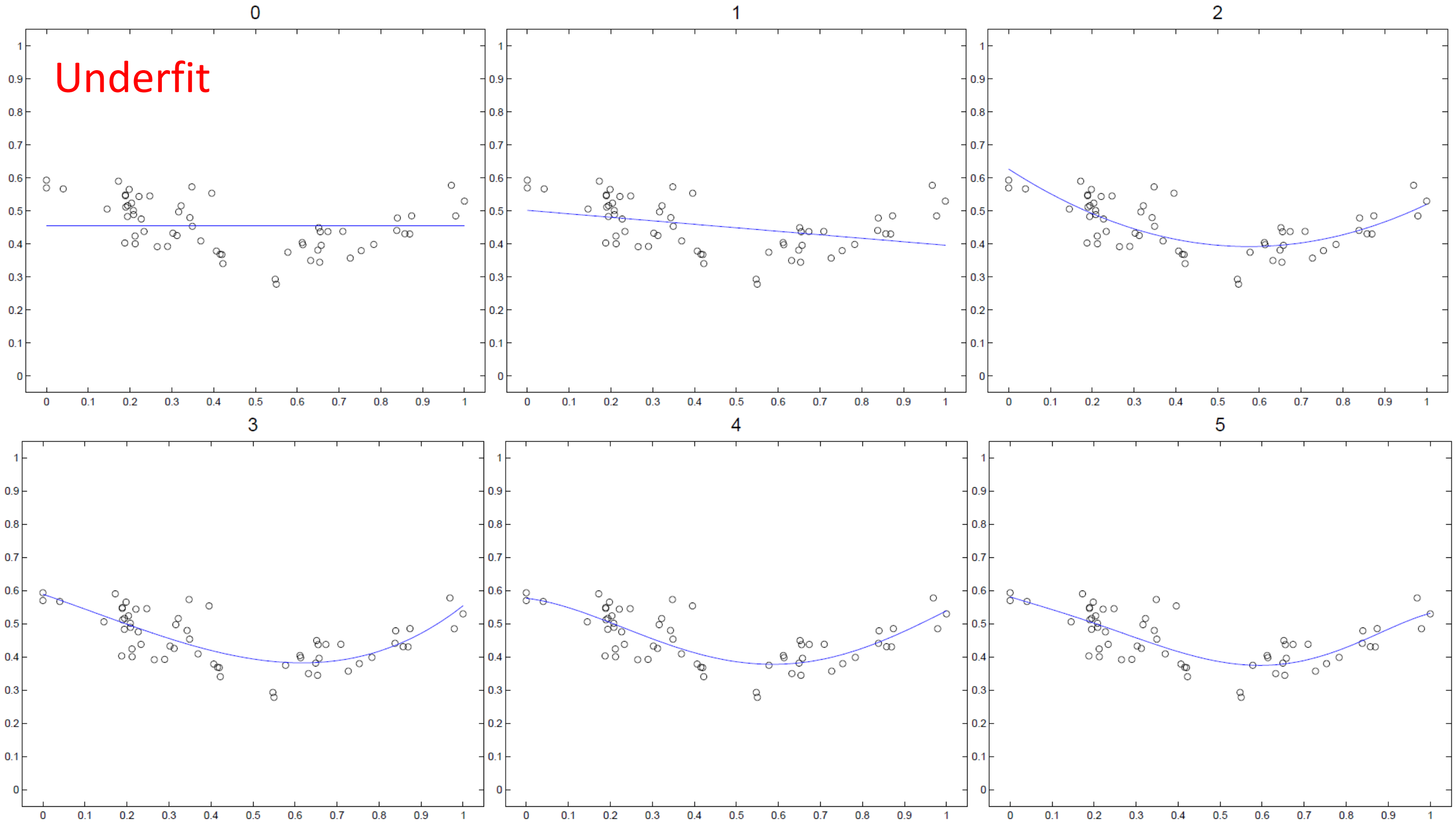
$y \ = (y^1, y^2, \dots, y^m)^\top$

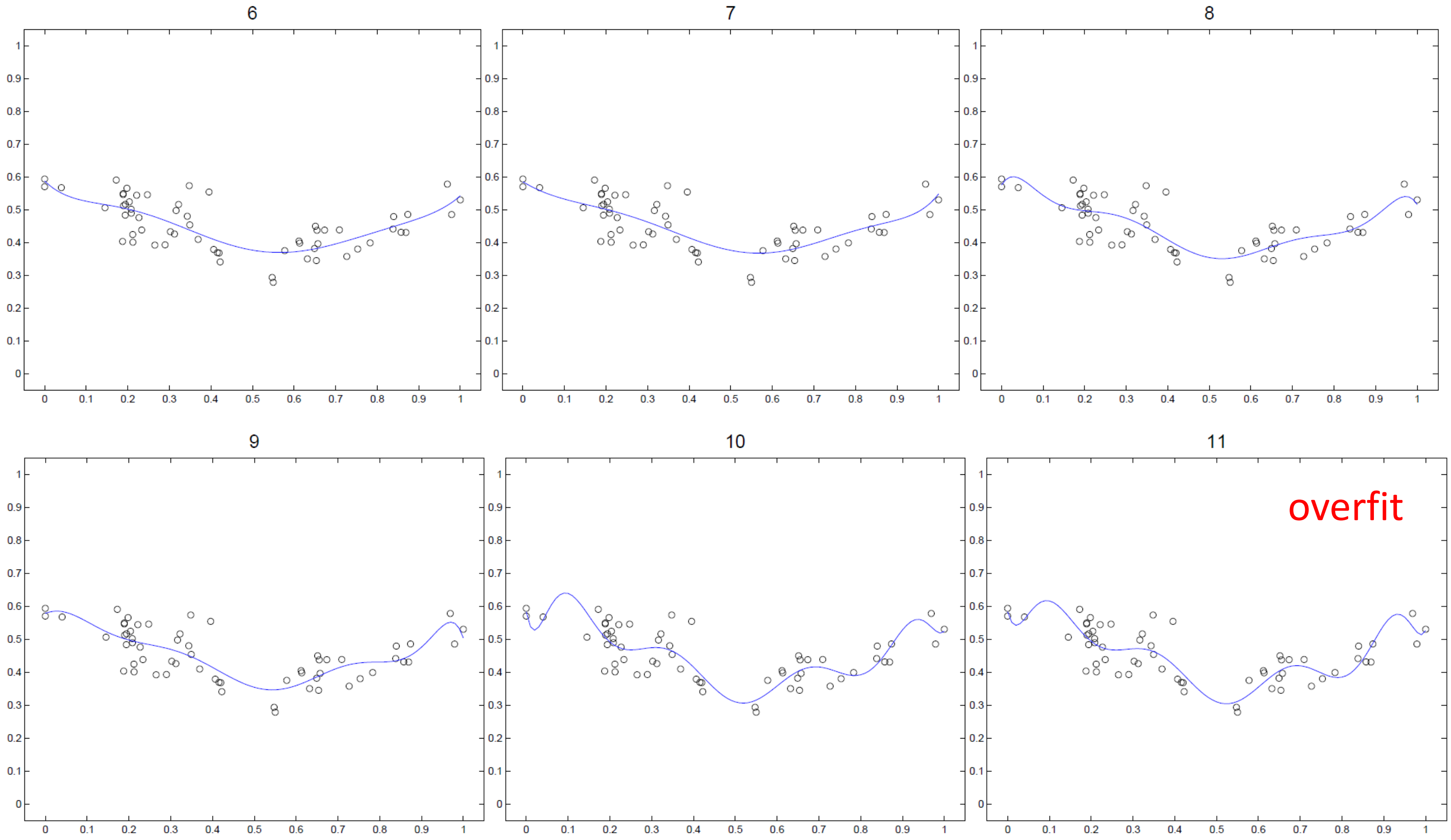
gradient becomes

$$\frac{\partial L(\theta)}{\partial \theta} = -\frac{2}{m}\tilde{X}y + \frac{2}{m}\tilde{X}\tilde{X}^\top \theta = 0$$
$$\Rightarrow \hat{\theta} = \left(\tilde{X}\tilde{X}^\top\right)^{-1}\tilde{X}y$$

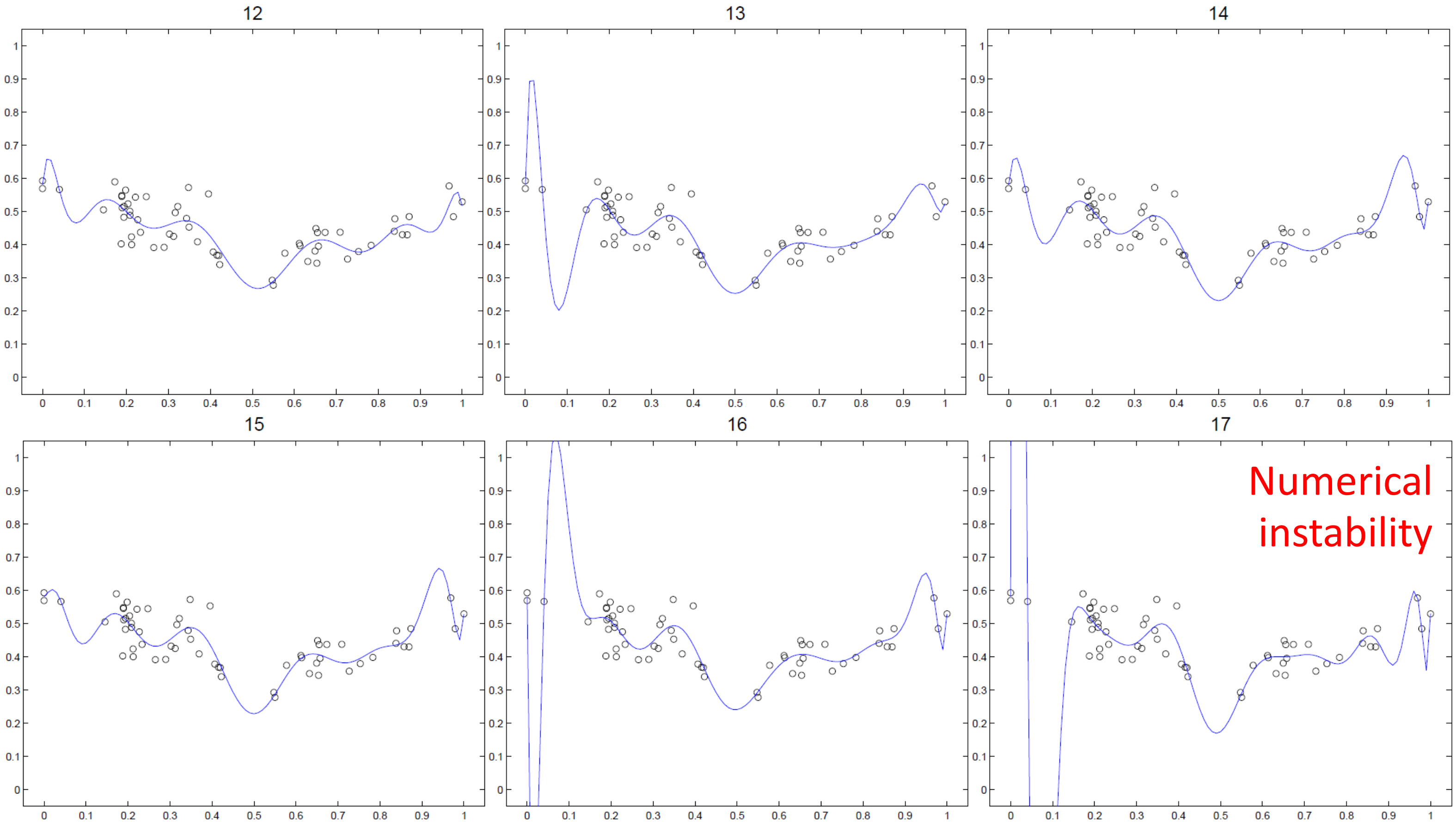
If we choose a different maximal degree n for the polynomial, the solution will be different…

What happens if we increase maximum degree



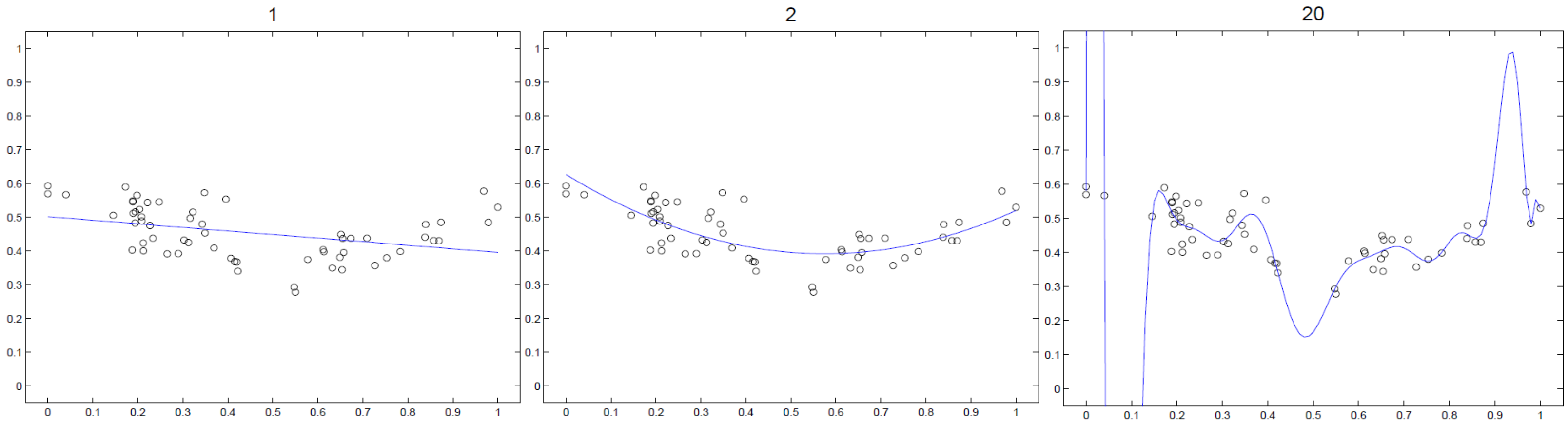


overfit



Numerical instability

How to choose the degree?



- What one is better?

- If we set the maximal polynomial degree to be very large, we can pass through all training points? Is that good?

- No: because the model does not generalize.

What happens here?

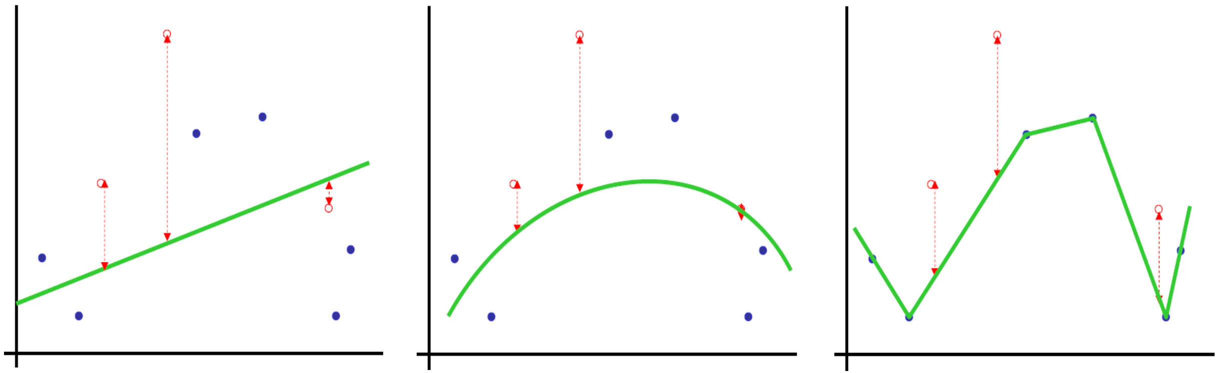
- Numerical issues:

$$\hat{\theta} = \left(\tilde{X}\tilde{X}^\top\right)^{-1}\tilde{X}y, \qquad \tilde{X} = (\tilde{x}^1, \tilde{x}^2, \ldots \tilde{x}^m) \in R^{n\times m}$$

$$\tilde{X}\tilde{X}^\top \in R^{n\times n}, \qquad \mathrm{rank}\left(\tilde{X}\tilde{X}^\top\right) = \min\{n, m\}$$

When $n > m$, $\tilde{X}\tilde{X}^\top$ is not invertible.

- Overfitted model does not generalize to test data



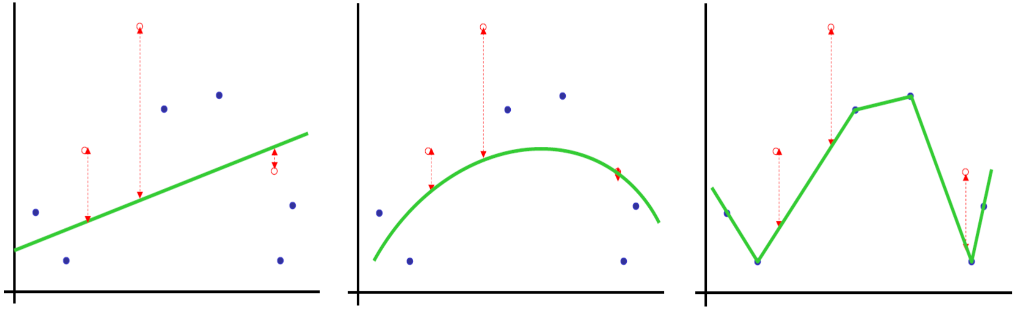
Underfitting

Overfitting

Blue: training data

Red: testing data

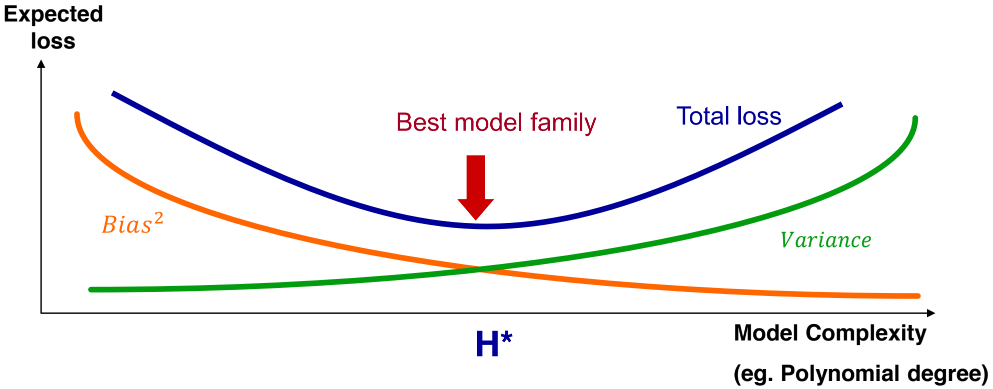
Intuition of bias-variance trade-off



Underfitting

Overfitting

Find the right model family s.t. expected loss becomes minimum



Explain the expected loss (test error) for test data

Cross-Validation (CV): Motivation

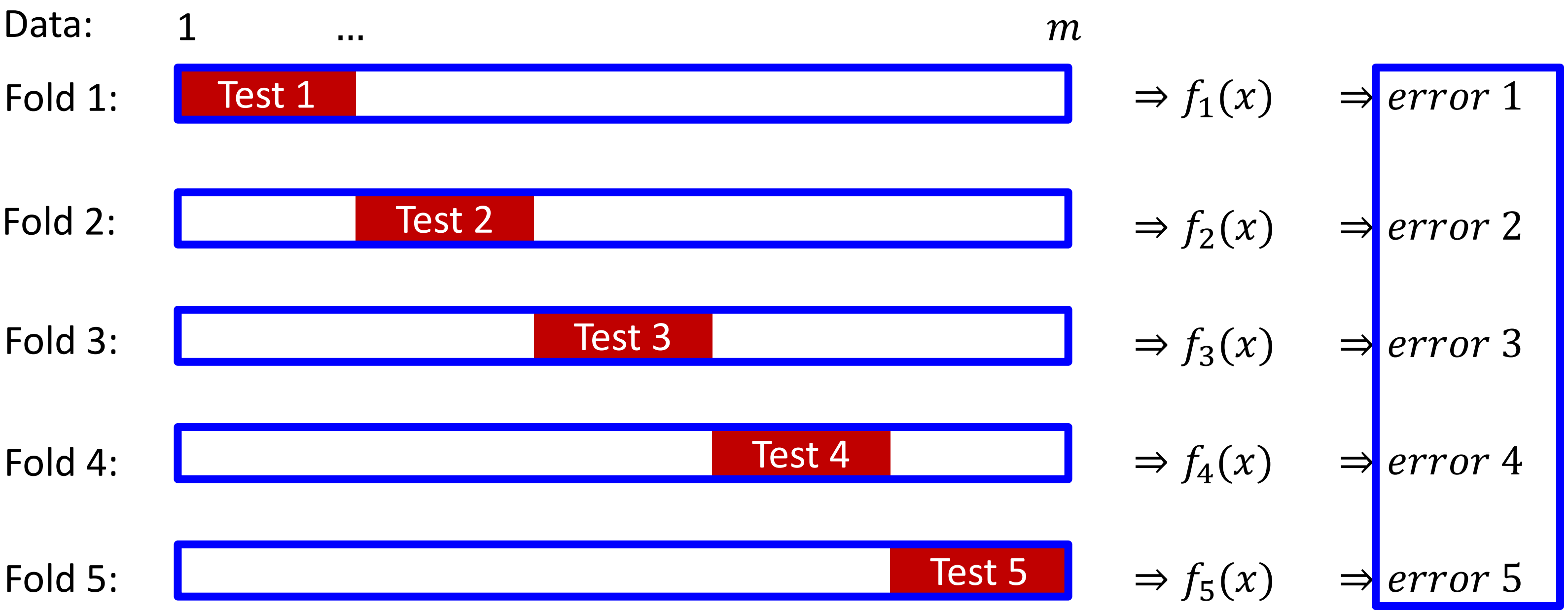
- To have a fair evaluation, use part of data for training, and the rest for testing.

- To use data more efficiently, K-fold cross validation use parts of available data for training, and the rest to validate, and then iterate K times

1	2	3	4	5
Train	Train	Validation	Train	Train

K-fold CV

E.g. 5-fold cross-validation (blank: training; red: test)



Data: 1 … m

Fold 1: Test 1 $\Rightarrow f_1(x)$ $\Rightarrow$ *error* 1

Fold 2: Test 2 $\Rightarrow f_2(x)$ $\Rightarrow$ *error* 2

Fold 3: Test 3 $\Rightarrow f_3(x)$ $\Rightarrow$ *error* 3

Fold 4: Test 4 $\Rightarrow f_4(x)$ $\Rightarrow$ *error* 4

Fold 5: Test 5 $\Rightarrow f_5(x)$ $\Rightarrow$ *error* 5

Important: test data in block k is not used to fit model $f_k(x)$

Average =
cross-validation error

More details: K-fold cross-validation (CV)

For a given model $M(\Theta)$ index by Θ

- Divide training data into K blocks
- For each fold $k, k = 1, \dots, K$
    - Use $(K - 1)$ blocks as training data to fit the model $f_k(x)$
    - Use the remaining 1 block of data (X^k, Y^k) for validation to evaluate performance $L(Y^k, f_k(X^k))$

(choosing a **different** validation block at each time)

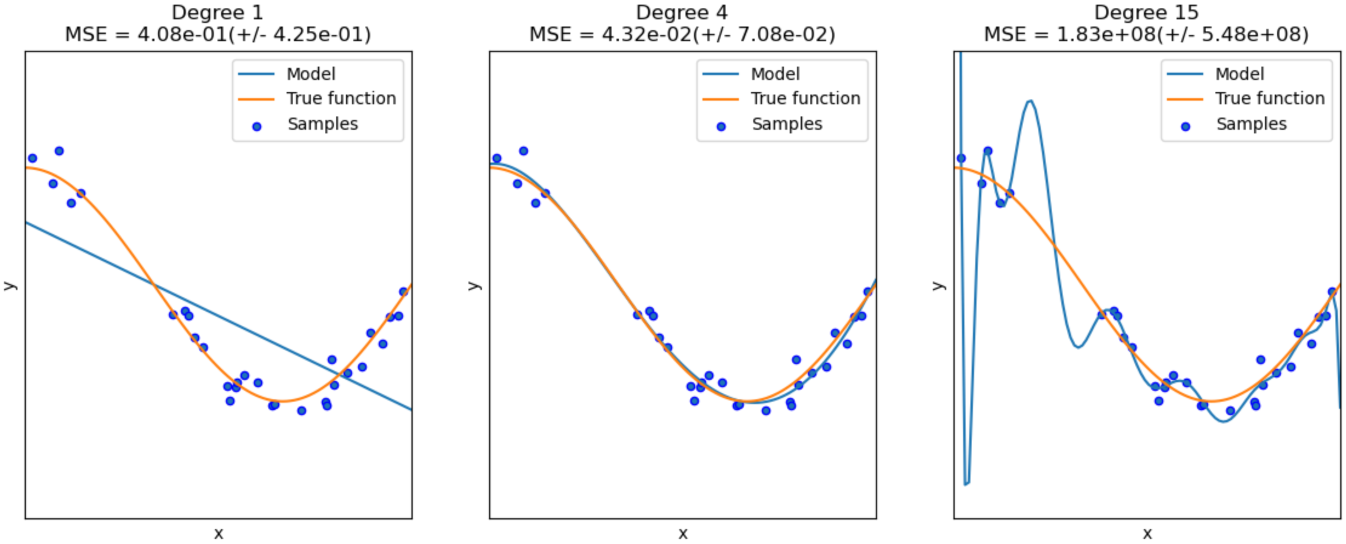
Calculating cross validation error $CV(\Theta) = \frac{1}{K}\sum_{k=1}^{K} L(Y^k, f_k(X^k))$

Choose the Θ that minimizes $CV(\Theta)$

Demo: Fitting polynomial, using MSE for CV

$$MSE = \frac{1}{L}\sum_{l=1}^{L}(\tilde{y}^{l} - \hat{\theta}^{T}\tilde{x}^{l})^{2}$$

Error for each fold (L samples)



https://scikit-learn.org/stable/auto_examples/model_selection/plot_underfitting_overfitting.html#sphx-glr-auto-examples-model-selection-plot-underfitting-overfitting-py

Example: CV for ridge regression

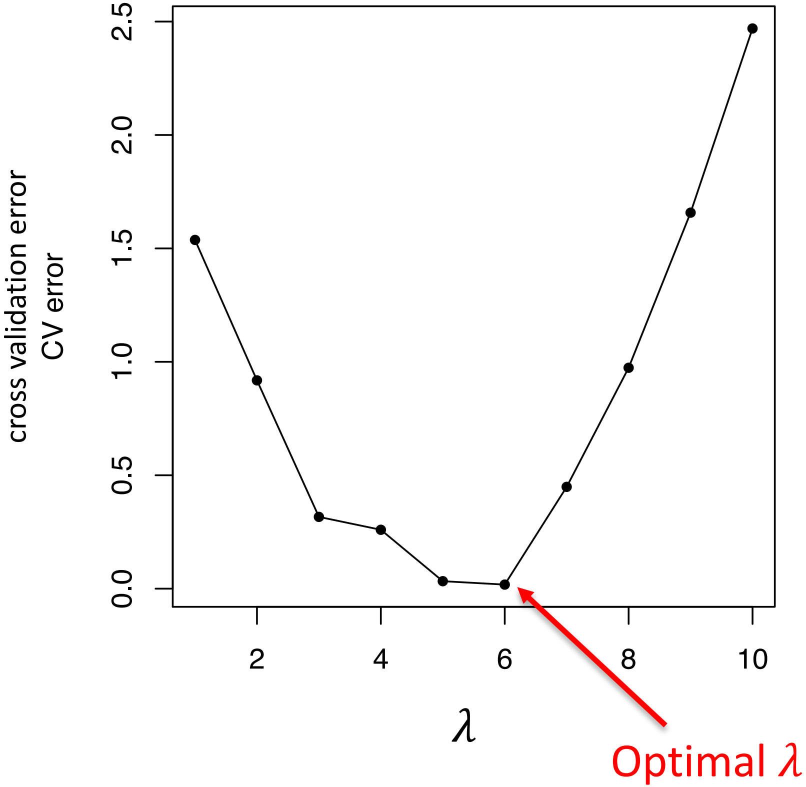
$$\min_{\theta} L(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(y^i - \theta^\top x^i\right)^2 + \lambda\|\theta\|^2$$

$$\theta = \left(\frac{1}{m} XX^T + \lambda I\right)^{-1}\left(\frac{1}{m} Xy\right)$$

Error for each fold (L samples)

$$MSE = \frac{1}{L}\sum_{l=1}^{L}(\tilde{y}^l - \hat{\theta}^T\tilde{x}^l)^2$$

CV Error = Average of error over all folds



Demo: CV for diabetes dataset using lasso

Ten baseline variables, age, sex, body mass index, average blood pressure, and six blood serum measurements were obtained for each of 442 diabetes patients, as well as the response of interest, a quantitative measure of disease progression one year after baseline.

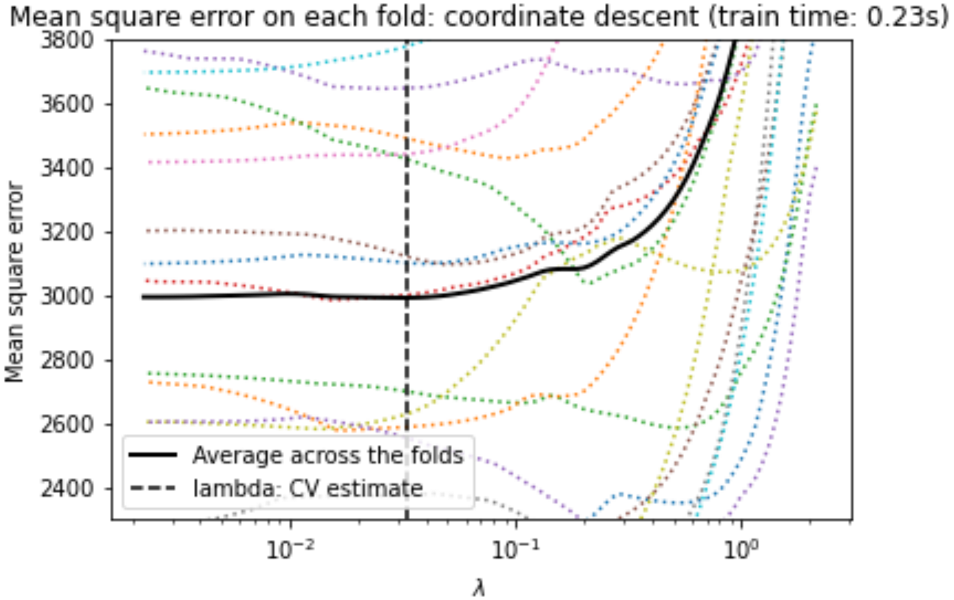
https://scikit-learn.org/stable/auto_examples/feature_selection/plot_select_from_model_diabetes.html#select-from-the-model-features-with-the-higest-score

https://www4.stat.ncsu.edu/~boos/var.select/diabetes.html

Demo: CV for diabetes dataset using lasso

$$\frac{1}{m}\sum_{i=1}^{m}\left(y^i - \theta^\top x^i\right)^2 + \lambda\|\theta\|_1$$

20-fold cross validation
Optimal $\lambda \approx 0.03$

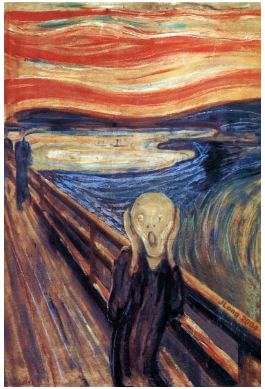


Practical issues for K-fold CV

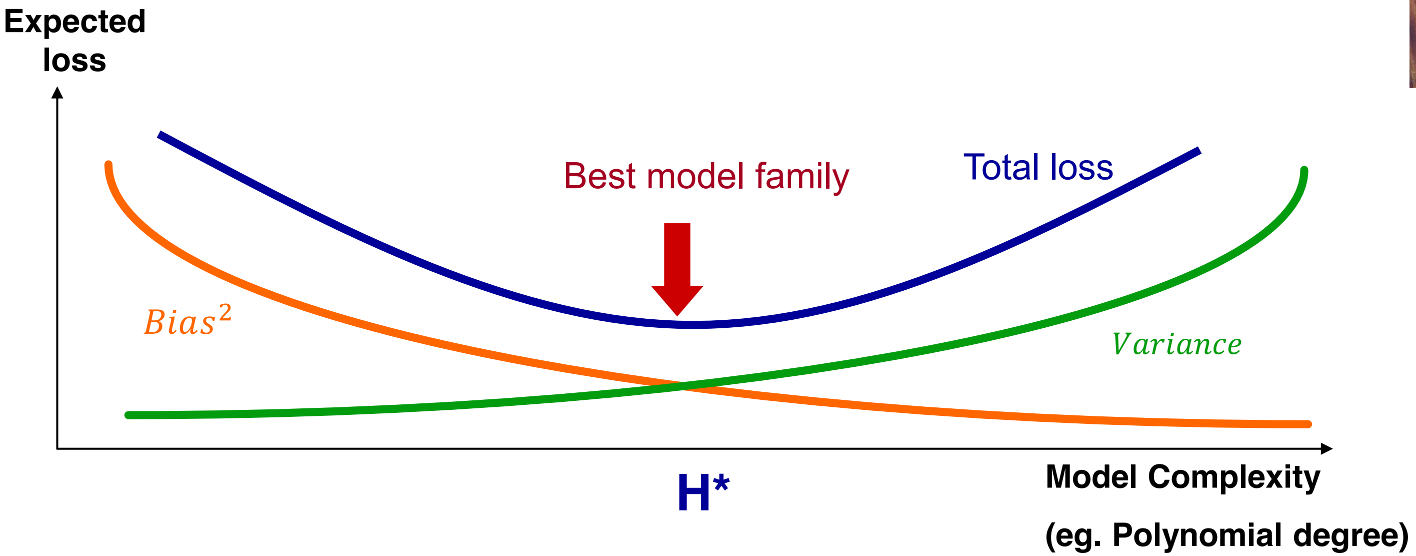
- How to decide the values for K
  - Commonly used $K = 5, 10$
  - Large K can be time-consuming
- Extreme case “leave-one-out”: use $K = m$ (number of samples)
  - Fit model using $m - 1$ samples and test on 1 sample (most expensive, but report reliable estimate of the model performance)
  - For linear regression, leave-one-out CV error have closed form expression called generalized cross-validation (GCV)

$$GCV\left(\hat{f}\right) = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{y^i - \hat{f}(x^i)}{1 - \frac{tr(S)}{m}}\right)^2, \; S = X^T(XX^T)^{-1}X$$

Theory of bias-variance tradeoff



Find the right model family s.t. expected loss becomes minimum



Consider bias-variance tradeoff in linear regression

- Given m training data points $D = \left\{(x^i, y^i)\right\}$, find θ that minimizes the mean square error

$$\hat{\theta} = argmin_\theta\, \hat{L}(\theta) := \frac{1}{m}\sum_{i=1}^{m}\left(y^i - \theta^\top x^i\right)^2$$

- But we really want to minimize the error for unseen ”test” data points $(\tilde{x}, \tilde{y})$ with respect to the entire distribution of data

$$\theta^* = argmin_\theta\, L(\theta) \coloneqq \mathbb{E}_{(\tilde{x},\tilde{y})}[(\tilde{y} - \theta^\top \tilde{x})^2]$$

General bias-variance trade-off

- Estimate your function from a finite training data set D

$$\hat{f} = argmin_f\ \hat{L}(f) := \frac{1}{m}\sum_{i=1}^{m}\left(y^i - f(x^i)\right)^2$$

- $\hat{f}$ is a random function depending on the distribution of training data D
- Expected loss of $\hat{f}$ (with respect to the training data D and test data distributions)

$$L(\hat{f}) := \mathbb{E}_D \mathbb{E}_{(x,y)}\left[\left(y - \hat{f}(x)\right)^2\right]$$

(x, y) denotes the test data

- Bias-variance decomposition

Expected loss = (bias)^2 + variance + noise

What is the optimal predictor?

The expected loss is

$$L(\hat{f}) := \mathbb{E}_D \mathbb{E}_{(x,y)} \left[\left(y - \hat{f}(x) \right)^2 \right]$$

$$= \mathbb{E}_D \left[\underbrace{\int \int \left(y - \hat{f}(x) \right)^2 p(x,y) dx dy}_{A(\hat{f})} \right]$$

First, let's figure out theoretical optimal predictor: the f that minimize $A(f)$.

$$\frac{\partial A(f)}{\partial f(x)} = \frac{\partial}{\partial f(x)} \int \int \left(y - f(x) \right)^2 p(x,y) dx dy$$

$$= 2 \int \left(y - f(x) \right) p(x,y) dy = 0$$

What is the optimal predictor?

(continue)

$$\frac{\partial A(f)}{\partial f(x)} = \frac{\partial}{\partial f(x)} \int \int \left(y - f(x)\right)^2 p(x,y)dxdy = 2\int \left(y - f(x)\right)p(x,y)dy = 0$$

$$\Leftrightarrow \int f(x)p(x,y)dy = \int yp(x,y)dy$$

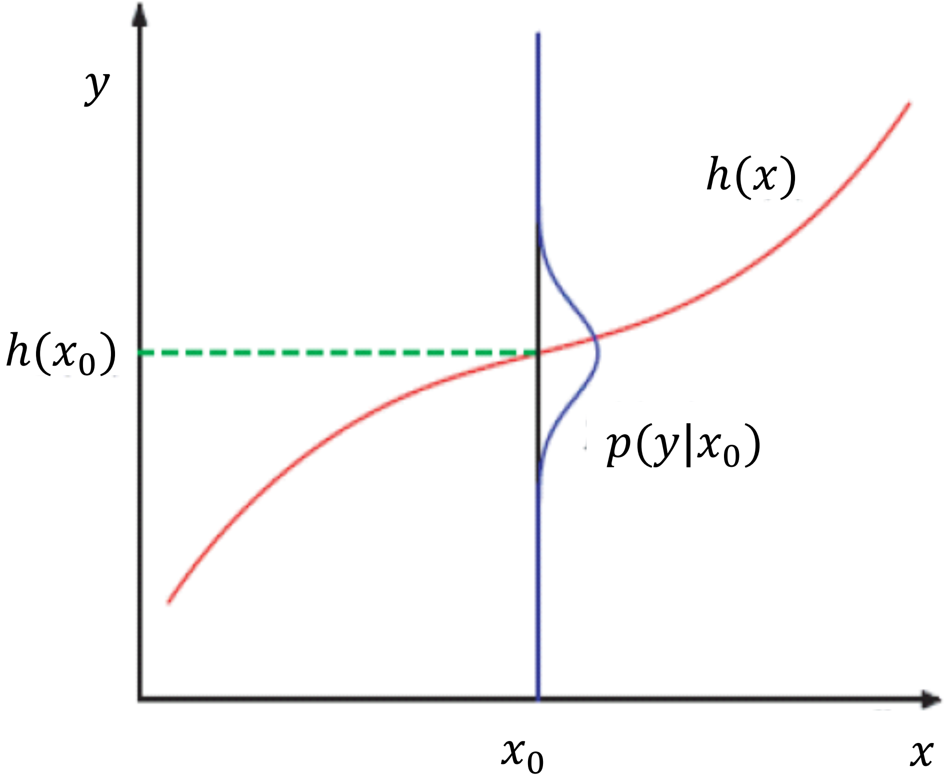
The left-hand-side: $\int f(x)p(x,y)dy = f(x)p(x)$

Hence, we have

$$f(x) = \ h(x) := \int \frac{yp(x,y)}{p(x)} \mathrm{dy} = \int yp(y|x)dy = \mathbb{E}[y|x]$$

The best predictor is the expected value

The best you can do is $h(x) = \mathbb{E}[y|x]$: the expected value of y given a particular x



Expected loss for optimal predictor

$h(x) = \mathbb{E}(y|x)$ is the **optimal** predictor, and $\hat{f}(x)$ our actual predictor, decompose the error a bit

$$\mathbb{E}_D\mathbb{E}_{(x,y)}\left[\left(y-\hat{f}(x)\right)^2\right] = \mathbb{E}_D\left[\int\int\left(y-h(x)+h(x)-\hat{f}(x)\right)^2 p(x,y)dxdy\right]$$

$$= \mathbb{E}_D\left[\int\int\left(\left(\hat{f}(x)-h(x)\right)^2 + 2\left(\hat{f}(x)-h(x)\right)(h(x)-y)\right.\right.$$

$$\left.\left. + (h(x)-y)^2\right)p(x,y)dxdy\right]$$

$$= \underbrace{\mathbb{E}_D\left[\int\left(\hat{f}(x)-h(x)\right)^2 p(x)dx\right]}_{\text{Will decompose further}} + \underbrace{\int\int(h(x)-y)^2 p(x,y)dxdy}_{\substack{\text{Noise term. can not do better than this.} \\ \text{a lower bound of the expected loss}}}$$

The cross term vanishes because:

$$\iint 2\left(\hat{f}(x) - h(x)\right)(h(x) - y)p(x,y)dxdy$$

$$= \int \left(\hat{f}(x) - h(x)\right)\left(h(x) - \underbrace{\int y\,p(y|x)dy}_{\mathbb{E}(y|x)}\right)p(x)dx$$

Since by definition $h(x) = \mathbb{E}(y|x)$, the equation is 0.

Bias-variance decomposition

Note that $\hat{f}(x)$ is a random function, generally different for different dataset D

$\mathbb{E}_D\left[\hat{f}(x)\right]$: expected value of $\hat{f}(x)$ with respected to random dataset

$$\mathbb{E}_D\left[\int\left(\hat{f}(x)-h(x)\right)^2 p(x)dx\right] = \mathbb{E}_D\mathbb{E}_x\left[\left(\hat{f}(x)-h(x)\right)^2\right]$$

$$= \mathbb{E}_x\mathbb{E}_D\left[\left(\hat{f}(x)-\mathbb{E}_D[\hat{f}(x)]+\mathbb{E}_D[\hat{f}(x)]-h(x)\right)^2\right]$$

$$= \mathbb{E}_x\mathbb{E}_D\left[\left(\hat{f}(x)-\mathbb{E}_D[\hat{f}(x)]\right)^2\right] + \mathbb{E}_x\mathbb{E}_D\left[\left(\mathbb{E}_D[\hat{f}(x)]-h(x)\right)^2\right]$$

$$-2\mathbb{E}_x\underbrace{\mathbb{E}_D\left[\left(\hat{f}(x)-\mathbb{E}_D[\hat{f}(x)]\right)\right.}_{0}\left.\left(\mathbb{E}_D[\hat{f}(x)]-h(x)\right)\right]$$

$$= \underbrace{\mathbb{E}_x\mathbb{E}_D\left[\left(\hat{f}(x)-\mathbb{E}_D[\hat{f}(x)]\right)^2\right]}_{\text{Variance}} + \underbrace{\mathbb{E}_x\left[\left(\mathbb{E}_D[\hat{f}(x)]-h(x)\right)^2\right]}_{\text{Bias}^2}$$

General bias-variance tradeoff

Putting things together

Expected loss = (bias)^2 + variance + noise

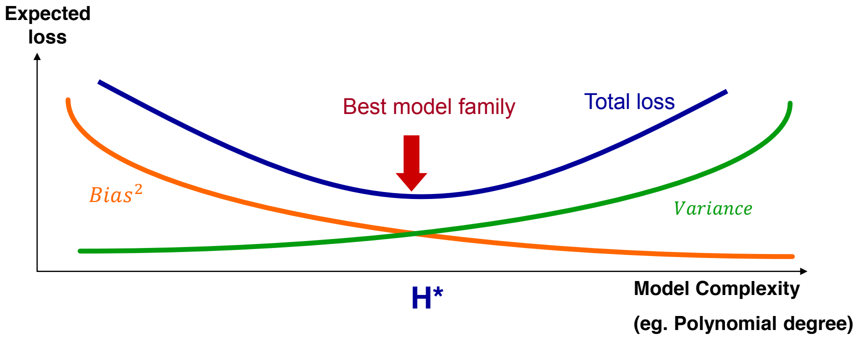
$$\mathbb{E}_D \mathbb{E}_{(x,y)}\left[\left(y - \hat{f}(x)\right)^2\right] = \mathbb{E}_x\left[\left(\mathbb{E}_D[\hat{f}(x)] - h(x)\right)^2\right]\ (bias^2)$$

$$+\ \mathbb{E}_x \mathbb{E}_D\left[\left(\hat{f}(x) - \mathbb{E}_D[\hat{f}(x)]\right)^2\right]\ (variance)$$

$$+\ \mathbb{E}_{(x,y)}[(h(x) - y)^2]\ (noise)$$

Key quantities

- $\hat{f}(x)$: actual predictor
- $\mathbb{E}_D[\hat{f}(x)]$: expected predictor
- $h(x) = \mathbb{E}(y|x)$: **optimal** predictor



SUMMARY