Overfitting and Model Generalization

Data Intelligence and Learning (DIAL) Lab

Prof. Jongwuk Lee

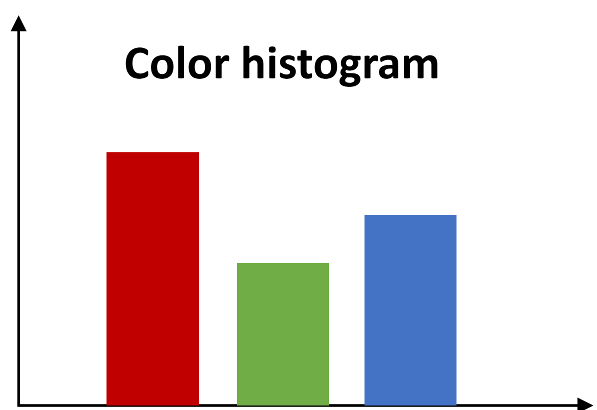
Math Formulation of Supervised Learning Models

Machine Learning 1-2-3

- Collecting data and extract features
- Building model: choose **hypothesis class** $\mathcal{H}$ and **loss function** $\mathcal{L}$
- Optimization: **minimize the empirical loss.**

- Q: How to extract the feature vector x?
- A: Still, it is difficult, e.g., image.

Feature vector: $\mathbf{x}^{(i)}$



Formulation in Supervised Learning

- **Training data**

$$\{(\mathbf{x}^{(i)}, y^{(i)}): 1 \leq i \leq n\}$$

- **Features**

$$\mathbf{x}^{(i)} \in \mathbb{R}^{d\times 1}$$

- **Target labels (ground-truth labels)**

$$y^{(i)} \in \{0, \dots, K-1\}$$

Classification

$$y^{(i)} \in \mathbb{R}$$

Regression

Formulation in Supervised Learning

- Given training data $\{(\mathbf{x}^{(\boldsymbol{i})}, y^{(i)}): \mathbf{1} \leq \boldsymbol{i} \leq \boldsymbol{n}\}$,
- Find $y = h(\mathbf{x})$ using training data,
- such that h is correct on test data.

Training Data vs. Test Data

- Given **training data** $\{(\mathbf{x}^{(i)}, y^{(i)}): 1 \leq i \leq n\}$,
- Find $y = h(\mathbf{x})$ using training data,
- such that h is correct on **test data**.

What is the connection between training data and test data?

Training Data vs. Test Data

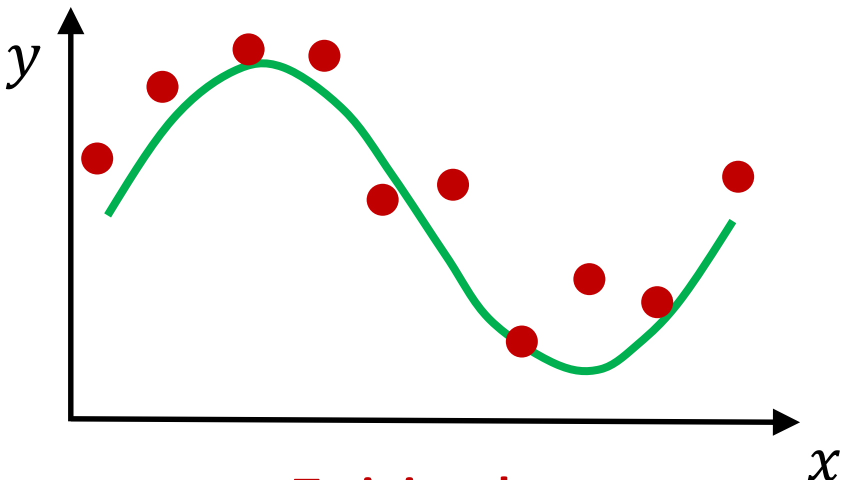
- **Given training data $\{(\mathbf{x}^{(i)}, y^{(i)}): 1 \le i \le n\}$ i.i.d. from distribution D,**
- **Find $y = h(\mathbf{x})$ using training data,**
- **such that h is correct on test data i.i.d. from distribution D.**

What is the connection between training data and test data?

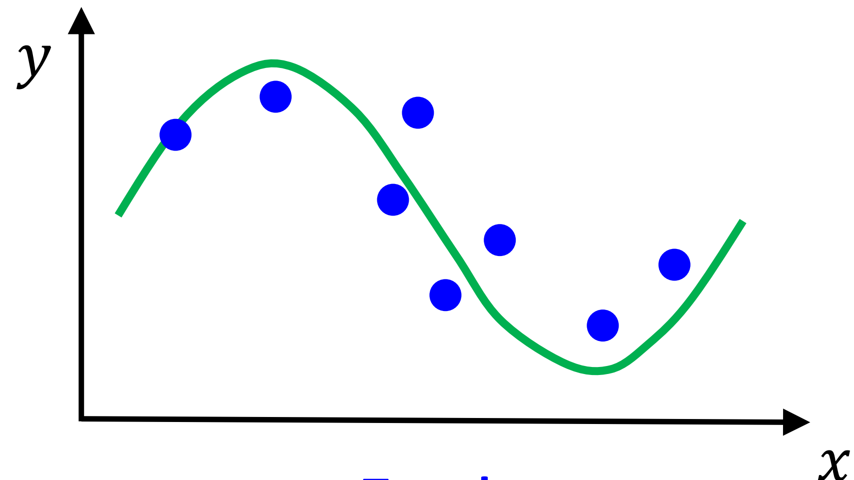
- **Assume that training and test data are sampled from the unknown but same distribution.**
  - i.i.d.: Independent and Identically Distributed

Training Data vs. Test Data

- Training data and test data are sampled from the same true data distribution.



Training data



Test data

- Assume that the unknown distribution is $\sin x$.

Hypothesis Function

- Given training data $\{(\mathbf{x}^{(i)}, y^{(i)}): 1 \leq i \leq n\}$ i.i.d. from distribution D,
- Find $y = h(\mathbf{x})$ using training data,
- such that h is correct on test data i.i.d. from distribution D.

What kind of functions are defined?

Hypothesis Function

➢ **Assume that there is some ideal function such that**

$$y = h^*(\mathbf{x})$$

➢ **Now, the goal is to learn h^* from the data.**

- A hypothesis is a **certain function** that we believe is similar to the true function, i.e., the **target function** that we want to model.
- Machine learning algorithms try to guess the **hypothesis function** that **approximates the unknown $h^*(\mathbf{x})$**.

$$h(\mathbf{x}) \approx h^*(\mathbf{x}) \text{ for all } \mathbf{x}^{(i)} \in \mathbb{R}^{d\times 1}$$

Hypothesis Function

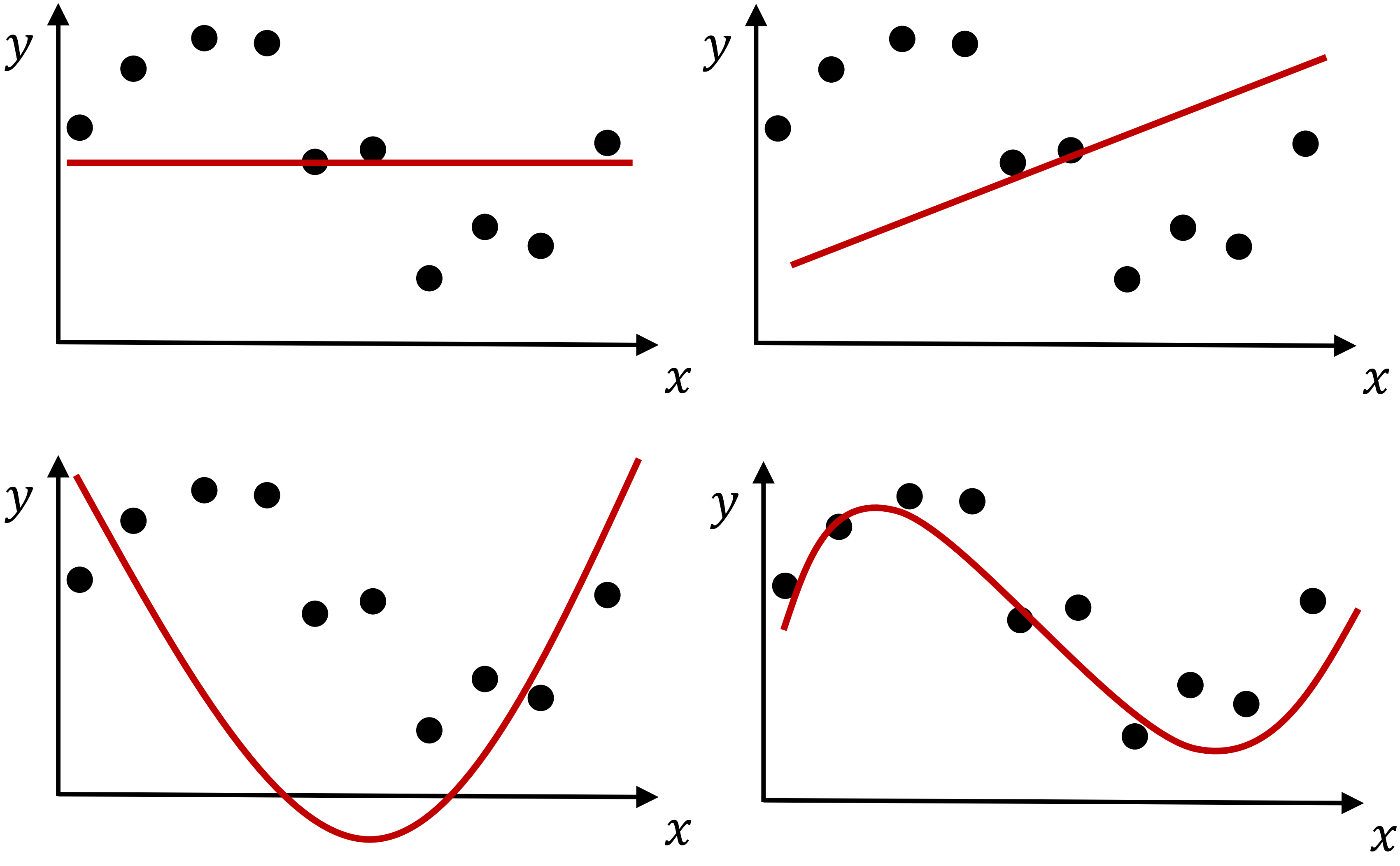
- Given training data $\{(\mathbf{x}^{(i)}, y^{(i)}): 1 \leq i \leq n\}$ **i.i.d. from distribution $\boldsymbol{D}$,**
- **Find $y = h(\mathbf{x})$ using training data,**
  - $h \in \mathcal{H}$ **: hypothesis class (or set)**
- **such that h is correct on test data i.i.d. from distribution $\boldsymbol{D}$.**

What kind of functions are defined?

⬇

$$h(\mathbf{x}) \approx h^*(\mathbf{x}) \text{ for all } \mathbf{x}^{(i)} \in \mathbb{R}^{d\times 1}$$

Possible Hypothesis Classes



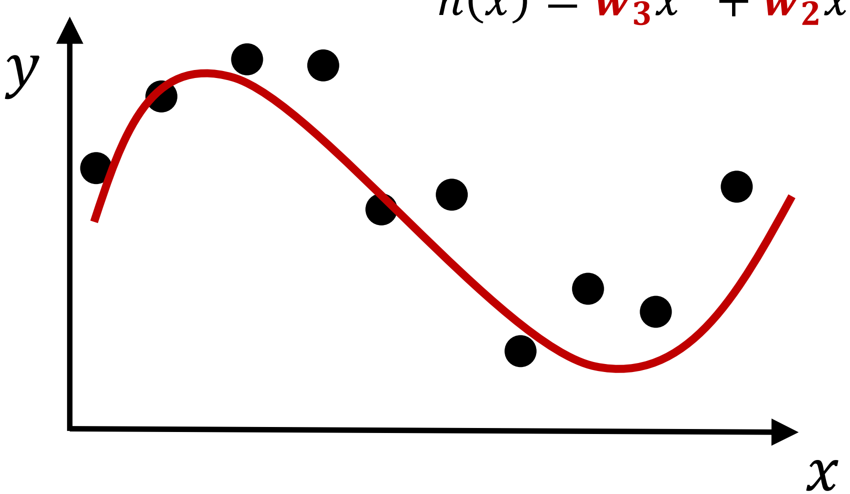
How to Search Parameters?

- **We hope finding h such that**

$$h(\mathbf{x}) \approx h^*(\mathbf{x}) \text{ for all } \mathbf{x}^{(i)} \in \mathbb{R}^{d\times 1}$$

- **There can be infinitely many h's to search!**
  - Typically, we assume a hypothesis set $\mathcal{H}$ and search among them.
  - Searching $\mathcal{H}$ is an important decision.

$$h(x) = \boldsymbol{w_3}x^3 + \boldsymbol{w_2}x^2 + \boldsymbol{w_1}x + \boldsymbol{w_0}$$



Loss Function

- Given training data $\{(\mathbf{x}^{(i)}, y^{(i)}) : 1 \leq i \leq n\}$ i.i.d. from distribution D,
- Find $y = h(\mathbf{x})$ using training data,
- such that h is correct on test data i.i.d. from distribution D.

What kind of performance is measured?

Loss Function

- Given training data $\{(\mathbf{x}^{(i)}, y^{(i)}): 1 \le i \le n\}$ i.i.d. from distribution D,
- Find $y = h(\mathbf{x})$ using training data,
- such that h minimizes the expected loss.

What kind of performance is measured?

⬇

$$\mathcal{L}(h(\mathbf{x}), y) = \mathcal{L}(h(\mathbf{x}), h^*(\mathbf{x}))$$

Loss Function

- **How to search $h \in \mathcal{H}$?**
  - Use a **loss function** to measure the difference between h^* and h.

$$\mathcal{L}(h(\mathbf{x}), y) = \mathcal{L}(h(\mathbf{x}), h^*(\mathbf{x}))$$

- **Examples**

$$\mathcal{L}(h(\mathbf{x}), y) = \left(y - h(\mathbf{x})\right)^2$$

$$\mathcal{L}(h(\mathbf{x}), y) = \begin{cases} 0, & if\ y = h(\mathbf{x}) \\ 1, & otherwise \end{cases}$$

Loss Function

- Given training data $\{(\mathbf{x}^{(i)}, y^{(i)}): 1 \leq i \leq n\}$ i.i.d. from distribution D,
- Find $y = h(\mathbf{x})$ that minimizes empirical loss,
- such that h minimizes the expected loss.

How to minimize the expected loss using training data?

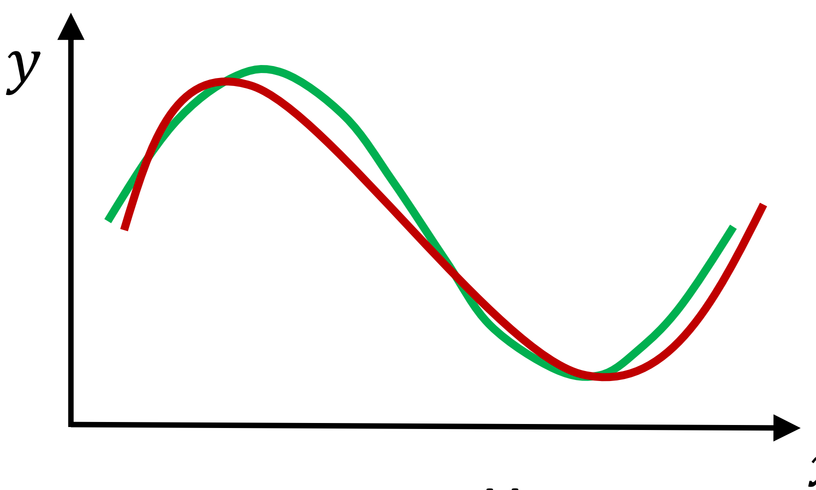
⬇

$$\underset{h \in \mathcal{H}}{\operatorname{argmin}} \mathbb{E}_{(\mathbf{x},y)\sim D}[\mathcal{L}(h(\mathbf{x}), y)] \approx \underset{h \in \mathcal{H}}{\operatorname{argmin}} \sum_{i=1}^{n} \mathcal{L}\left(h(\mathbf{x}^{(i)}), y^{(i)}\right)$$

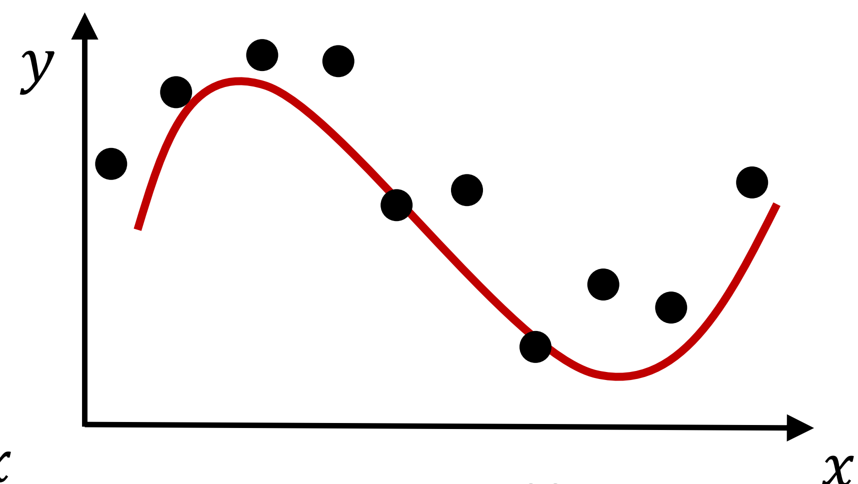
Expected loss **Empirical loss**

Expected Loss vs. Empirical Loss

- **The expected loss can utilize almost infinite training data sampled from the true distribution.**
  - However, it is impossible to know the true distribution.
- **Instead, the empirical loss can utilize a given training data.**
  - Although it is feasible, it may incur a **potential** problem.



Expected loss



Empirical loss

Empirical Loss Function

- **The exact expectation is impossible to compute.**
  - We replace with an **empirical loss**.

Expected loss **Empirical loss**

$$\underset{h\in\mathcal{H}}{\operatorname{argmin}}\, \mathbb{E}_{(\mathbf{x},y)\sim D}[\mathcal{L}(h(\mathbf{x}),y)] \approx \underset{h\in\mathcal{H}}{\operatorname{argmin}} \sum_{i=1}^{n} \mathcal{L}\left(h(\mathbf{x}^{(i)}),y^{(i)}\right)$$

How to minimize?
(optimization problem)

$$\underset{h\in\mathcal{H}}{\operatorname{argmin}} \sum_{i=1}^{n} \mathcal{L}\left(h(\mathbf{x}^{(i)}),y^{(i)}\right)$$

How to choose a hypothesis function?

How to choose $\mathcal{L}$? (regression/classification)

Inference

- Given a new sample $\mathbf{x}_{test}$, the prediction becomes

$$\hat{y} = h(\mathbf{x}_{test})$$

- We want to minimize

$$\mathbb{E}_{(\mathbf{x},\boldsymbol{y})\sim \boldsymbol{D}}[\mathcal{L}(h(\mathbf{x}_{test}), y)] = \mathbb{E}_{(\mathbf{x},\boldsymbol{y})\sim \boldsymbol{D}}[\mathcal{L}(\hat{y}, y)]$$

- The **overfitting problem** may happen.

Overfitting Problem

Recap: Generalized Linear Regression

➢ Linear regression

- Find $\mathbf{w}$ so that $f(\mathbf{x})$ best fits a given data

$$f(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_d x_d = w_0 + \sum_{j=1}^{d} w_j x_j$$

➢ Generalized linear regression

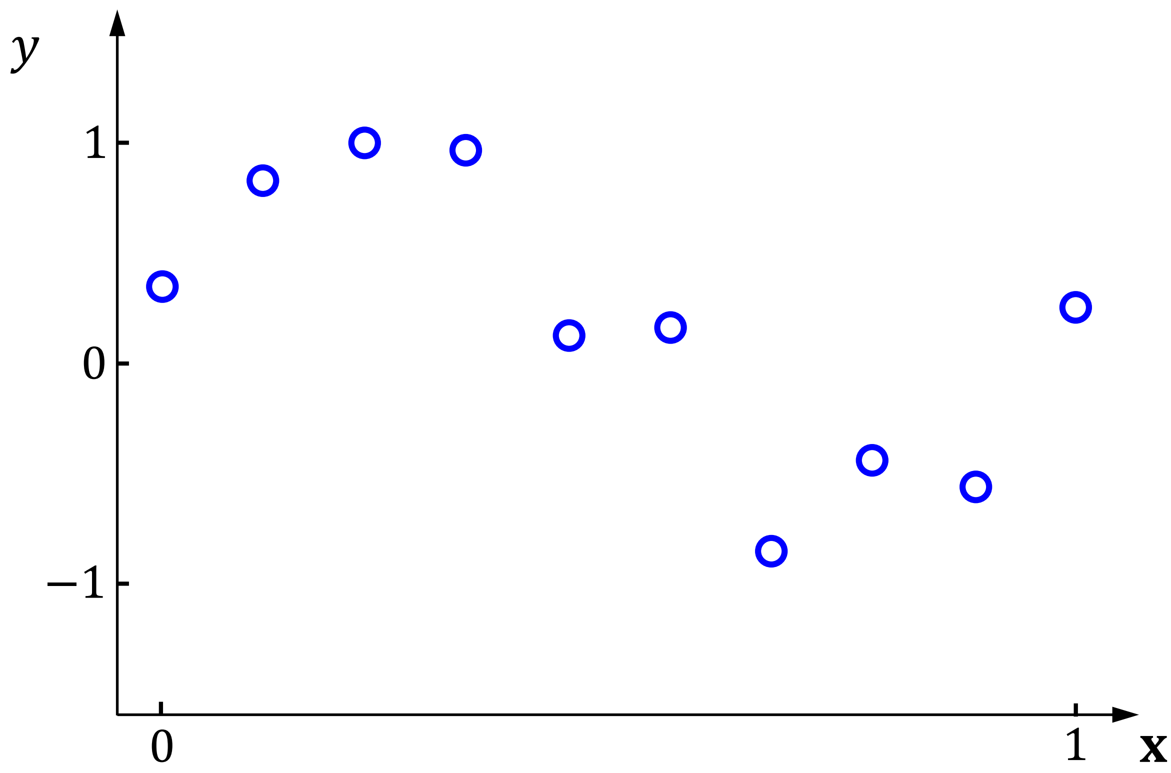
- Instead of using variables, use a **basis function $\phi_i(\mathbf{x})$** of $\mathbf{x}$.

$$f(\mathbf{x}) = w_0 + w_1 \phi_1(\mathbf{x}) + w_2 \phi_2(\mathbf{x}) + \cdots + w_d \phi_d(\mathbf{x}) = w_0 + \sum_{j=1}^{d} w_j \phi_j(\mathbf{x})$$

Polynomial Curve Fitting

- Which order polynomial does best fit for the data?

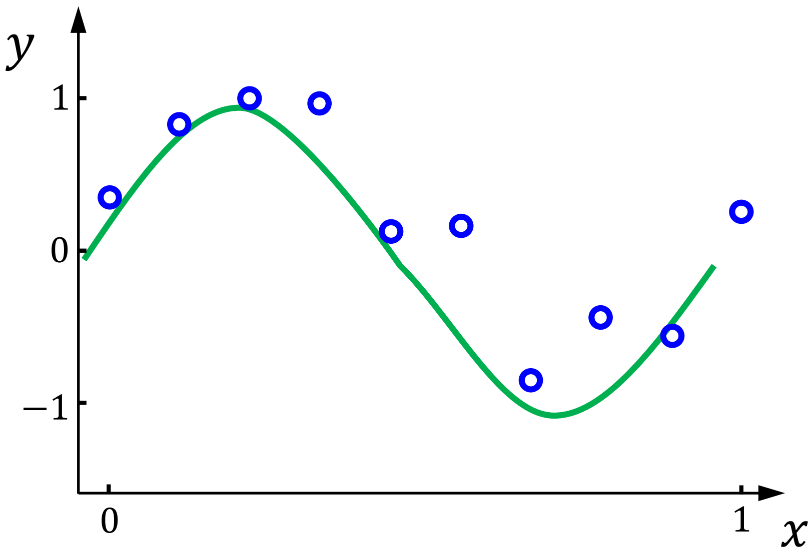
$$f(\mathbf{x}) = w_0 + w_1 x + w_2 x^2 + \cdots w_M x^M = \sum_{i=0}^{M} w_i^T x^i$$



Polynomial Curve Fitting

- **Considering a training data consisting of 1-dimensional observation with a corresponding label y**
  - The polynomial function is a **non-linear function** of x, but it is a **linear function** of the coefficients $\mathbf{w}$.

$$f(\mathbf{x}) = w_0 + w_1 x + w_2 x^2 + \cdots w_M x^M = \sum_{i=0}^{M} w_i^T x^i$$



What M should we choose?
Model selection

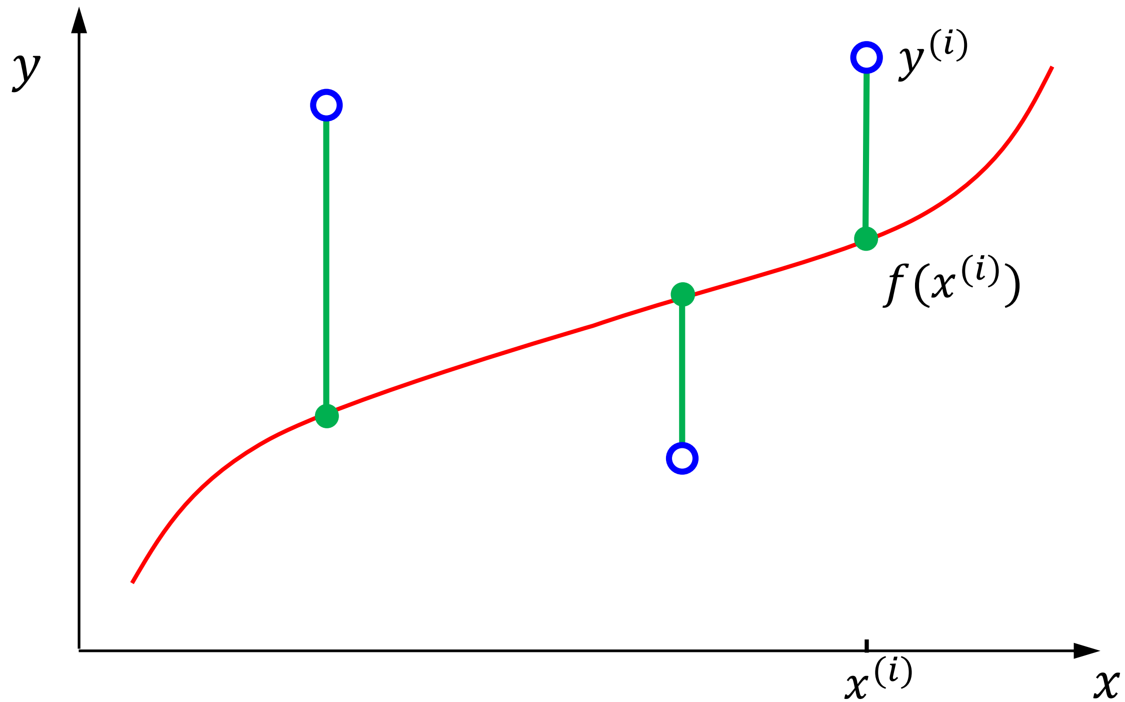
Given M, what w's should we choose?
Parameter selection

Ground truth: $sinx$

Error Function of Polynomial Curve

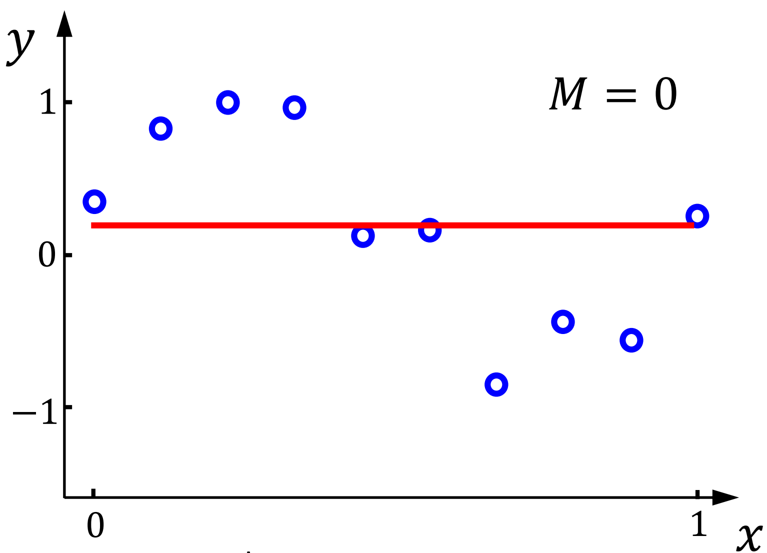
- We want to minimize the sum-of-squared error function.

$$E(\mathbf{w}) = \frac{1}{2n}\sum_{i=1}^{n}\left(y^{(i)} - f\left(x^{(i)}\right)\right)^2$$

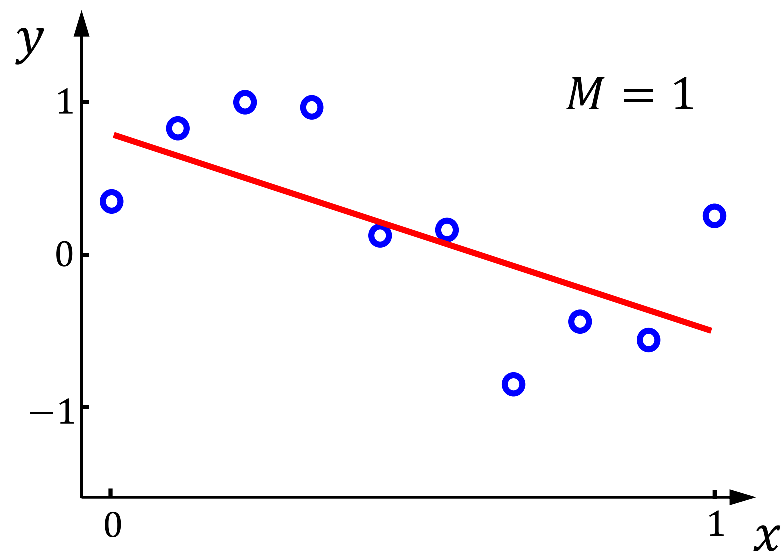


Example: Model Comparison

➢ **Which model is better?**



0th order polynomial regression

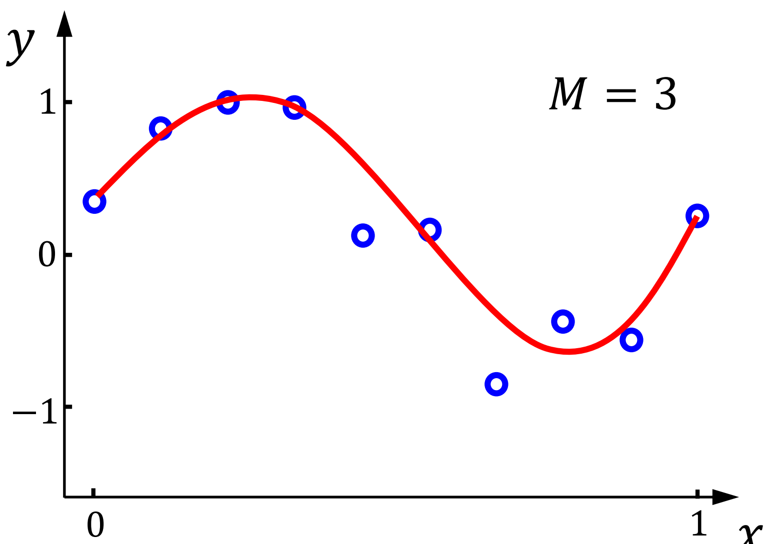


1st order polynomial regression

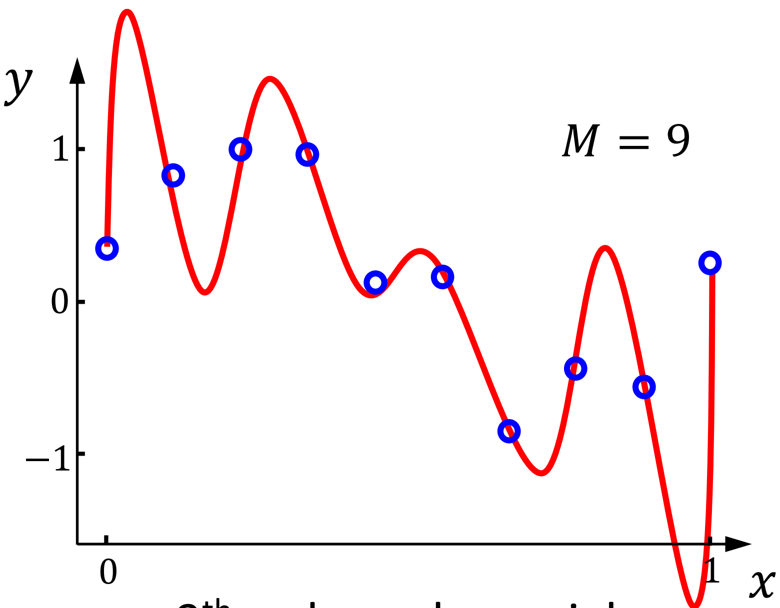
➢ **The right model is better because it has less error.**

Example: Model Comparison

- **Which model is better?**



3rd order polynomial regression



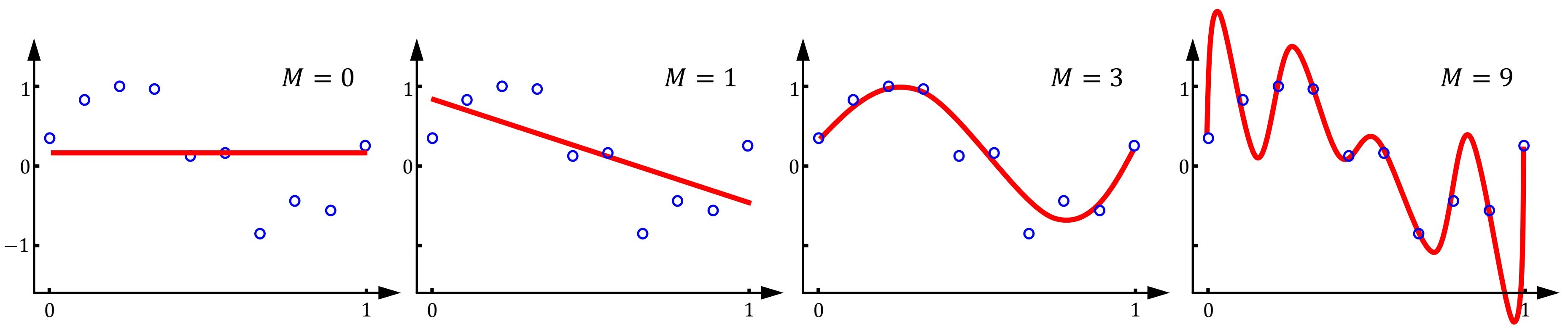
9th order polynomial regression

- **The right model is better. Do you agree?**

Model Complexity vs. Accuracy

- **As the order (M) increases,**
  - The complexity of model increases.

- **As the complexity of model increases,**
  - The model can more exactly learn the given data.
  - **The prediction accuracy does not necessarily increase.**



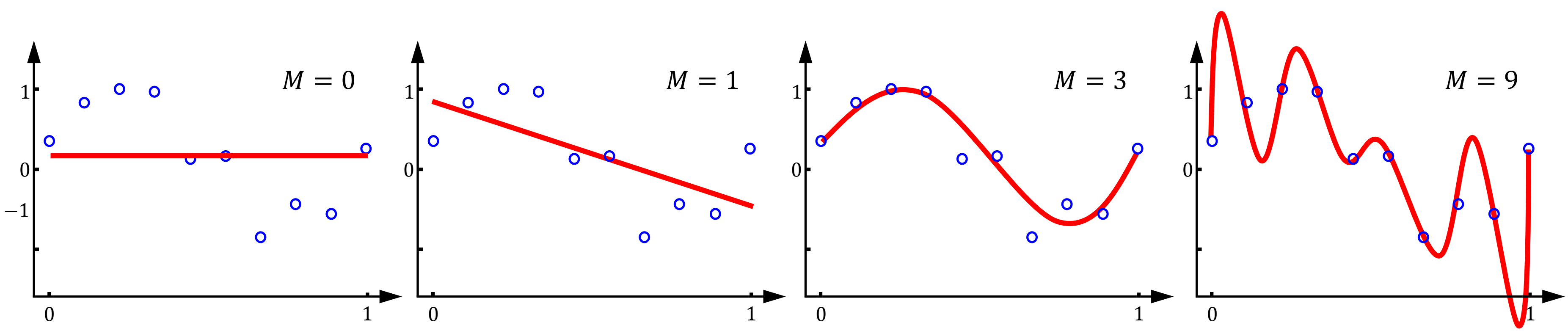
Overfitting vs. Generalization

➢ **What is the purpose of machine learning?**

Learning the given data as exactly as possible

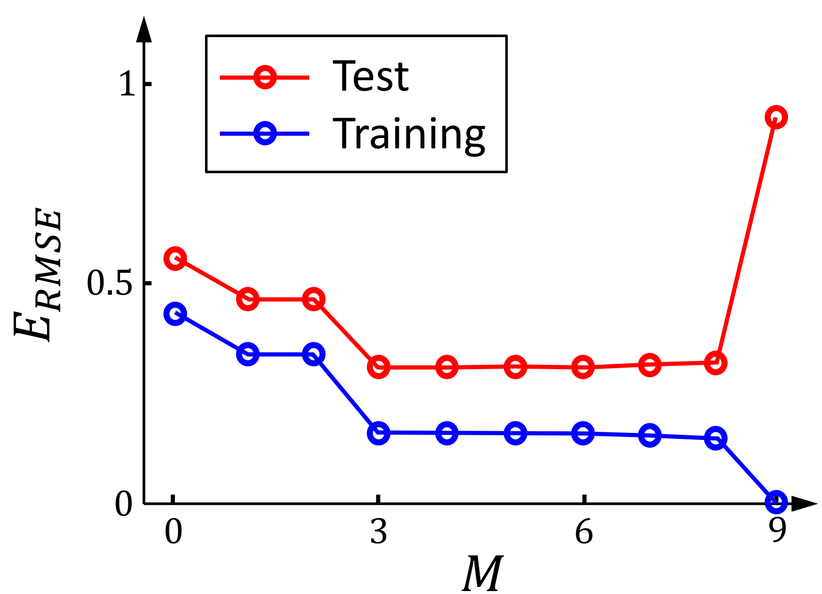
vs.

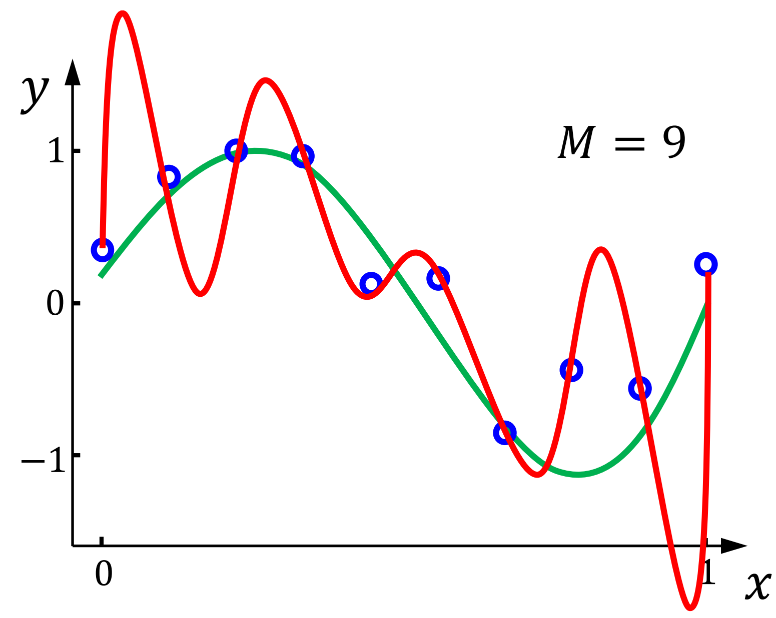
Predict the unknown data as exactly as possible based on the given data



Overfitting Problem

- **For $M = 9$, the training error is zero.**
  - The polynomial contains 10 degrees of freedom corresponding to 10 parameters, so we can be fixed exactly to the 10 data points.
- **However, the test error has become very large. Why?**

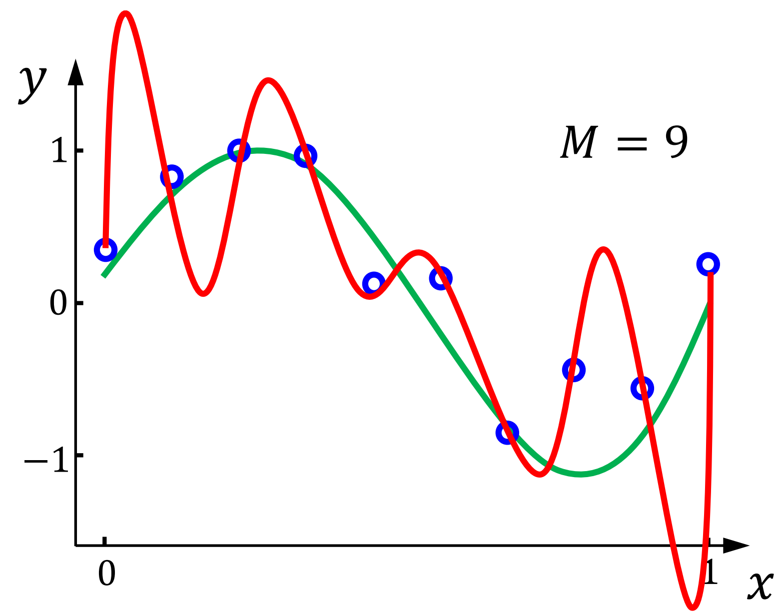




Overfitting Problem

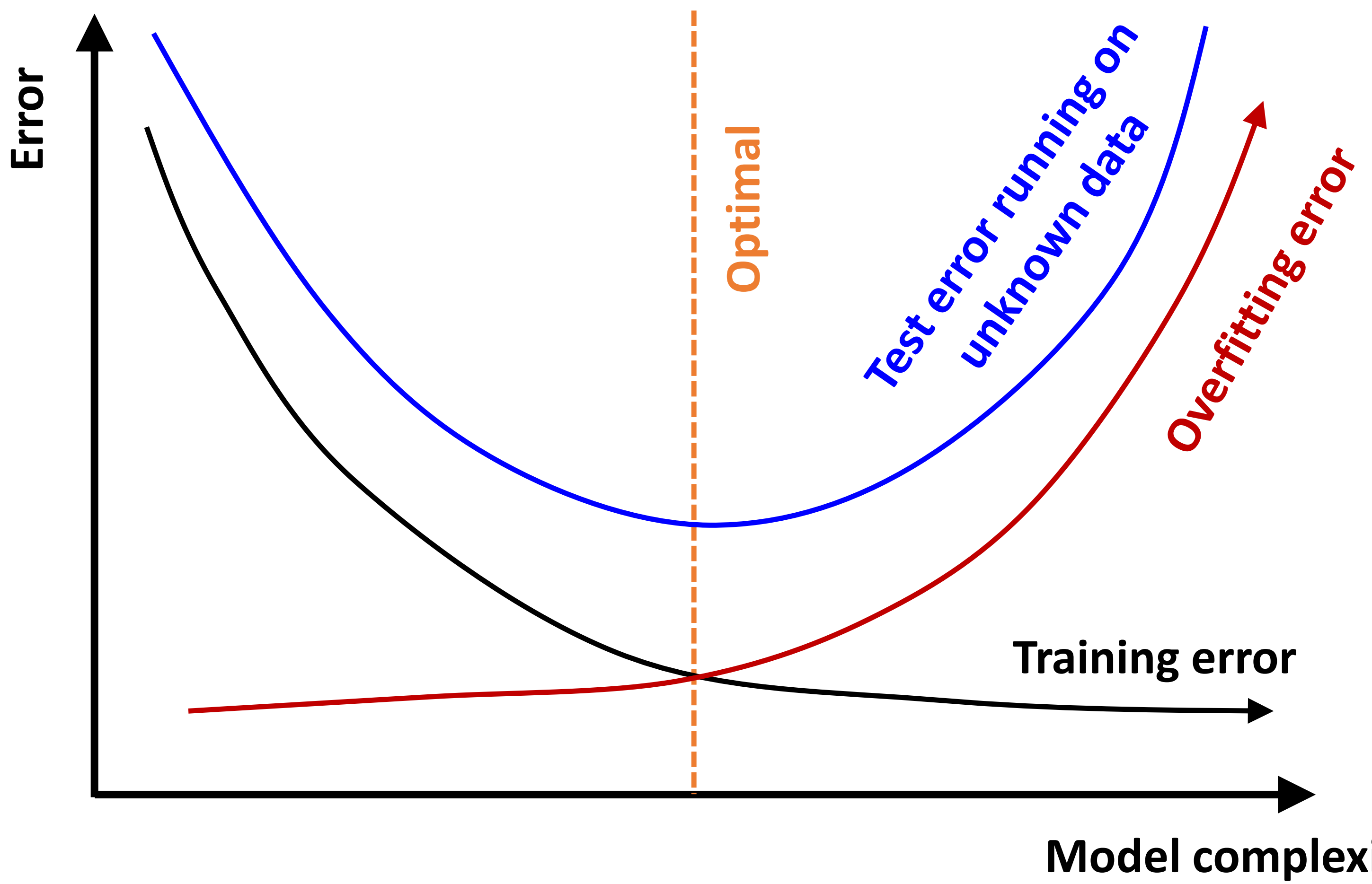
- **As M increases, the magnitude of coefficients gets larger.**
  - For $M = 9$, the coefficients have become finely tuned to the data.
  - Between data points, the function exhibits **large oscillations**.

	$M=0$	$M=1$	$M=3$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43



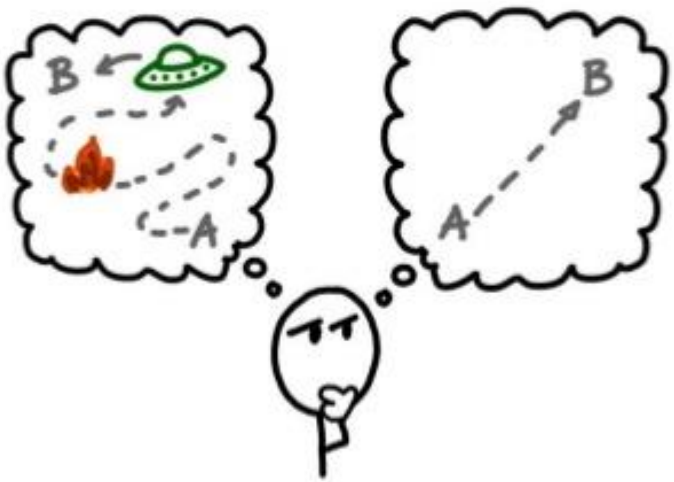
What is Generalization?

- Expect the model to generalize if it explains the data well given the complexity of the model.



Occam's Razor Principle

- **How to control model complexity to optimize generalization?**
- **Among competing hypotheses, select the one that makes the fewest assumptions and is thus most open to being tested.**



When faced with two equally good hypotheses, always choose the simpler.

How to Achieve Generalization?

- **The goal is to achieve good generalization by making accurate predictions for test data.**
  - Choosing the values of parameters that minimize the loss function on the training data may not be the best option.

- **We would like to model the true regularities in the data and ignore the noise in the data.**
  - Adding **more information** to overcome the overfitting problem

- **Examples**
  - **Data augmentation**
  - Weight decay: L^p **regularization**
  - **Early stopping** with a validation set

Increasing the Size of Data

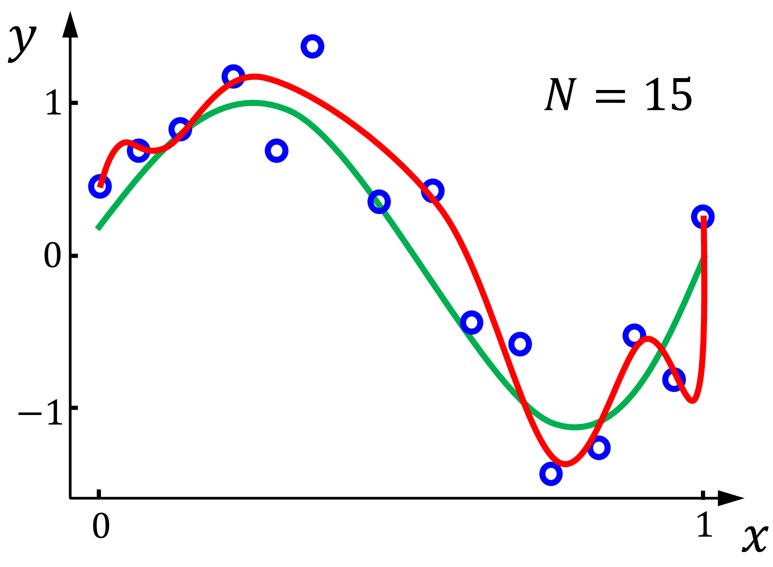
- For a given model complexity, the overfitting problem becomes less severe as the data size increases.

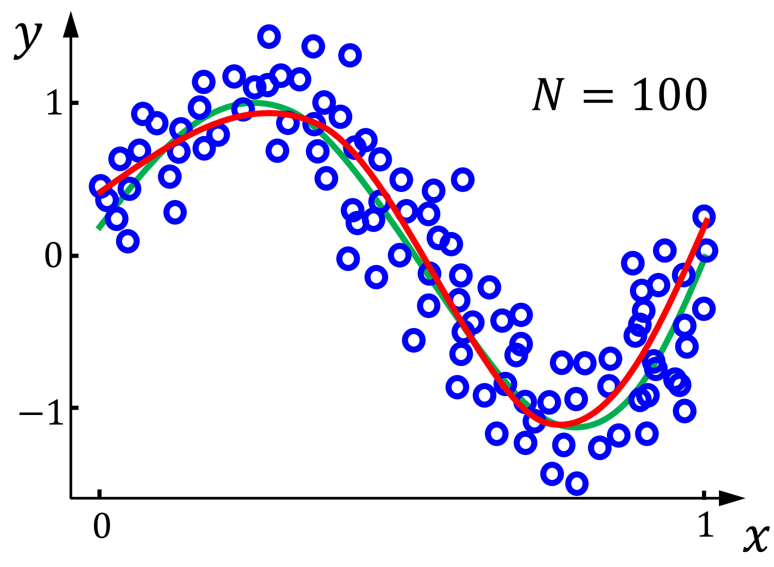
- The number of parameters is not necessarily the most appropriate measure of the model complexity.

- Collecting more data is an easy task?

Example: Fitting Polynomial Curve

➢ They are both 9^{th} order polynomials with **different** data size.





Penalizing the Model Complexity

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}\left(f\left(x^{(i)}\right), y^{(i)}\right) + \lambda \Omega(\theta)$$

Fit the data

Penalize complex models

Regularization parameter

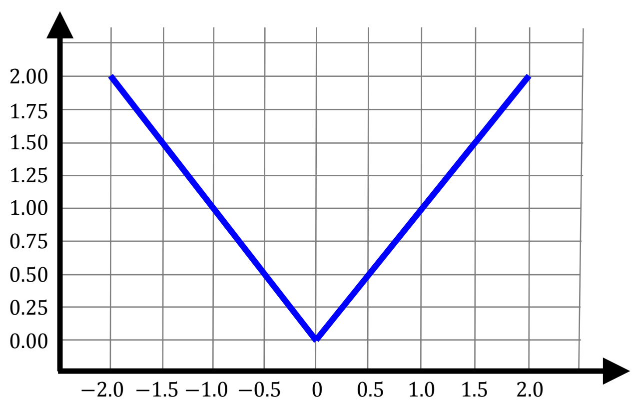
- How should we define $\Omega(\theta)$?
- How should we define λ?

Common Regularization Functions

➢ Lasso regression (L1-Reg)

$$\Omega_{\text{Lasso}}(\theta) = \sum_{i=1}^{d} |\theta_i|$$

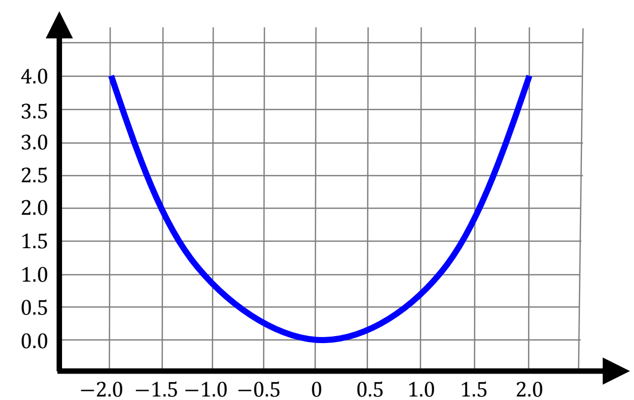
- Encourage sparsity by setting weight = 0.
  - Used to select the most informative features.
- Does not have an analytic solution → **numerical methods**.



➢ Ridge regression (L2-Reg)

$$\Omega_{\text{Ridge}}(\theta) = \sum_{i=1}^{d} \theta_i^2$$

- Does not encourage sparsity → **small but non-zero weights.**
- Distributes weight across related features (robust).
- Analytic solution (easy to compute)



Example: Fitting a Polynomial Curve

- **One technique for controlling overfitting problem is regularization, which amounts to adding a penalty term to the error function.**
  - **Shrinking to zero**: penalize coefficients based on their size.
  - For a penalty function which is the sum of the squares of the parameters, this is known as "**weight decay**", or "**ridge regression**".

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2n}\sum_{i=1}^{n}\left(y^{(i)} - f\left(\mathbf{x}^{(i)}\right)\right)^2$$

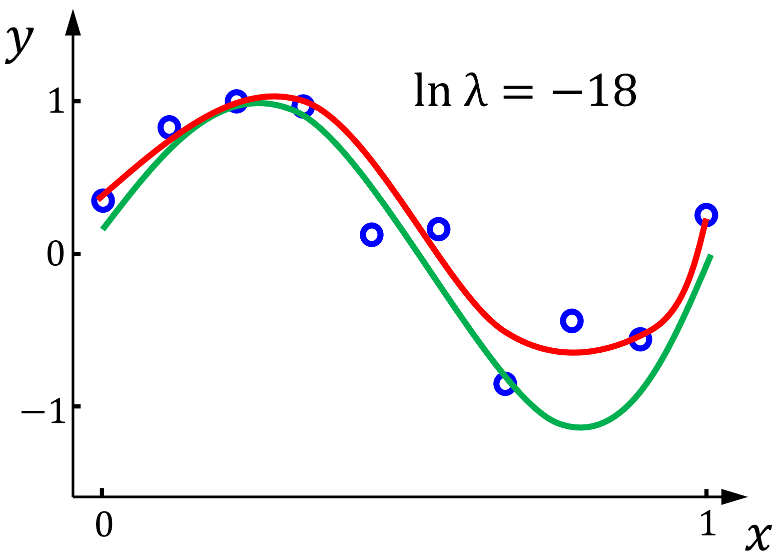
$$\Downarrow$$

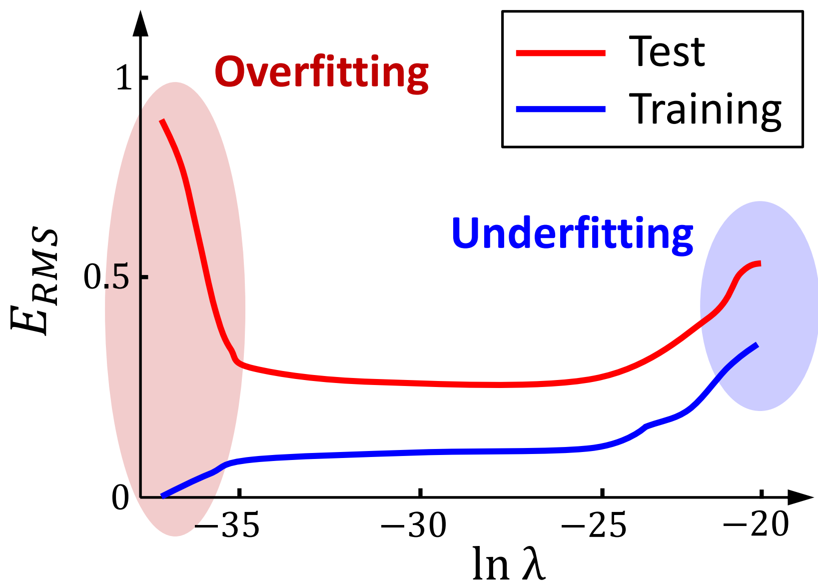
$$\mathcal{L}(\mathbf{w}) = \frac{1}{2n}\sum_{i=1}^{n}\left(y^{(i)} - f\left(\mathbf{x}^{(i)}\right)\right)^2 + \lambda\|\mathbf{w}\|^2$$

Example: Fitting a Polynomial Curve

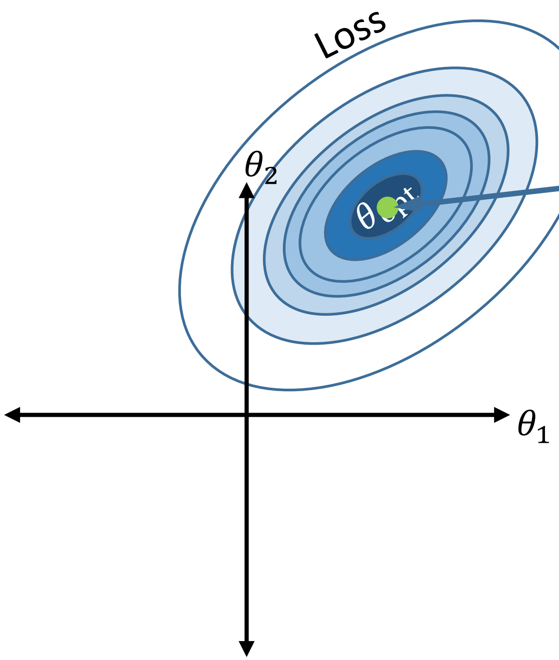
- Training and test errors vs. regularization for the $M = 9$ polynomial

- Small λ vs. Large λ





Regularization and Norm Balls



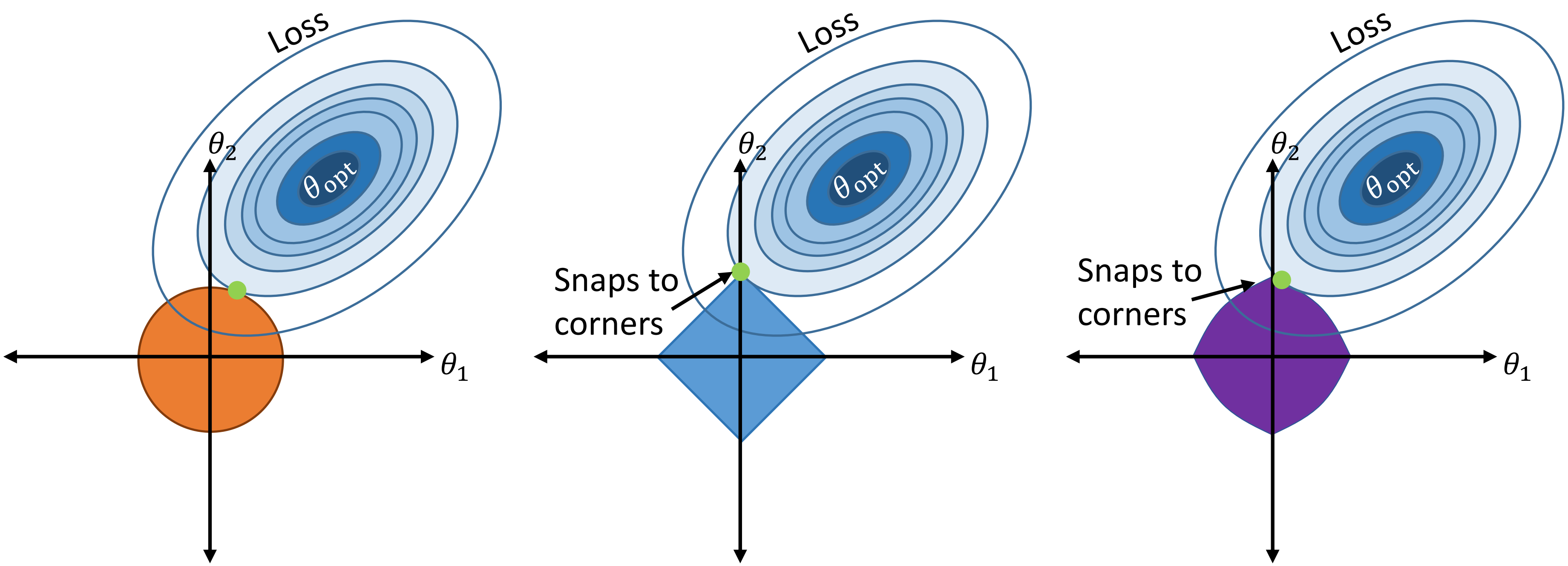
This parameter minimizes the error function.

Without **regularization**, we aim to find **an optimal parameter**.

In terms of **generalization**, the optimal value can **incur large errors**.

Regularization and Norm Balls

- Regularization makes a **constraint** for finding the parameter.
- The error increases but the model is **less complicated, which can be better for generalization error.**



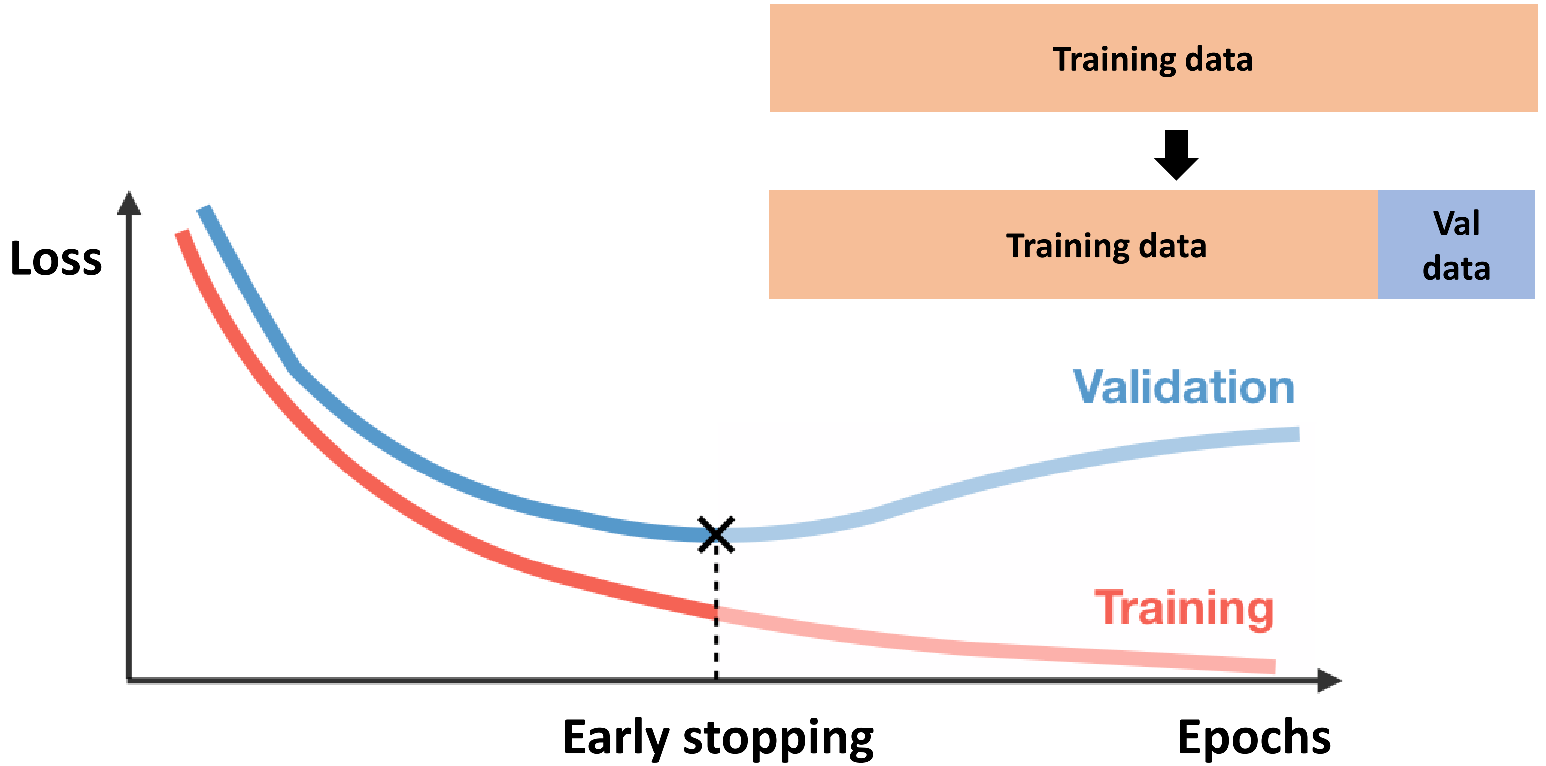
L2 Norm (Ridge)

L1 Norm (LASSO)

L1 + L2 Norm (Elastic Net)

Early Stopping with a Validation Set

- **It is difficult to stop learning before converging too much.**
- **Usually, it is determined by a validation set.**
  - The validation set is randomly chosen from the training set.
  - The validation set is **NOT used** for model training.



Bias-Variance Trade-off

True vs. Empirical Risk

- **True risk: Target performance measure**
  - Minimize the performance on all samples from true distribution
  - **Classification**: the number of misclassified samples
  - **Regression**: mean squared error

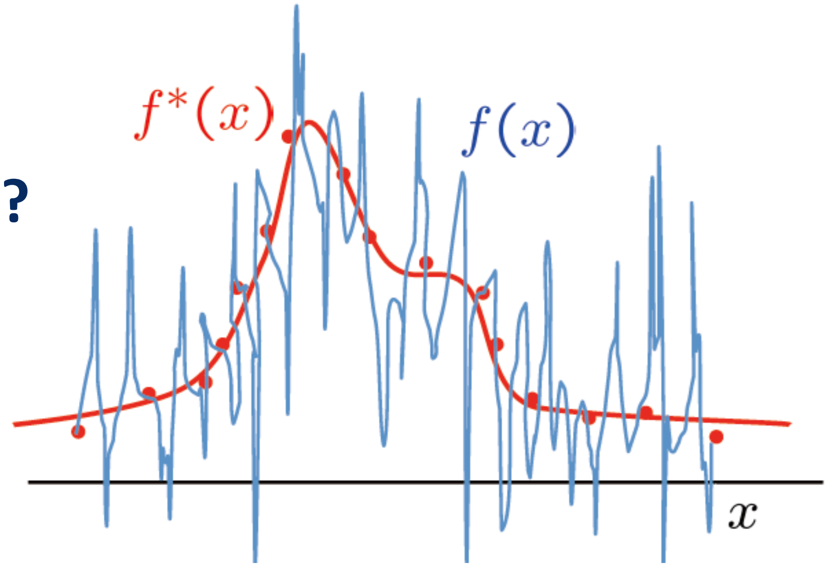
$$\mathbb{E}\left[\left(\mathcal{L}(h(\mathbf{x}), y)\right)\right] = \int \left(\mathcal{L}(h(\mathbf{x}), y)\right) dP(x, y)$$

- **Empirical risk: performance on training data**
  - **Classification**: a proportion of misclassified samples
  - **Regression**: average squared error

$$\frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(h(\mathbf{x}), y)$$

Overfitting Problem

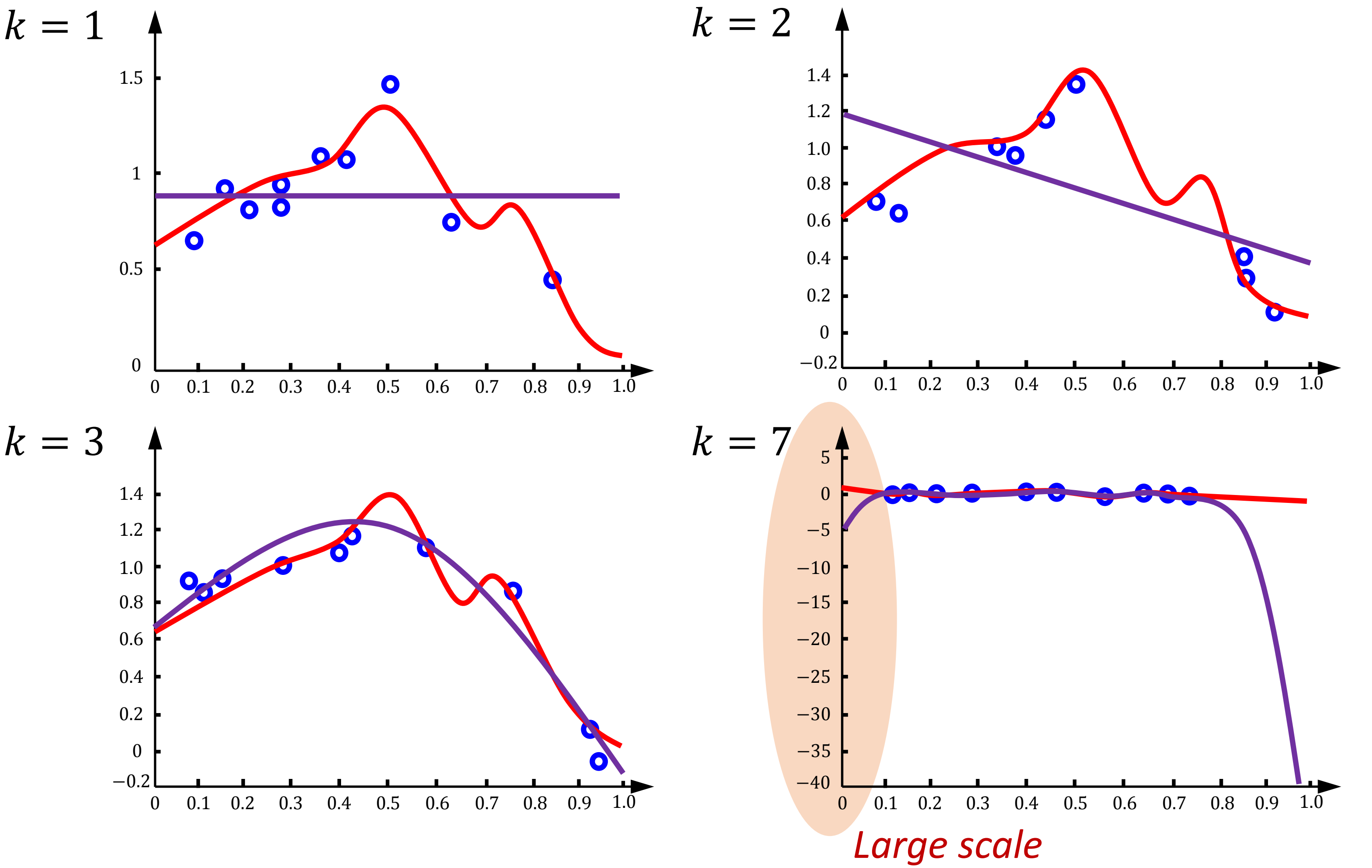
- What is the **empirical risk**? (performance on training data)
- zero!
- What is the **true risk** for $f^*(x)$?
- > zero



- $f(x)$ will predict very poorly on a new random test sample.
- **Incur a large generalization error!**

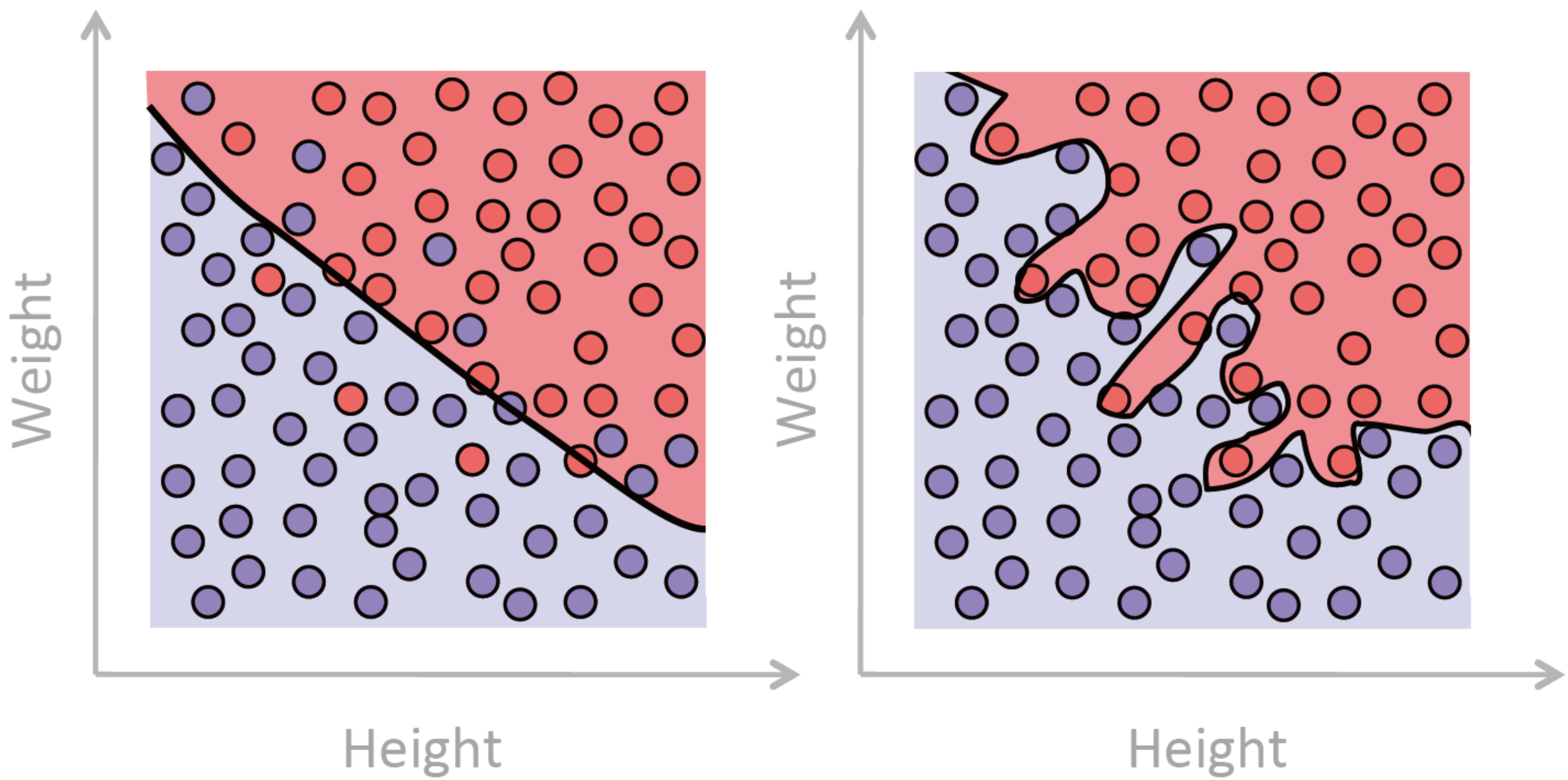
Example: Overfitting in Regression

- For very complicated predictors, we can overfit training data.



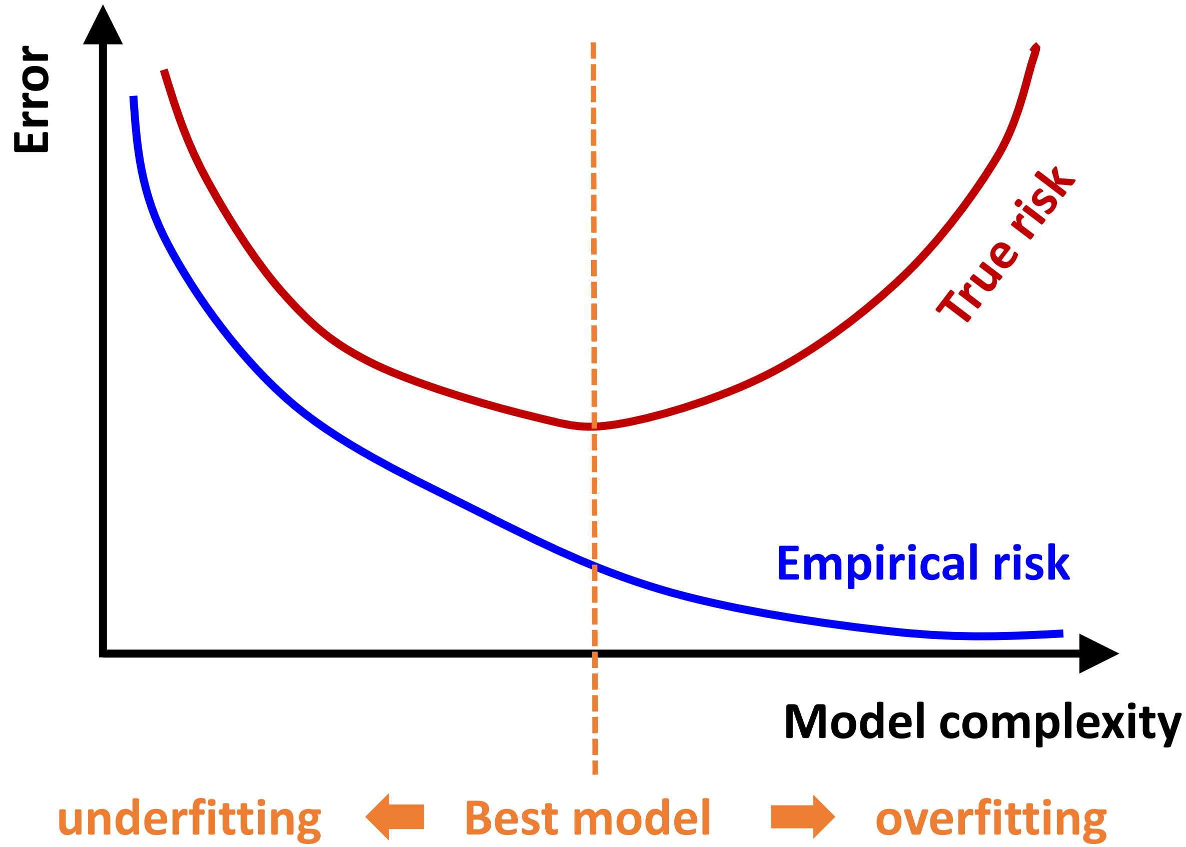
Example: Overfitting in Classification

- For very complicated predictors, we can overfit training data.



Effect of Model Complexity

➢ For fixed # of training data, empirical risk is no longer a good indicator of true risk.



Fundamental Challenges

- **Model generalization**
  - After learning from the training data, we can effectively predict the unobserved data without the **overfitting problem**.

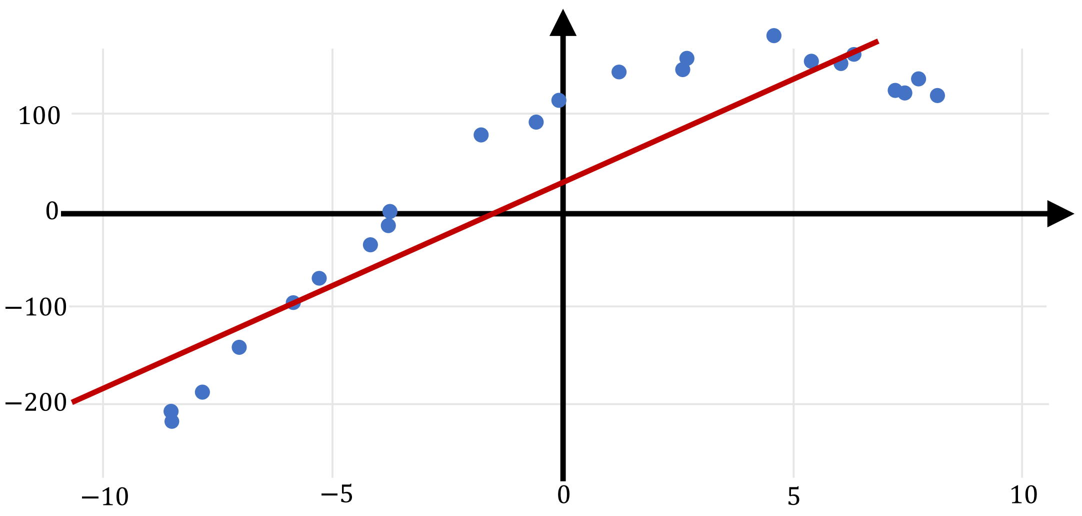
- **Bias**
  - The **expected deviation** between predicted value and the true value

- **Variance**
  - **Observation variance**: the **variability** of the random noise in the process we are trying to the model.
  - **Estimated model variance**: the **variability** in the predicted value across different training datasets.

Bias

- **The expected deviation between predicted value and the true value**
  - Depends on **the choice of f** or **learning procedure**.

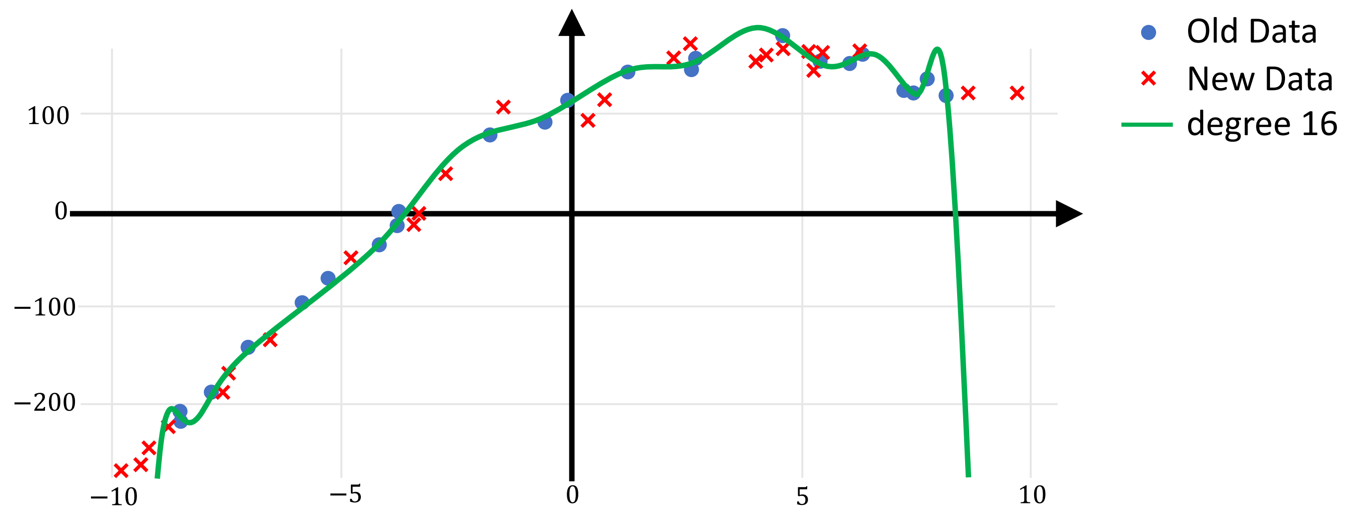
- **Underfitting**



Estimated Model Variance

- **Variability in the predicted value across different training datasets**
  - Sensitivity to the variation in the training data
  - Poor generalization

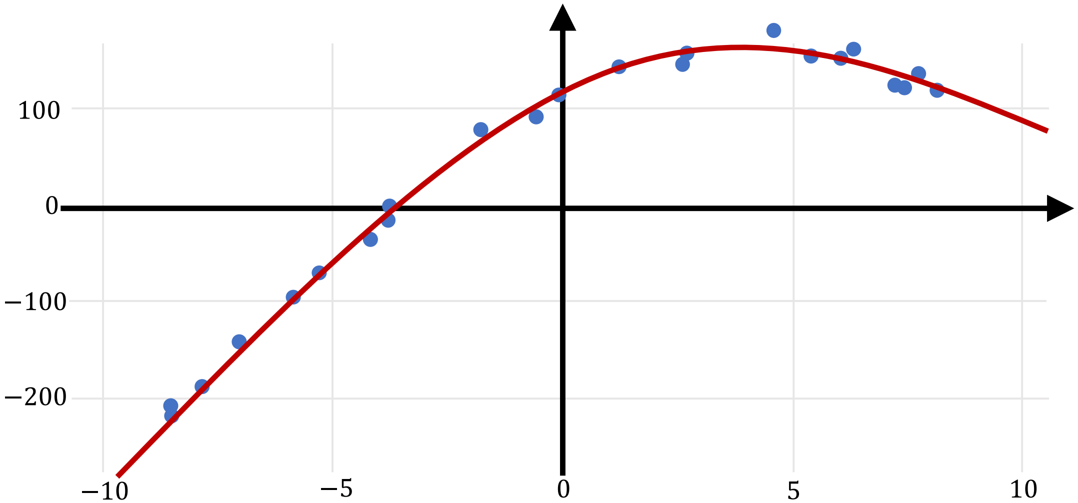
- **Overfitting**



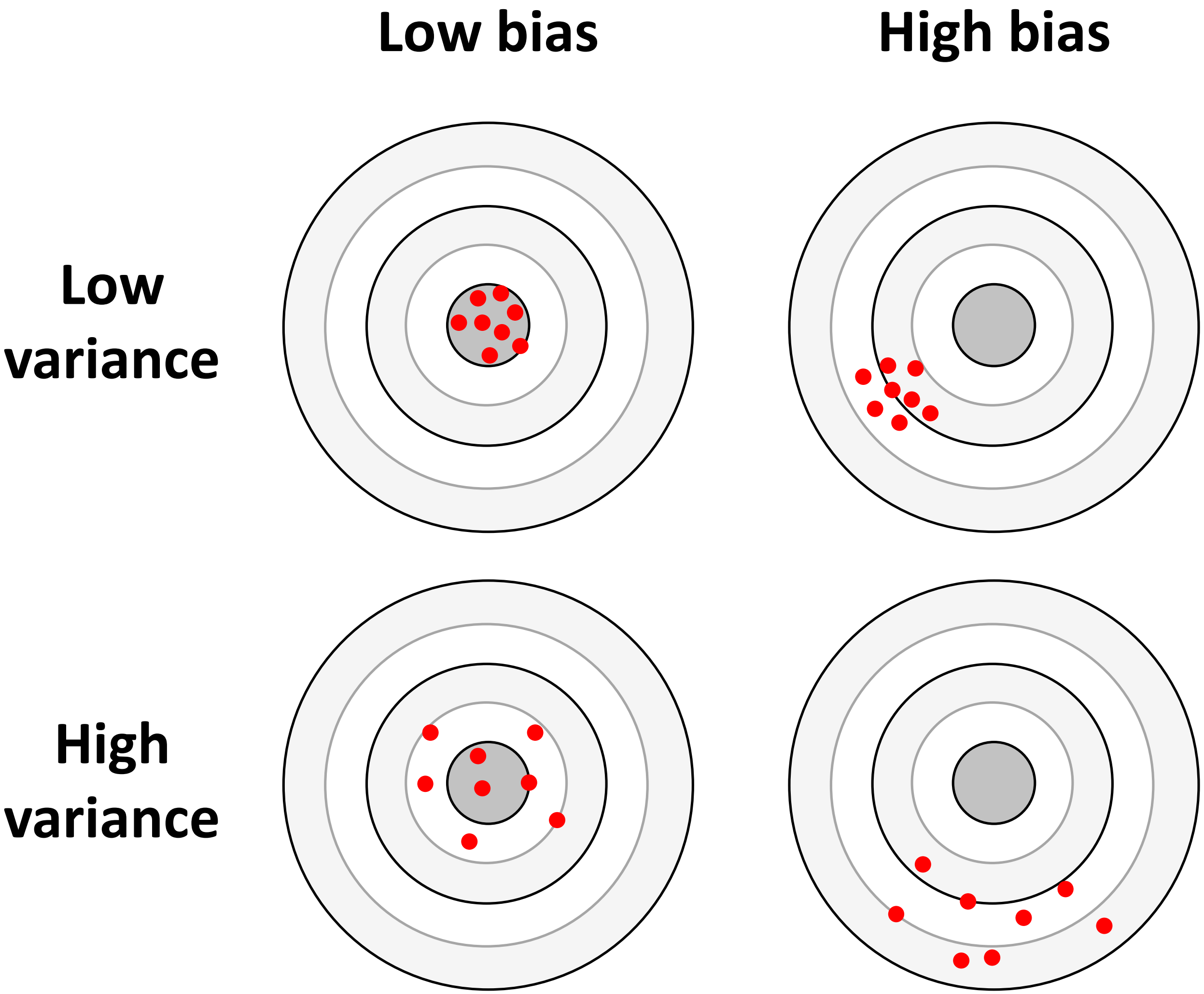
Observation Variance

- **The variability of the random noise in the process we are trying to the model**
  - Measurement variability
  - Stochasticity
  - Missing information

- **Usually, it is beyond our control.**



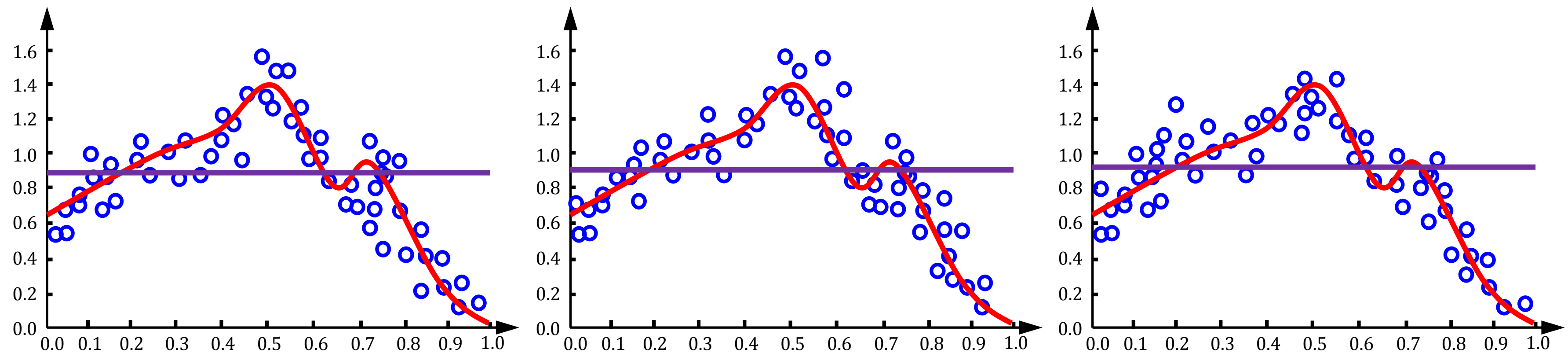
Visualization: Bias and Variance



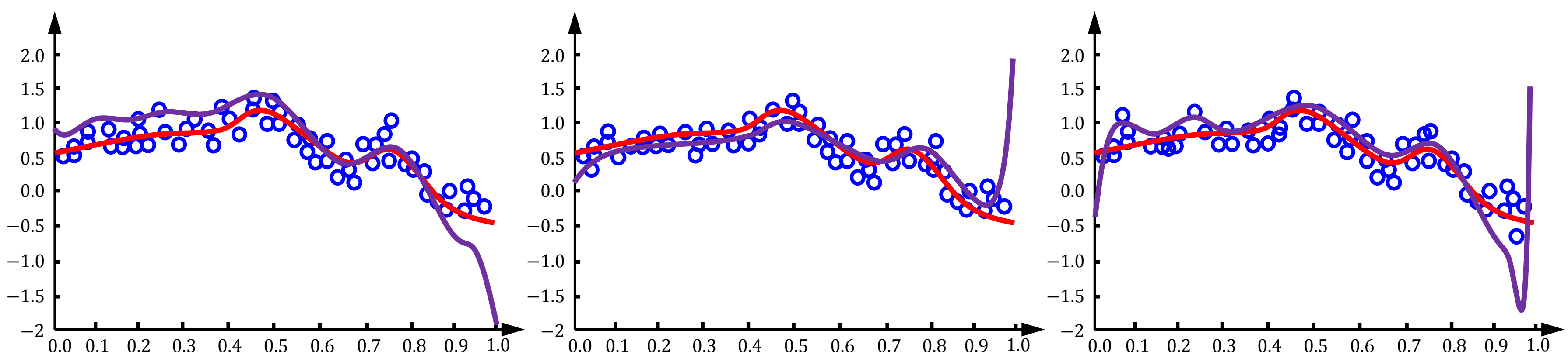
https://jkeun.github.io/2018-05-03/bias-variance-tradeoff/

Example: Bias-Variance Trade-Off

➢ **Large bias, small variance – poor approximation but stable**



➢ **Small bias, large variance – good approximation but instable**

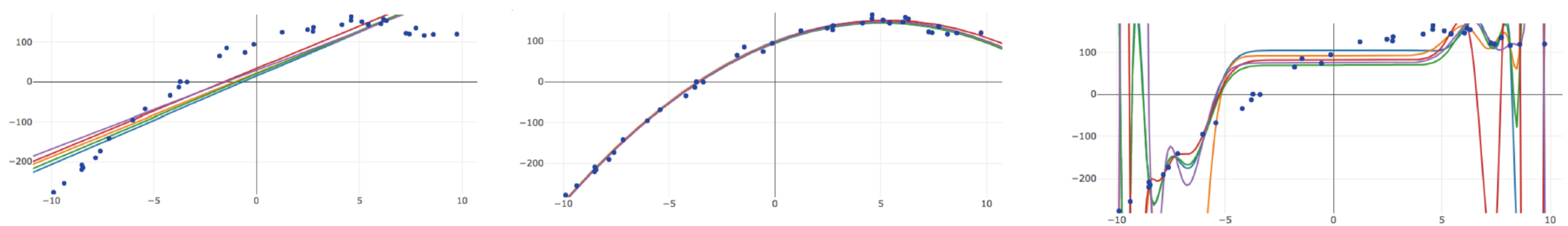


http://norman3.github.io/prml/docs/chapter03/2.html

Bias-Variance Trade-Off

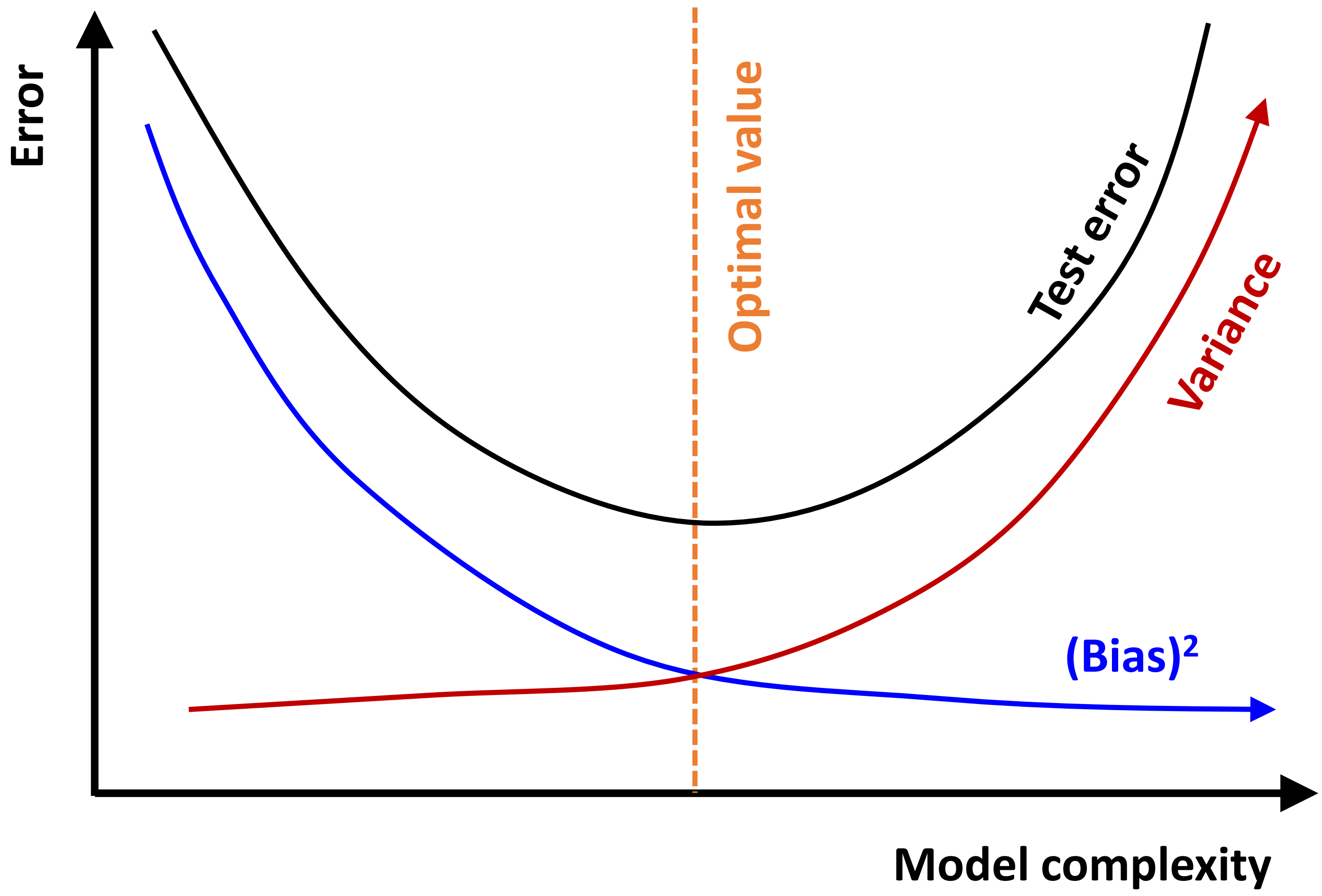
- **Bias: the model does not fit the training data effectively.**
  - **Solution**: use a more complicated model.
- **Variance: the models can fit the training data but does not fit the test data.**
  - **Solution** : use a less complicated model.

Estimated Model Variance



Bias

Bias-Variance Trade-Off



How to Control Model Complexity?

Regularization

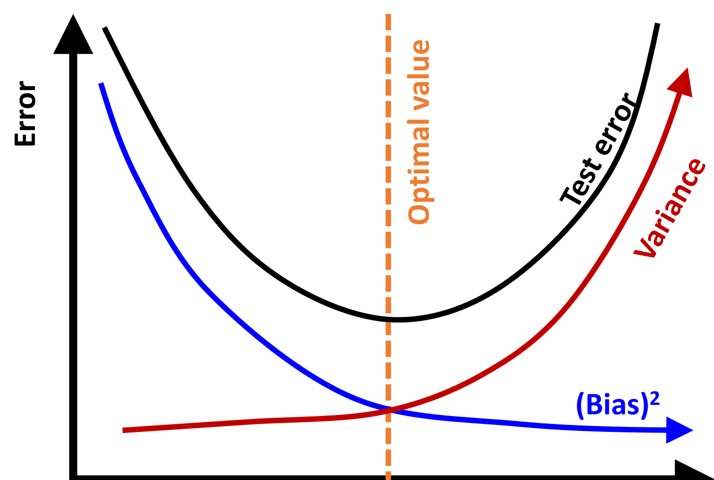
Parametrically Controlling the *Model Complexity*

- Tradeoff:
  - **Increase bias**
  - **Decrease variance**

How to Control λ?

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}\left(f\left(x^{(i)}\right), y^{(i)}\right) + \lambda \Omega(\theta)$$

- **The value of λ determines the bias-variance trade-off.**
  - Large value → more bias → less variance → more generalization



Q&A

Behavior of True Risk

- The regression model has true function and noise.

$$y = f^*(x) + \varepsilon, \text{ where } \varepsilon \sim \mathcal{N}(0, \sigma^2)$$

$$\mathbb{E}[\varepsilon] = \mathbf{0}$$
$$\mathbb{E}[\varepsilon^2] = \boldsymbol{\sigma}^2$$

- True risk error expectation function

Dataset and noise

$$\mathbb{E}_{\boldsymbol{D},\boldsymbol{\varepsilon}}\left[\left(\boldsymbol{f}^*(\boldsymbol{x}) + \boldsymbol{\varepsilon} - \boldsymbol{h}(\boldsymbol{x})\right)^2\right]$$

True function + noise

Learned from data

Lemma for Expectation

- **Let** X **be a random variable with a probability** $P(X)$
- **Let** $\boldsymbol{\bar{X} = E[X]}$ **be the average value of** $\boldsymbol{\bar{X}}$

- $\mathbb{E}[(X-\bar{X})^2] = \mathbb{E}[(X^2 - 2X\bar{X} + \bar{X}^2)]$

$$= \mathbb{E}[X^2] - 2\mathbb{E}[X]\mathbb{E}[\bar{X}] + \mathbb{E}[\bar{X}^2]$$

$$= \mathbb{E}[X^2] - 2\bar{X}\mathbb{E}[\bar{X}] + \mathbb{E}[\bar{X}^2]$$

$$= \mathbb{E}[X^2] - 2\bar{X}^2 + \mathbb{E}[\bar{X}^2]$$

$$= \mathbb{E}[X^2] - 2\bar{X}^2 + \bar{X}^2 = \mathbb{E}[X^2] - \bar{X}^2$$

- **Corollary:** $\mathbb{E}[X^2] = \mathbb{E}[(X-\bar{X})^2] + \bar{X}^2$

$$= \mathbb{E}[(X - \mathbb{E}(X))^2] + \big(\mathbb{E}(X)\big)^2$$

Bias-Variance-Noise Decomposition

$$y = f^*(x) + \varepsilon$$
$$h = h(x)$$
$$\mathbb{E}[X^2] = \mathbb{E}[(X - \mathbb{E}[X])^2] + (\mathbb{E}[X])^2$$

- $\mathbb{E}_{D,\varepsilon}\left[(y-h)^2\right] = \mathbb{E}_{D,\varepsilon}\left[(h-y)^2\right]$

$$= \mathbb{E}[h^2 - 2hf + y^2]$$
$$= \mathbb{E}\left[h^2\right] - 2\mathbb{E}[h]\mathbb{E}[y] + \mathbb{E}\left[y^2\right]$$
$$= \mathbb{E}\left[(h - \mathbb{E}[h])^2\right] + (\mathbb{E}[h])^2 - 2\mathbb{E}[h]\mathbb{E}[y] + \mathbb{E}\left[(y - \mathbb{E}[y])^2\right] + (\mathbb{E}[y])^2$$
$$= \mathbb{E}\left[(h - \mathbb{E}[h])^2\right] + (\mathbb{E}[h])^2 - 2\mathbb{E}[h]f^*(x) + \mathbb{E}\left[(y - f^*(x))^2\right] + (f^*(x))^2$$

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

Variance (Bias)^2 Noise

Bias-Variance-Noise Decomposition

$$y = f^*(x) + \varepsilon$$
$$h = h(x)$$
$$\mathbb{E}[X^2] = \mathbb{E}[(X - \mathbb{E}[X])^2] + (\mathbb{E}[X])^2$$

- $\mathbb{E}_{D,\varepsilon}\left[(h-y)^2\right]$

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

$$= var[h] + (\mathbb{E}[h] - f^*(x))^2 + \mathbb{E}\left[(y - f^*(x))^2\right]$$

$$= var[h] + bias(h)^2 + \mathbb{E}\left[(y - f^*(x))^2\right]$$

$$= var[h] + bias(h)^2 + \mathbb{E}\left[\varepsilon^2\right]$$

$$= var[h] + bias(h)^2 + \sigma^2$$

Variance (Bias)^2 Noise

Bias-Variance-Noise Decomposition

- Expected prediction error = Variance + Bias^2 + Noise

- Variance: $\mathbb{E}\left[\left(\boldsymbol{h}(\boldsymbol{x}) - \mathbb{E}[\boldsymbol{h}(\boldsymbol{x})]\right)^2\right]$
  - Describes how much varies from one training set to another

- Bias: $\mathbb{E}[\boldsymbol{h}(\boldsymbol{x})] - \boldsymbol{f}^*(\boldsymbol{x})$
  - Describes the average error of $\boldsymbol{h}(\boldsymbol{x})$

- Noise: $\mathbb{E}\left[\left(\boldsymbol{y} - \boldsymbol{f}^*(\boldsymbol{x})\right)^2\right] = \mathbb{E}\left[\boldsymbol{\varepsilon}^2\right] = \boldsymbol{\sigma}^2$
  - Describes how much y varies from $\boldsymbol{f}^*(\boldsymbol{x})$

Quiz: Bias-Variance Trade-Off

- **Match each of the following:**
- $\mathbb{E}[\boldsymbol{y}]$
- $\mathbb{E}[\boldsymbol{\varepsilon}^2]$
- $\mathbb{E}\left[(\mathbb{E}[\boldsymbol{h}(\boldsymbol{x})] - \mathbb{E}[\boldsymbol{y}])^2\right]$
- $\mathbb{E}\left[\boldsymbol{\varepsilon}(\boldsymbol{h}(\boldsymbol{x}) - \boldsymbol{y})^2\right]$

A. 0

B. Bias^2

C. Model variance

D. Observation variance

E. $f^*(x)$

F. $f^*(x) + \varepsilon$

Quiz: Bias-Variance Trade-Off

- **Match each of the following:**

- $\mathbb{E}[\boldsymbol{y}]$
- $\mathbb{E}[\boldsymbol{\varepsilon}^2]$
- $\mathbb{E}\left[(\mathbb{E}[\boldsymbol{h}(\boldsymbol{x})] - \mathbb{E}[\boldsymbol{y}])^2\right]$
- $\mathbb{E}\left[\boldsymbol{\varepsilon}(\boldsymbol{h}(\boldsymbol{x}) - \boldsymbol{y})^2\right]$

A. 0

B. Bias^2

C. Model variance

D. Observation variance

E. $f^*(x)$

F. $f^*(x) + \varepsilon$