# Generalization, Regularization
# Bias-variance trade-off

Iran University of Science and Technology

By: M. S. Tahaei Ph.D.

Fall 2024

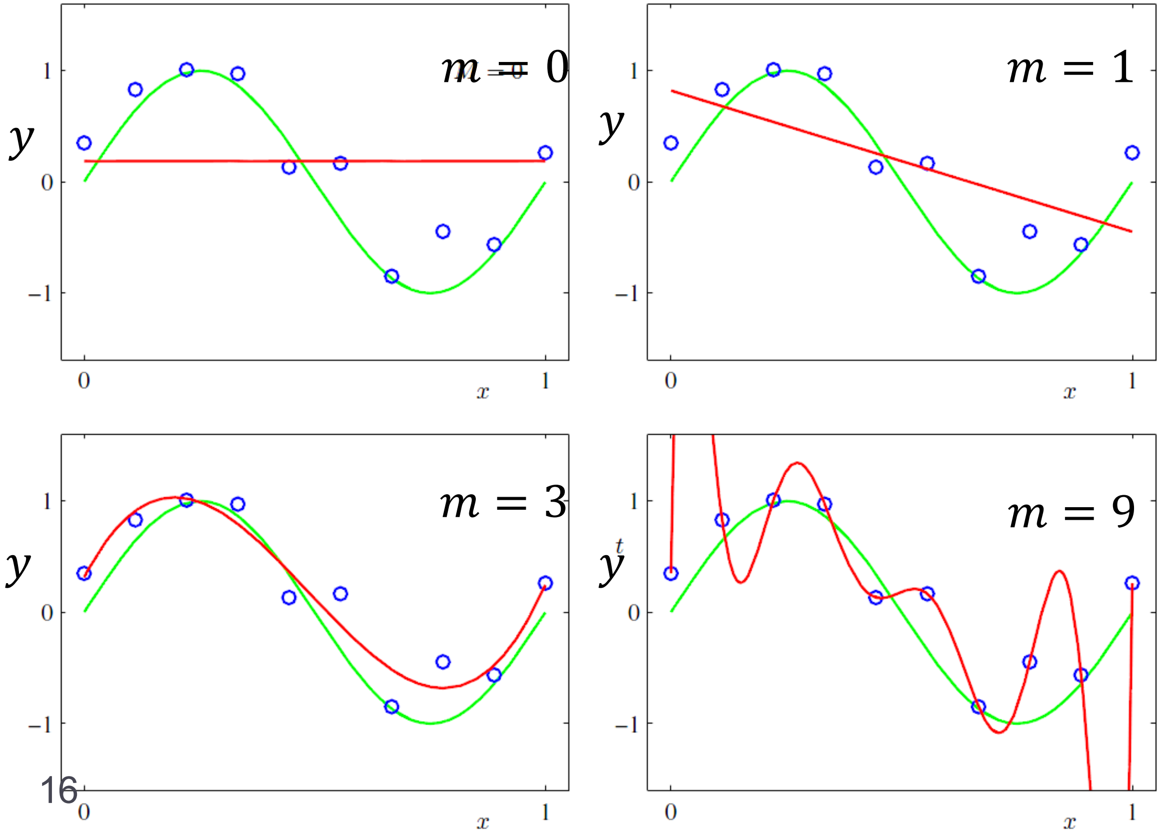
Courtesy: slides are adopted partly from Dr. Soleymani, Sharif University

# Outline

- Generalization
- Regularization
- Bias-variance

# Model complexity

- Example:
  - Polynomials with larger $m$ are becoming increasingly tuned to the random noise on the target values.



# Evaluation and model selection

- **Evaluation**:
  - We need to measure how well the learned function can predicts the target for unseen examples

- **Model selection:**
  - Most of the time we need to select among a set of models
    - Example: polynomials with different degree $m$
  - and thus we need to evaluate these models first

# Avoiding over-fitting

- Determine a suitable value for model complexity
  - **Simple hold-out method**
  - **Cross-validation**

- Regularization (Occam's Razor)
  - Explicit preference towards simple models
  - Penalize for the model complexity in the objective function

# Simple hold-out: model selection

- Steps:
  - Divide training data into <u>training</u> and <u>validation set</u> $v\_set$
  - Use only the training set to train a set of models
  - Evaluate each learned model on the validation set
    - $J_v(\boldsymbol{w}) = \frac{1}{|v\_set|} \quad {}_{i \in v\_set} \left( y^{(i)} - f\left(\boldsymbol{x}^{(i)}; \boldsymbol{w}\right)\right)^2$
  - Choose the best model based on the validation set error

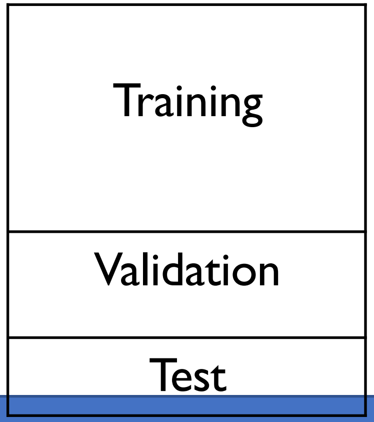
- Usually, too wasteful of valuable training data
  - Training data may be limited.
  - On the other hand, small validation set give a relatively noisy estimate of performance.

# Simple hold out: training, validation, and test sets

- Simple hold-out chooses the model that minimizes error on validation set.

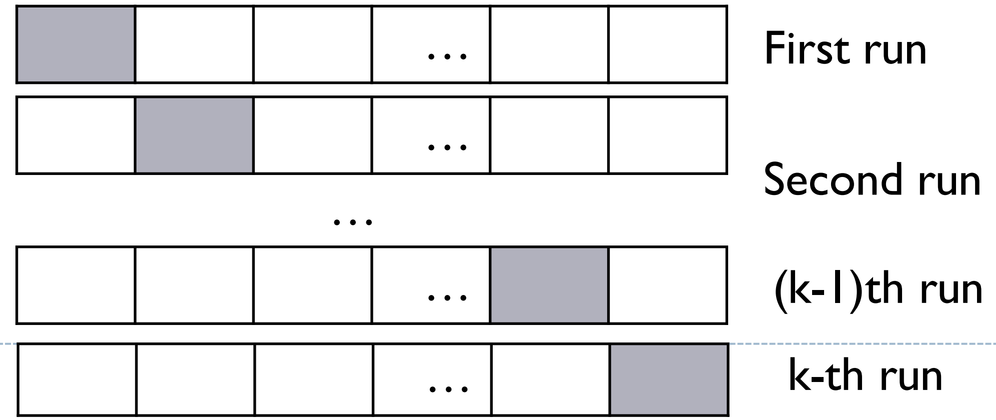
- $J_v(\boldsymbol{w})$ is likely to be an optimistic estimate of generalization error.
  - extra parameter (e.g.,degree of polynomial) is fit to this set.

- Estimate generalization error for the test set
  - performance of the selected model is finally evaluated on the test set



# Cross-Validation (CV): Evaluation

- $k$-fold cross-validation steps:
  - Shuffle the dataset and randomly partition training data into $k$ groups of approximately equal size
  - for $i = 1$ to $k$
    - Choose the $i$-th group as the held-out validation group
    - Train the model on all but the $i$-th group of data
    - Evaluate the model on the held-out group
  - Performance scores of the model from $k$ runs are **averaged**.
    - The average error rate can be considered as an estimation of the true performance.



# Cross-Validation (CV): Model Selection

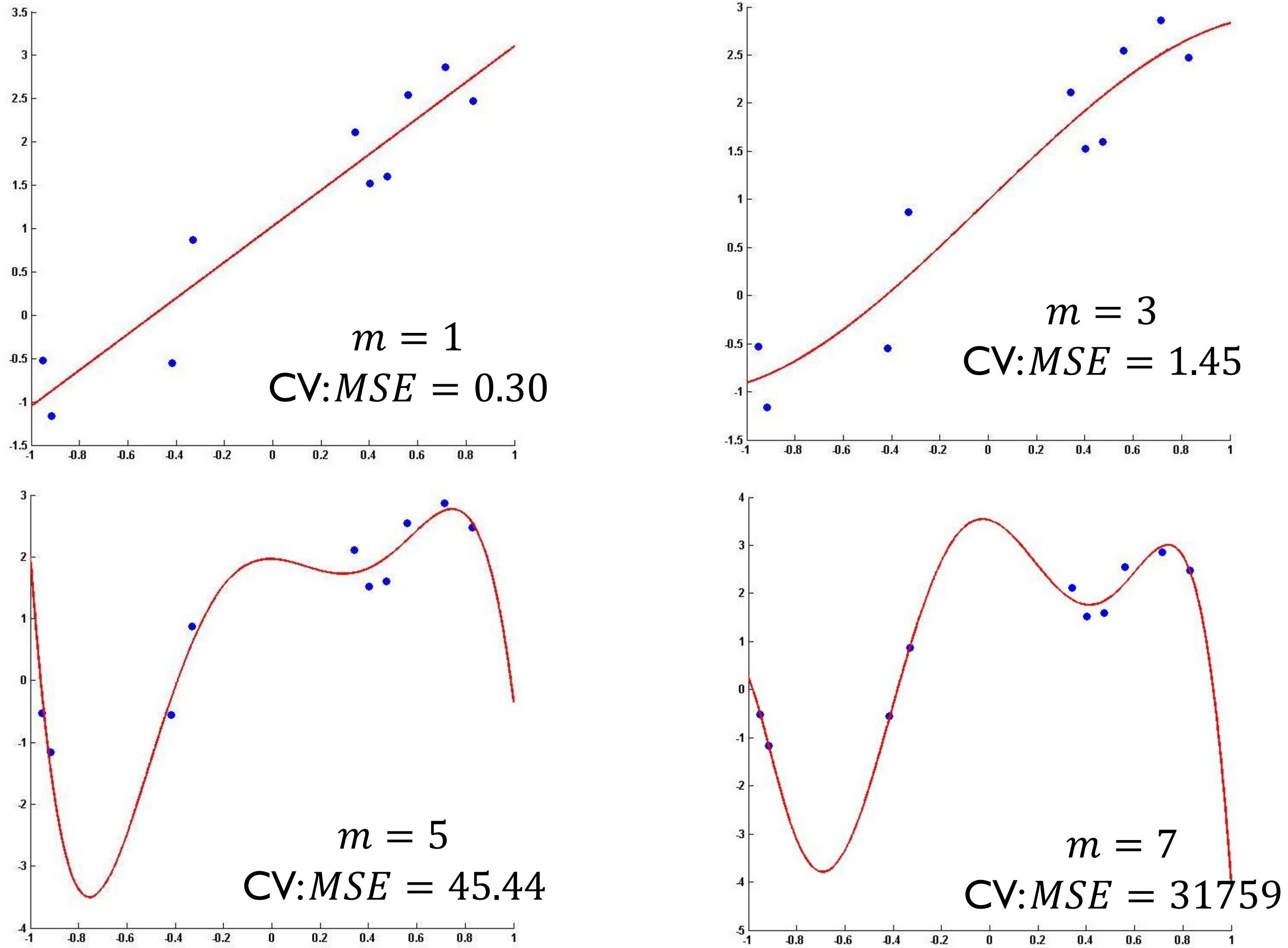
- For each model we first find the average error find by CV.
- The model with the best average performance is selected.

# Cross-validation: polynomial regression example

5-fold CV

100 runs

average



# Leave-One-Out Cross Validation (LOOCV)

- When data is particularly scarce, cross-validation with $k = N$
  - Leave-one-out treats each training sample in turn as a test example and all other samples as the training set.

- Use for small datasets
  - When training data is valuable
  - LOOCV can be time expensive as $N$ training steps are required.

# Regularization

- Adding a penalty term in the cost function to discourage the coefficients from reaching large values.

- Ridge regression (weight decay):

$$J(\boldsymbol{w}) = \sum_{i=1}^{n} \left( y^{(i)} - \boldsymbol{w}^T \boldsymbol{\phi}\left(\boldsymbol{x}^{(i)}\right) \right)^2 + \lambda \boldsymbol{w}^T \boldsymbol{w}$$

$$\boldsymbol{w} = \left( \boldsymbol{\Phi}^T \boldsymbol{\Phi} + \lambda \boldsymbol{I} \right)^{\boldsymbol{-1}} \boldsymbol{\Phi}^T \boldsymbol{y}$$

# Polynomial order

- Polynomials with larger $m$ are becoming increasingly tuned to the random noise on the target values.
  - magnitude of the coefficients typically gets larger by increasing $m$.

| | $M=0$ | $M=1$ | $M=6$ | $M=9$ |
|---|---|---|---|---|
| $w_0^\star$ | 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 |

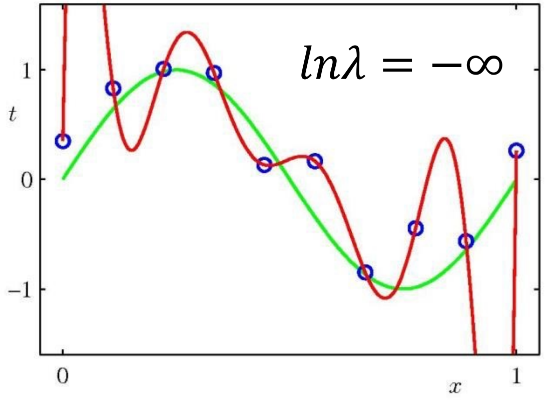
[Bishop]

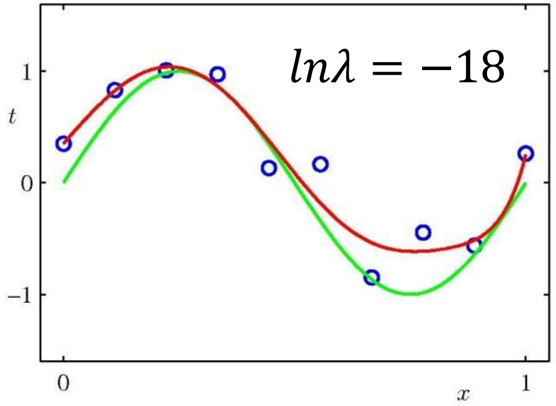
# Regularization parameter

$m = 9$

| | 0<br>$\ln\lambda = -\infty$ | $\ln\lambda = -18$ | 1<br>$\ln\lambda = 0$ |
|---|---|---|---|
| $w_0$ | 0.35 | 0.35 | 0.13 |
| $w_1$ | 232.37 | 4.74 | -0.05 |
| $w_2$ | -5321.83 | -0.77 | -0.06 |
| $w_3$ | 48568.31 | -31.97 | -0.05 |
| $w_4$ | -231639.30 | -3.89 | -0.03 |
| $w_5$ | 640042.26 | 55.28 | -0.02 |
| $w_6$ | -1061800.52 | 41.32 | -0.01 |
| $w_7$ | 1042400.18 | -45.95 | -0.00 |
| $w_8$ | -557682.99 | -91.53 | 0.00 |
| $w_9$ | 125201.43 | 72.68 | 0.01 |

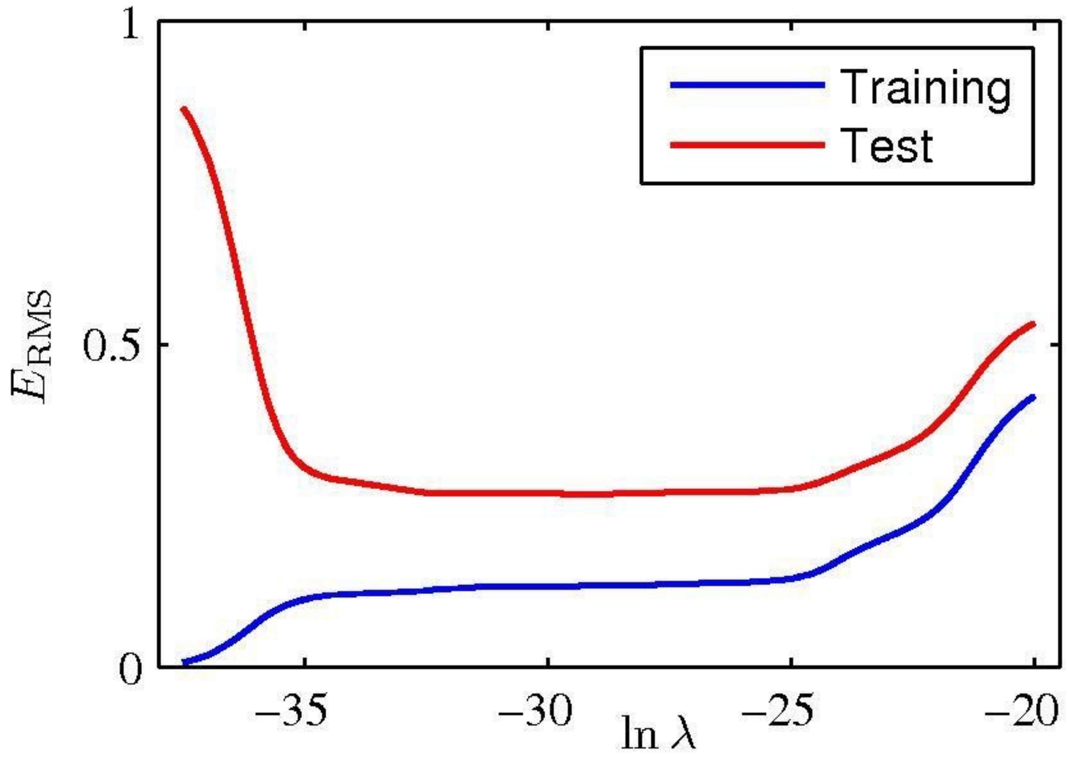
[Bishop]



# Regularization parameter

- Generalization
  - $\lambda$ now controls the effective complexity of the model and hence determines the degree of over-fitting




[Bishop]

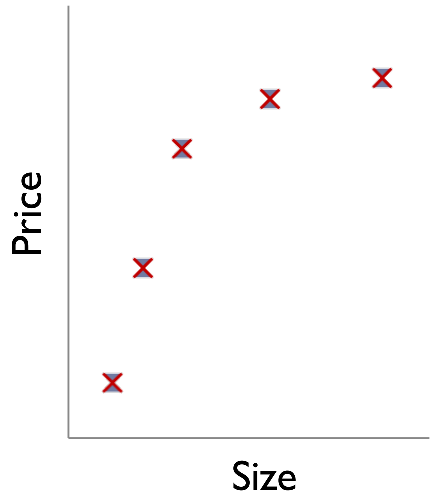
# The approximation-generailization trade-off

- Small true error shows good approximation of $f$ out of sample

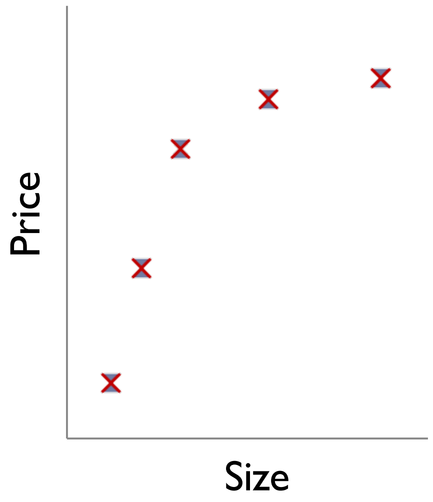
- More complex $\mathcal{H} \Rightarrow$ better chance of approximating $f$

- Less complex $\mathcal{H} \Rightarrow$ better chance of generalization out of $f$

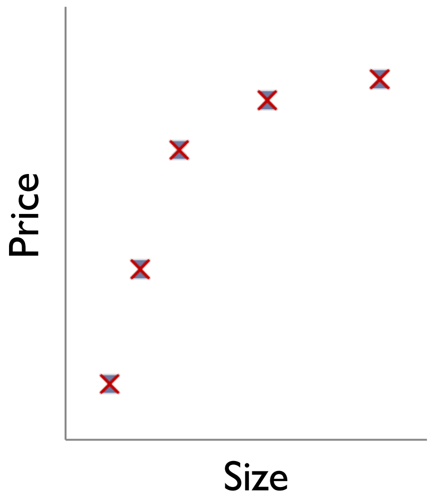
# Complexity of Hypothesis Space: Example




$w_0 + w_1 x$




$w_0 + w_1 x + w_2 x^2$



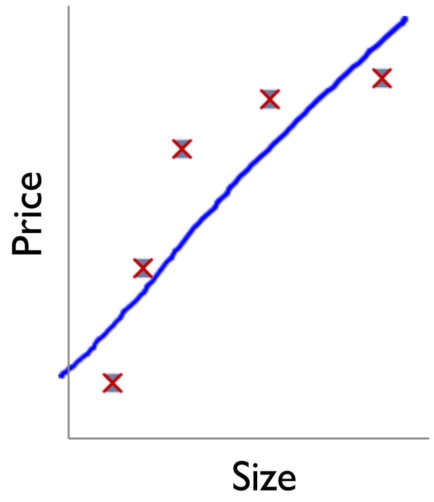

$w_0 + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4$

Less complex $\mathcal{H}$

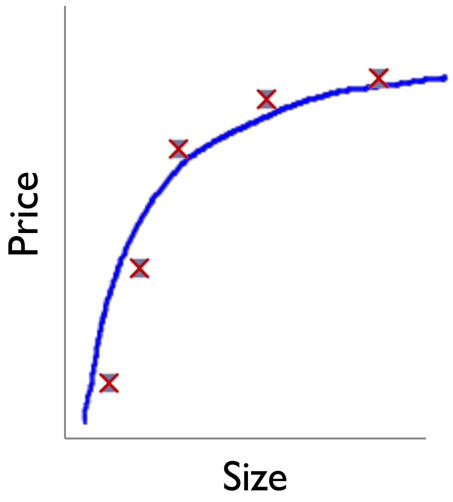
More complex $\mathcal{H}$

This example has been adapted from: Prof. Andrew Ng's slides

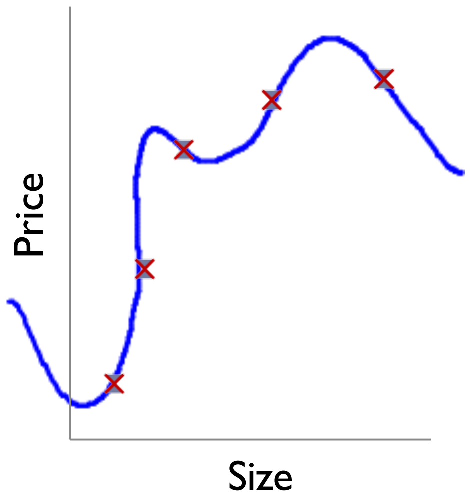
# Complexity of Hypothesis Space: Example




$w_0 + w_1 x$




$w_0 + w_1 x + w_2 x^2$




$w_0 + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4$

Less complex $\mathcal{H}$

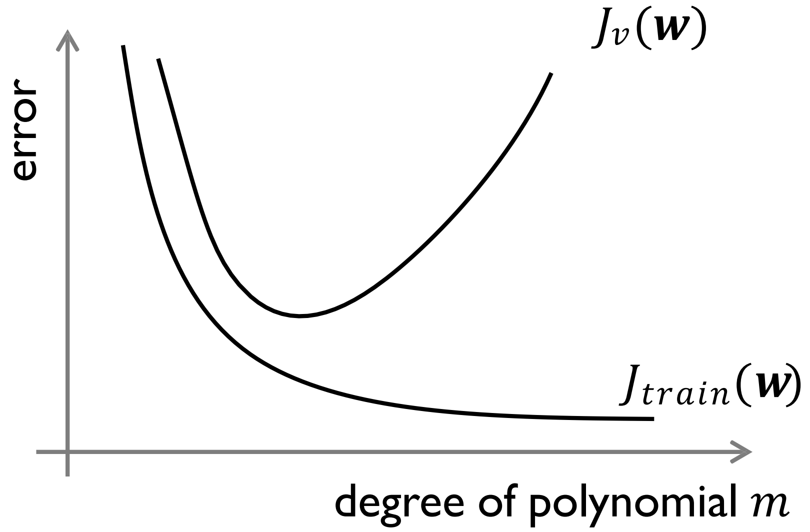
More complex $\mathcal{H}$

This example has been adapted from: Prof. Andrew Ng's slides

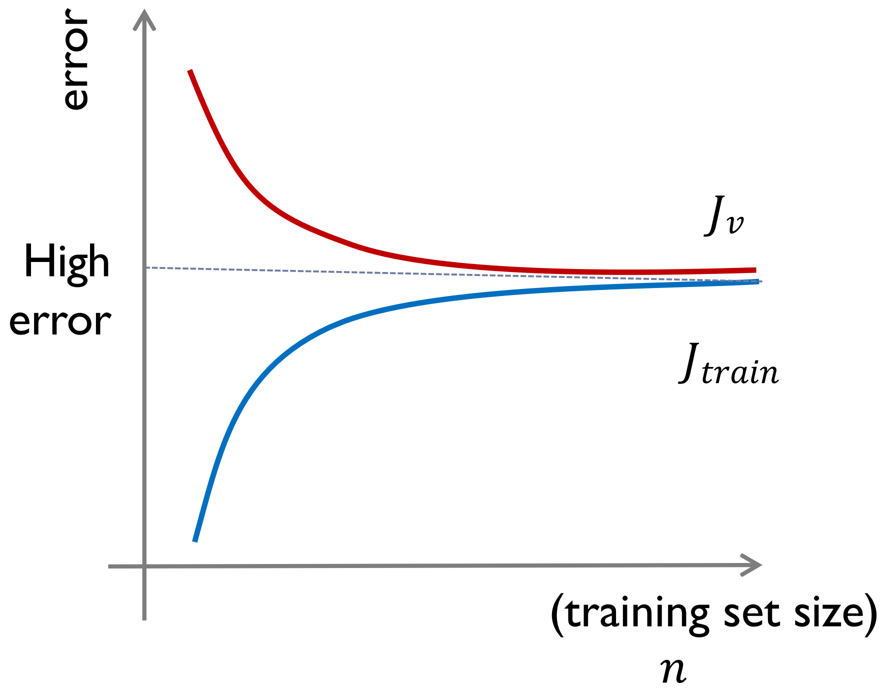
# Complexity of Hypothesis Space

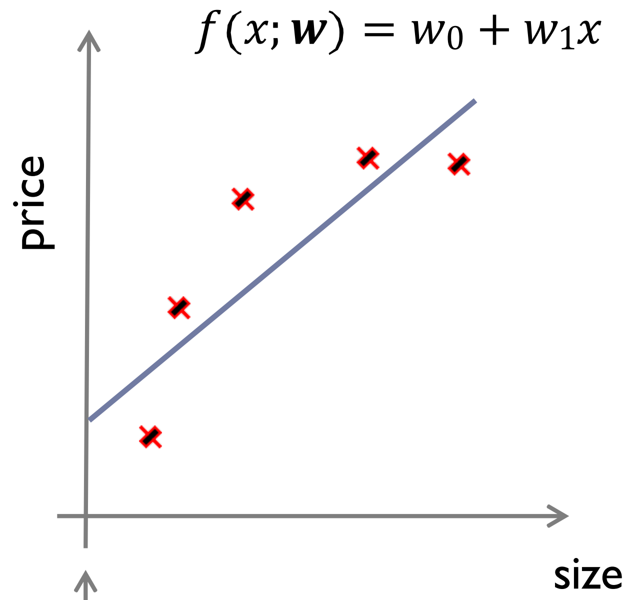
- Less complex $\mathcal{H}$:
  - $J_{train}(\boldsymbol{w}) \approx J_v(\boldsymbol{w})$ and $J_{train}(\boldsymbol{w})$ is very high

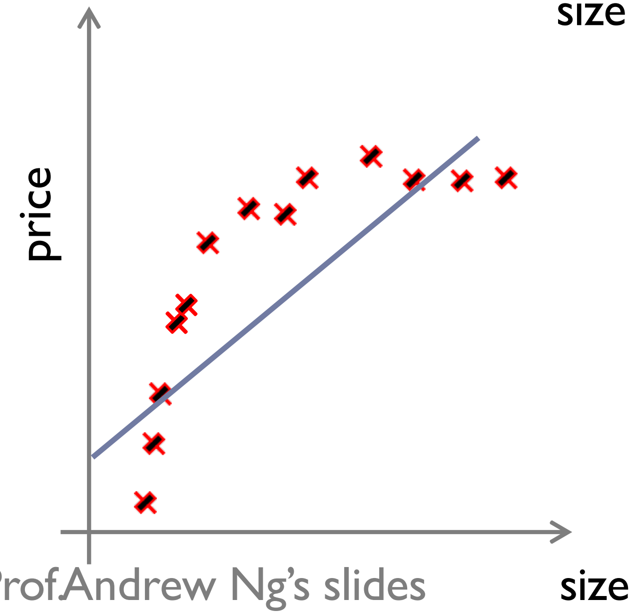
- More complex $\mathcal{H}$:
  - $J_{train}(\boldsymbol{w}) \ll J_v(\boldsymbol{w})$ and $J_{train}(\boldsymbol{w})$ is low



# Less complex $\mathcal{H}$

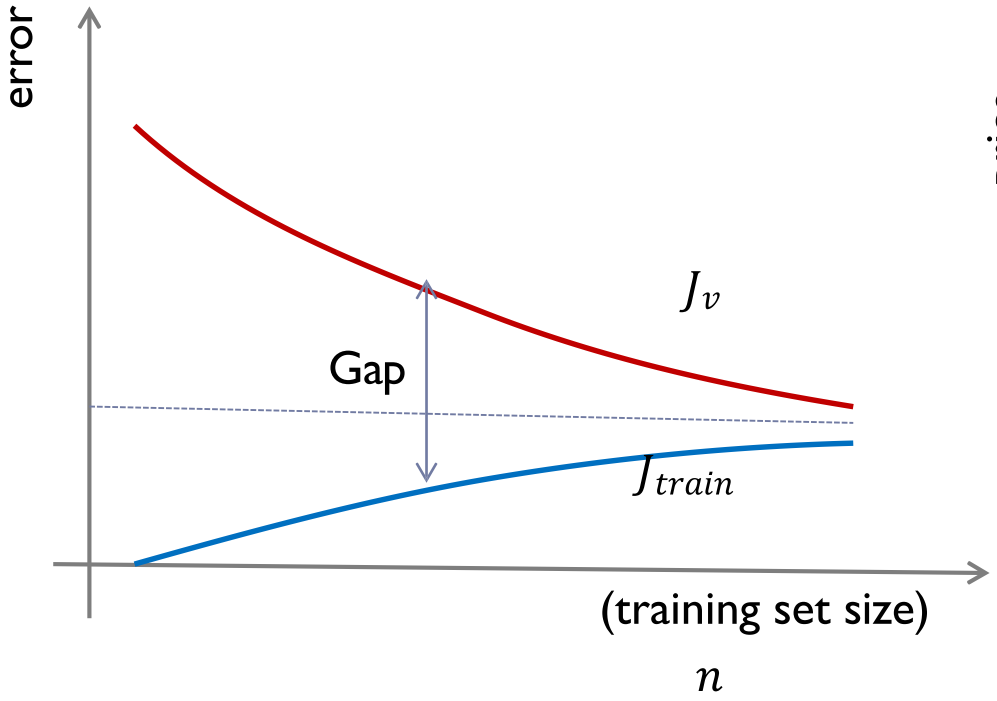


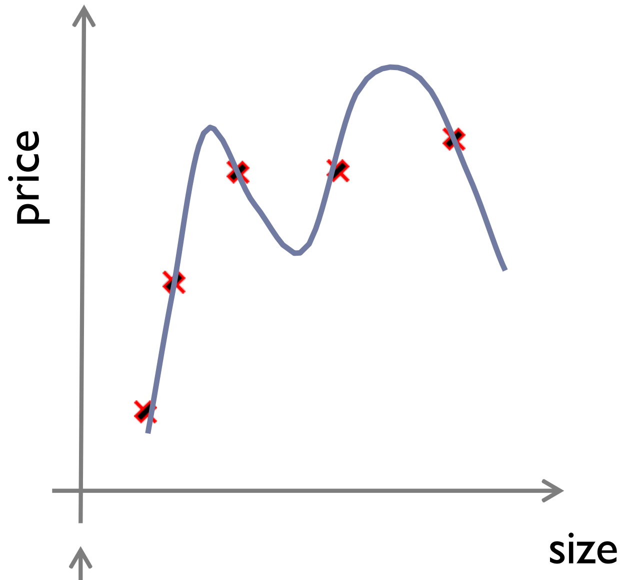



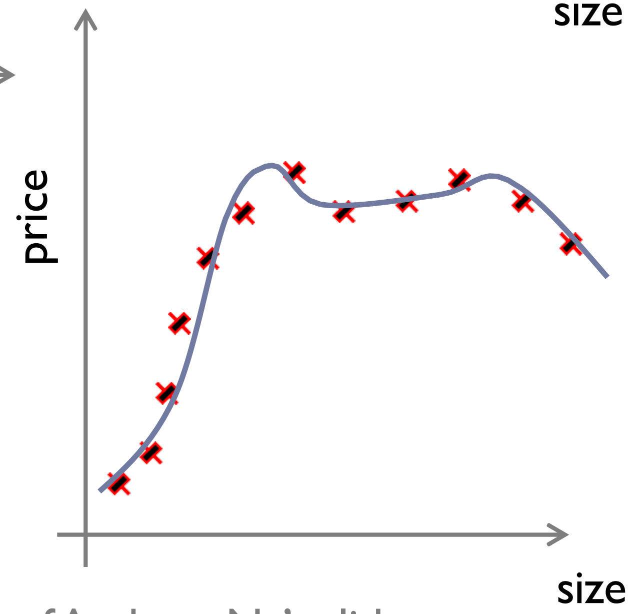

If model is very simple, getting more training data will not (by itself) help much.

# More complex $\mathcal{H}$




$$f(x;\boldsymbol{w}) = w_0 + w_1 x + \cdots w_{10} x^{10}$$



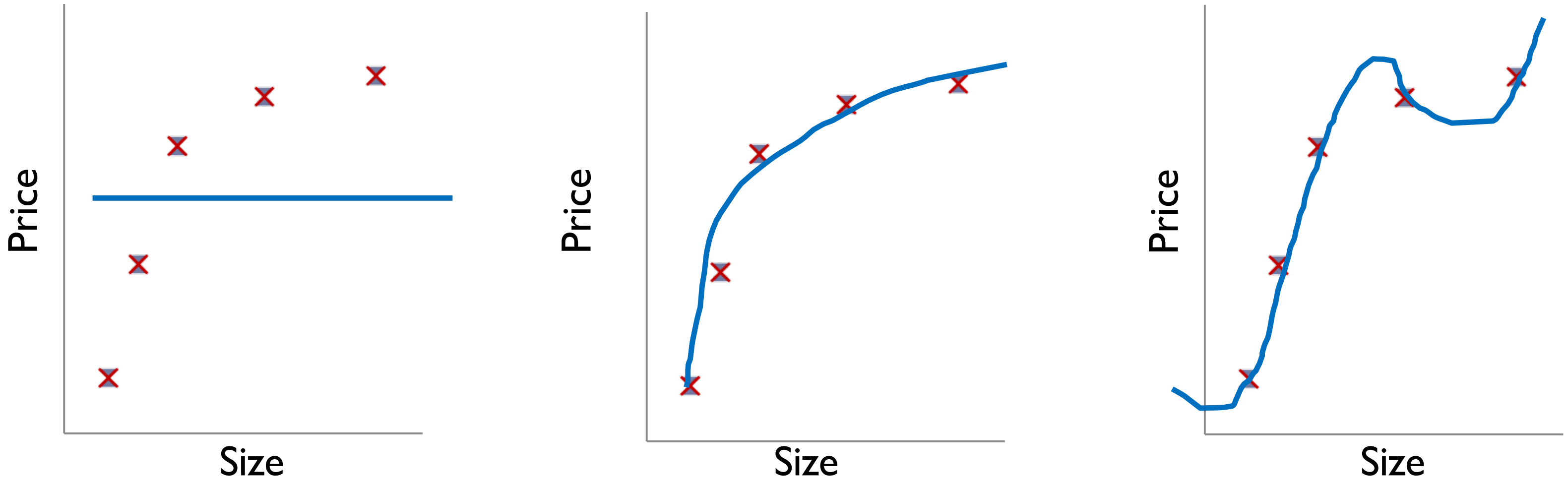




For more complex models, getting more training data is usually helps.

# Regularization: Example

$$f(x; \boldsymbol{w}) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4$$

$$J(\boldsymbol{w}) = \frac{1}{n}\left( \sum_{i=1}^{n} \left(y^{(i)} - f\left(x^{(i)}; \boldsymbol{w}\right)\right)^2 + \lambda \boldsymbol{w}^T \boldsymbol{w} \right)$$




Large $\lambda$
(Prefer to more simple models)

$w_1 = w_2 \approx 0$

Intermediate $\lambda$

Small $\lambda$
(Prefer to more complex models)

$\lambda = 0$

This example has been adapted from: Prof. Andrew Ng's slides

# Model complexity: Bias-variance trade-off

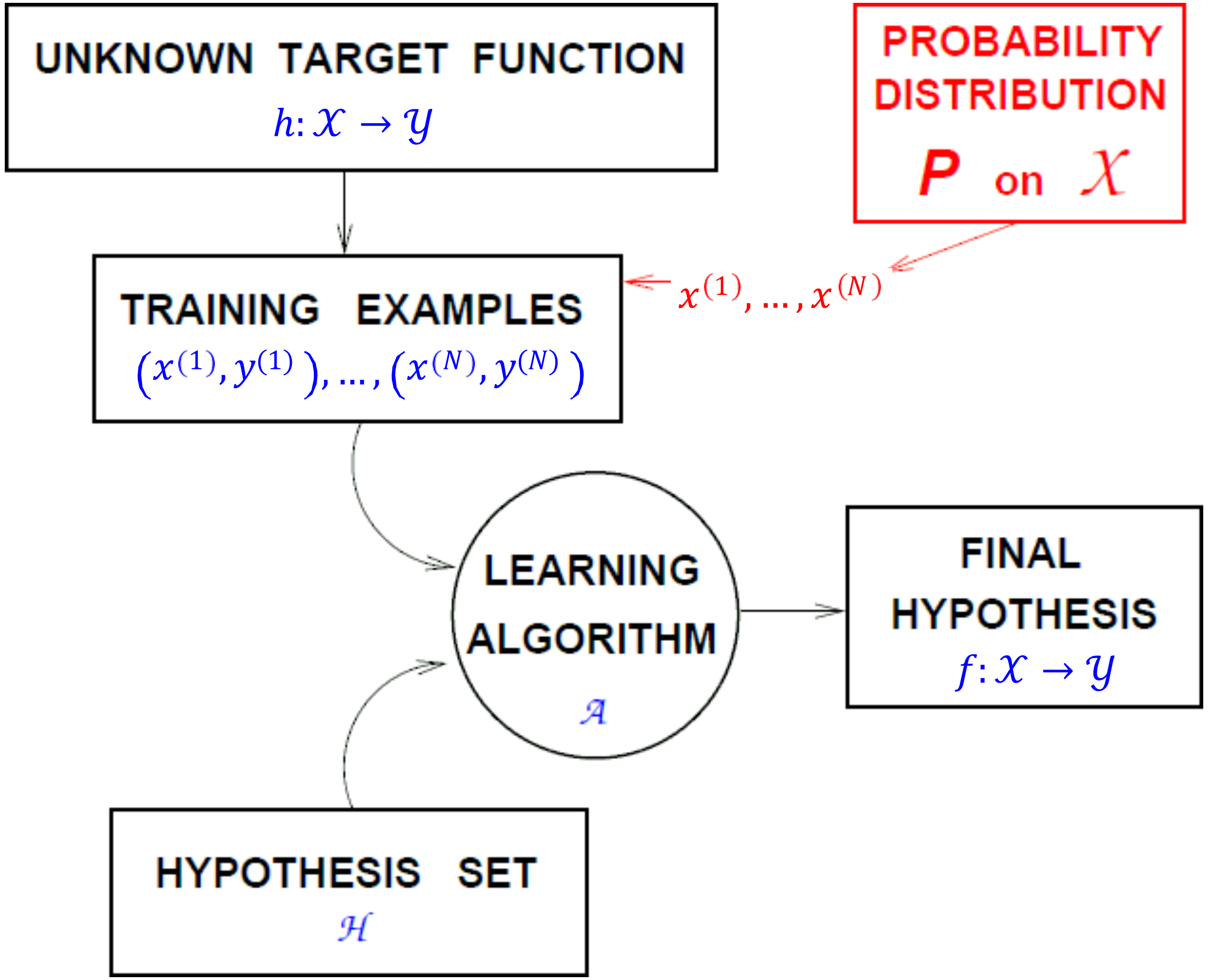
- Least squares, can lead to severe over-fitting if complex models are trained using data sets of limited size.

- A frequentist viewpoint of the model complexity issue, known as the *bias-variance* trade-off.

# Formal discussion on bias, variance, and noise

- **Best unrestricted regression function**
- **Noise**
- **Bias and variance**

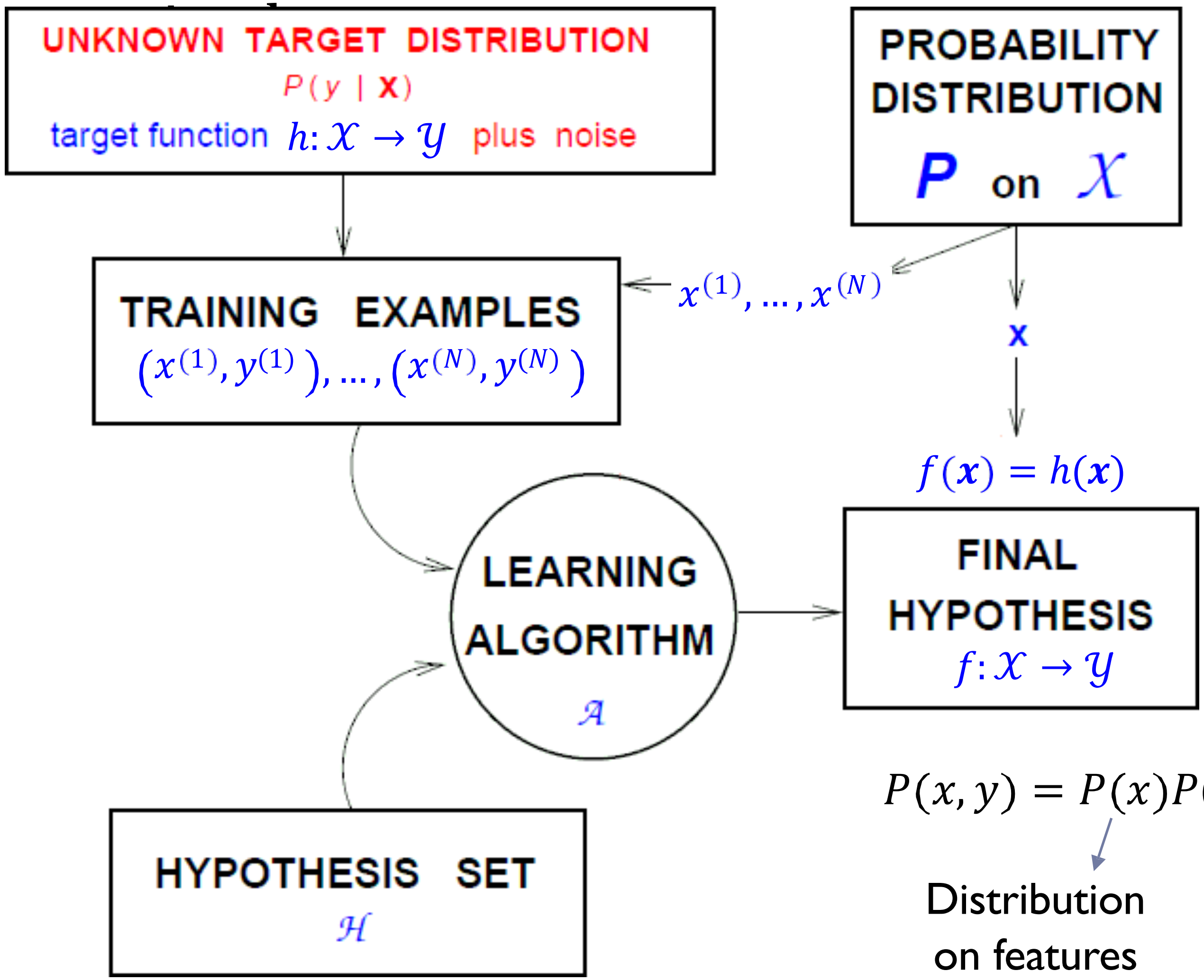
# The learning diagram: deterministic target




[Y.S Abou Mostafa, et. al]

# The learning diagram including noisy target

- Type




$$P(x, y) = P(x)P(y|x)$$

Distribution on features

Target distribution

[Y.S Abou Mostafa, et. al]

# Best unrestricted regression function

- If we know the joint distribution $P(\boldsymbol{x}, y)$ and no constraints on the regression function?
  - cost function: mean squared error

$$h^* = \underset{h:\mathbb{R}^d \to \mathbb{R}}{\operatorname{argmin}} \mathbb{E}_{\boldsymbol{x},y} \left[ \left( y - h(\boldsymbol{x}) \right)^2 \right]$$

$$h^*(\boldsymbol{x}) = \mathbb{E}_{y|\boldsymbol{x}}[y]$$

# Best unrestricted regression function: Proof

$$\mathbb{E}_{\boldsymbol{x},y}\left[\left(y-h(\boldsymbol{x})\right)^2\right] = \iint \left(y-h(\boldsymbol{x})\right)^2 p(\boldsymbol{x},y)d\boldsymbol{x}dy$$

- For each $\boldsymbol{x}$ separately minimize loss since $h(\boldsymbol{x})$ can be chosen independently for each different $\boldsymbol{x}$:

$$\frac{\delta\mathbb{E}_{\boldsymbol{x},y}\left[\left(y-h(\boldsymbol{x})\right)^2\right]}{\delta h(\boldsymbol{x})} = \int 2\left(y-h(\boldsymbol{x})\right)p(\boldsymbol{x},y)dy = 0$$

$$\Rightarrow h(\boldsymbol{x}) = \frac{\int yp(\boldsymbol{x},y)dy}{\int p(\boldsymbol{x},y)dy} = \frac{\int yp(\boldsymbol{x},y)dy}{p(\boldsymbol{x})} = \int yp(y|\boldsymbol{x})dy = \mathbb{E}_{y|\boldsymbol{x}}[y]$$

$$\Longrightarrow h^*(\boldsymbol{x}) = \mathbb{E}_{y|\boldsymbol{x}}[y]$$

# Error decomposition

$(\boldsymbol{x}, y) \sim P$

$h(\boldsymbol{x})$ :minimizes the expected loss

$$E_{true}\left(f_{\mathcal{D}}(\boldsymbol{x})\right) = \mathbb{E}_{\boldsymbol{x},y}\left[(f_{\mathcal{D}}(\boldsymbol{x}) - y)^2\right]$$

Expected loss

$$= \mathbb{E}_{\boldsymbol{x},y}\left[(f_{\mathcal{D}}(\boldsymbol{x}) - h(\boldsymbol{x}) + h(\boldsymbol{x}) - y)^2\right]$$

$$= \mathbb{E}_{\boldsymbol{x}}\left[\left(f_{\mathcal{D}}(\boldsymbol{x}) - h(\boldsymbol{x})\right)^2\right] + \mathbb{E}_{\boldsymbol{x},y}\left[(h(\boldsymbol{x}) - \boldsymbol{y})^2\right]$$

$$+2\underbrace{\mathbb{E}_{\boldsymbol{x},y}\left[\left(f_{\mathcal{D}}(\boldsymbol{x}) - h(\boldsymbol{x})\right)(h(\boldsymbol{x}) - y)\right]}$$

$$\mathbb{E}_{\boldsymbol{x}}\left[\left(f_{\mathcal{D}}(\boldsymbol{x}) - h(\boldsymbol{x})\right)\underbrace{\mathbb{E}_{y|\boldsymbol{x}}\left[(h(\boldsymbol{x}) - y)\right]}_{0}\right]$$

# Error decomposition

$(\boldsymbol{x}, y) \sim P$

$h(\boldsymbol{x})$ :minimizes the expected loss

$$E_{true}\big(f_{\mathcal{D}}(\boldsymbol{x})\big) = \mathbb{E}_{\boldsymbol{x},y}\left[(f_{\mathcal{D}}(\boldsymbol{x}) - y)^2\right]$$

$$= \mathbb{E}_{\boldsymbol{x},y}\left[(f_{\mathcal{D}}(\boldsymbol{x}) - h(\boldsymbol{x}) + h(\boldsymbol{x}) - y)^2\right]$$

$$= \mathbb{E}_{\boldsymbol{x}}\left[\left(f_{\mathcal{D}}(\boldsymbol{x}) - h(\boldsymbol{x})\right)^2\right] + \underbrace{\mathbb{E}_{\boldsymbol{x},y}\left[(h(\boldsymbol{x}) - \boldsymbol{y})^2\right]}_{\text{noise}}$$

$$+\ 0$$

- Noise shows the irreducible minimum value of the loss function

# Expectation of true error

$$E_{true}\left(f_{\mathcal{D}}(\boldsymbol{x})\right) = \mathbb{E}_{\boldsymbol{x},y}\left[(f_{\mathcal{D}}(\boldsymbol{x}) - y)^2\right]$$
$$= \mathbb{E}_{\boldsymbol{x}}\left[\left(f_{\mathcal{D}}(\boldsymbol{x}) - h(\boldsymbol{x})\right)^2\right] + noise$$

$$\mathbb{E}_{\mathcal{D}}\left[\mathbb{E}_{\boldsymbol{x}}\left[\left(f_{\mathcal{D}}(\boldsymbol{x}) - h(\boldsymbol{x})\right)^2\right]\right]$$
$$= \mathbb{E}_{\boldsymbol{x}}\left[\mathbb{E}_{\mathcal{D}}\left[\left(f_{\mathcal{D}}(\boldsymbol{x}) - h(\boldsymbol{x})\right)^2\right]\right]$$

We now want to focus on $\mathbb{E}_{\mathcal{D}}\left[\left(f_{\mathcal{D}}(\boldsymbol{x}) - h(\boldsymbol{x})\right)^2\right]$.

# The average hypothesis

$$\bar{f}(\boldsymbol{x}) \equiv E_{\mathcal{D}}[f_{\mathcal{D}}(\boldsymbol{x})]$$

$$\bar{f}(\boldsymbol{x}) \approx \frac{1}{K}\sum_{k=1}^{K} f_{\mathcal{D}^{(k)}}(\boldsymbol{x})$$

$K$ training sets (of size $N$) sampled from $P(\boldsymbol{x}, y)$:
$\mathcal{D}^{(1)}, \mathcal{D}^{(2)}, \ldots, \mathcal{D}^{(K)}$

# Using the average hypothesis

$$\mathbb{E}_{\mathcal{D}}\left[\left(f_{\mathcal{D}}(\boldsymbol{x}) - h(\boldsymbol{x})\right)^2\right]$$

$$= \mathbb{E}_{\mathcal{D}}\left[\left(f_{\mathcal{D}}(\boldsymbol{x}) - f(\boldsymbol{x}) + f(\boldsymbol{x}) - h(\boldsymbol{x})\right)^2\right]$$

$$= \mathbb{E}_{\mathcal{D}}\ \left(f_{\mathcal{D}}(\boldsymbol{x}) - f(\boldsymbol{x})\right)^2 + \left(f(\boldsymbol{x}) - h(\boldsymbol{x})\right)^2$$

# Bias and variance

$$\mathbb{E}_{\mathcal{D}}\left[\left(f_{\mathcal{D}}(\boldsymbol{x}) - h(\boldsymbol{x})\right)^2\right] = \underbrace{\mathbb{E}_{\mathcal{D}}\left[\left(f_{\mathcal{D}}(\boldsymbol{x}) - f(\boldsymbol{x})\right)^2\right]}_{\mathrm{var}(\boldsymbol{x})} + \underbrace{\left(f(\boldsymbol{x}) - h(\boldsymbol{x})\right)^2}_{\mathrm{bias}(\boldsymbol{x})}$$

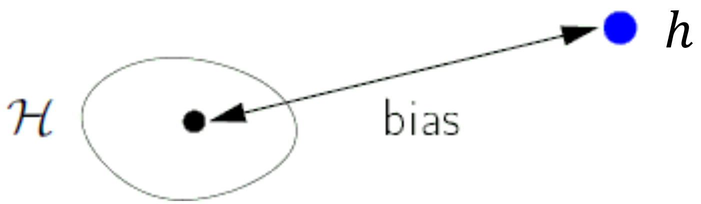
$$\mathbb{E}_{\boldsymbol{x}}\left[\mathbb{E}_{\mathcal{D}}\left[\left(f_{\mathcal{D}}(\boldsymbol{x}) - h(\boldsymbol{x})\right)^2\right]\right] = \mathbb{E}_{\boldsymbol{x}}[\mathrm{var}(\boldsymbol{x}) + \mathrm{bias}(\boldsymbol{x})]$$

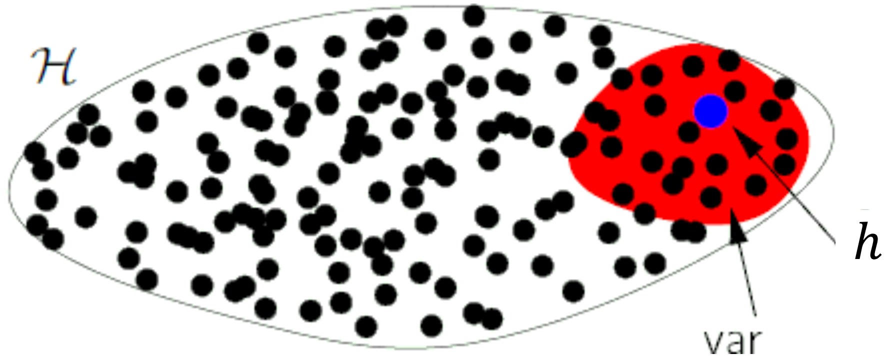
$$= \mathrm{var} + \mathrm{bias}$$

# Bias-variance trade-off

$$\text{var} = \mathbb{E}_{\boldsymbol{x}}\left[\mathbb{E}_{\mathcal{D}}\left[\left(f_{\mathcal{D}}(\boldsymbol{x}) - f(\boldsymbol{x})\right)^2\right]\right]$$

$$\text{bias} = \mathbb{E}_{\boldsymbol{x}}\left[f(\boldsymbol{x}) - h(\boldsymbol{x})\right]$$




More complex $\mathcal{H}$ $\Rightarrow$ lower bias but higher variance

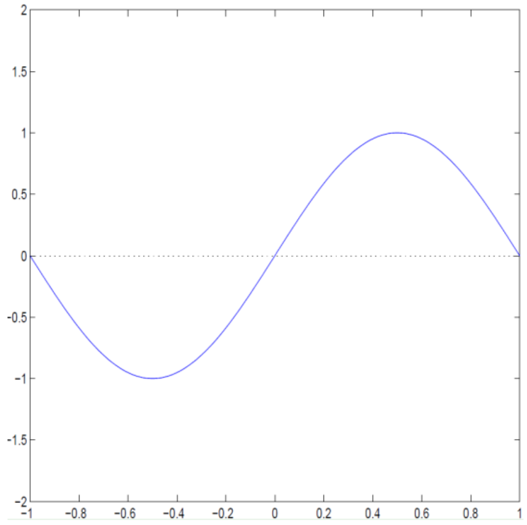
[Y.S Abou Mostafa, et. al]

# Example: sin target

- Only two training example $N = 2$

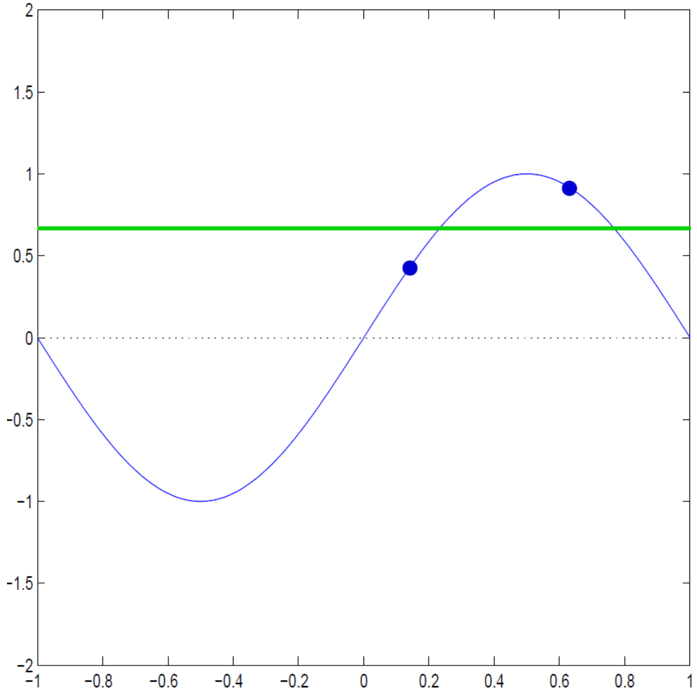
- Two models used for learning:
  - $\mathcal{H}_0$: $f(x) = b$
  - $\mathcal{H}_1$: $f(x) = ax + b$

- Which is better $\mathcal{H}_0$ or $\mathcal{H}_1$?

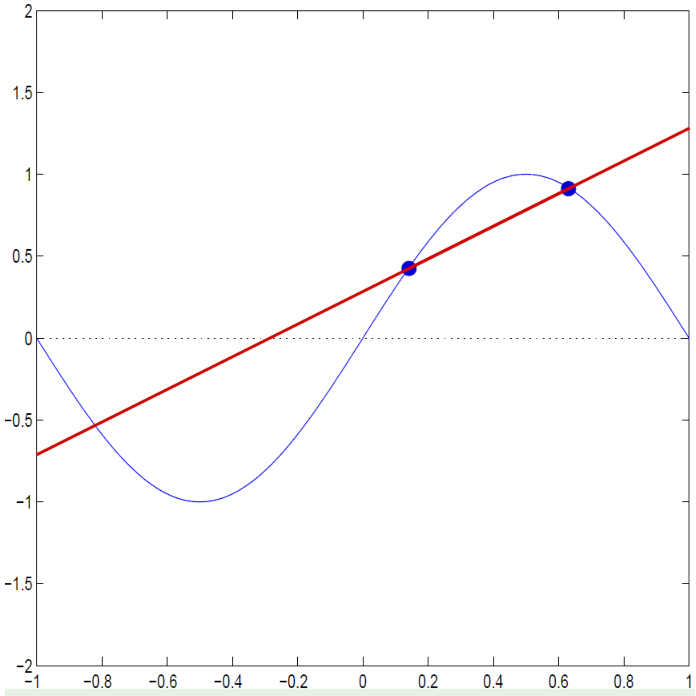


# Learning from a training set

$\mathcal{H}_0$

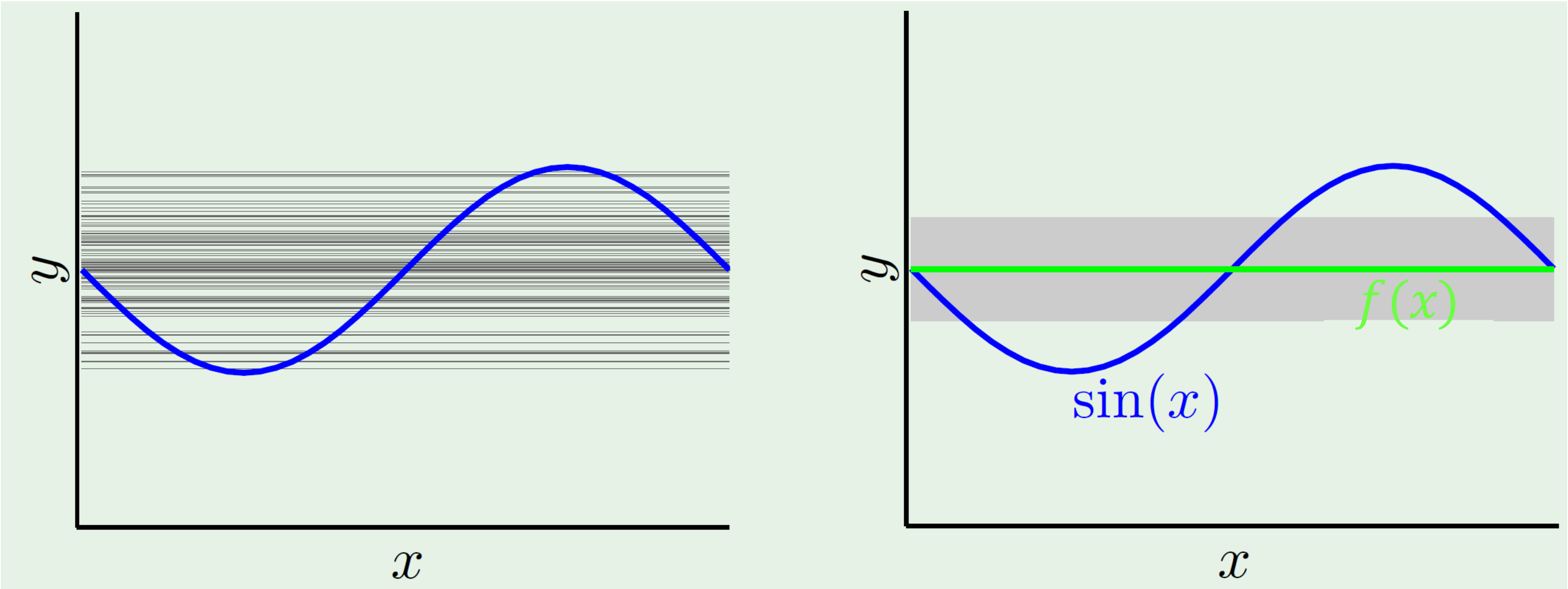



$\mathcal{H}_1$



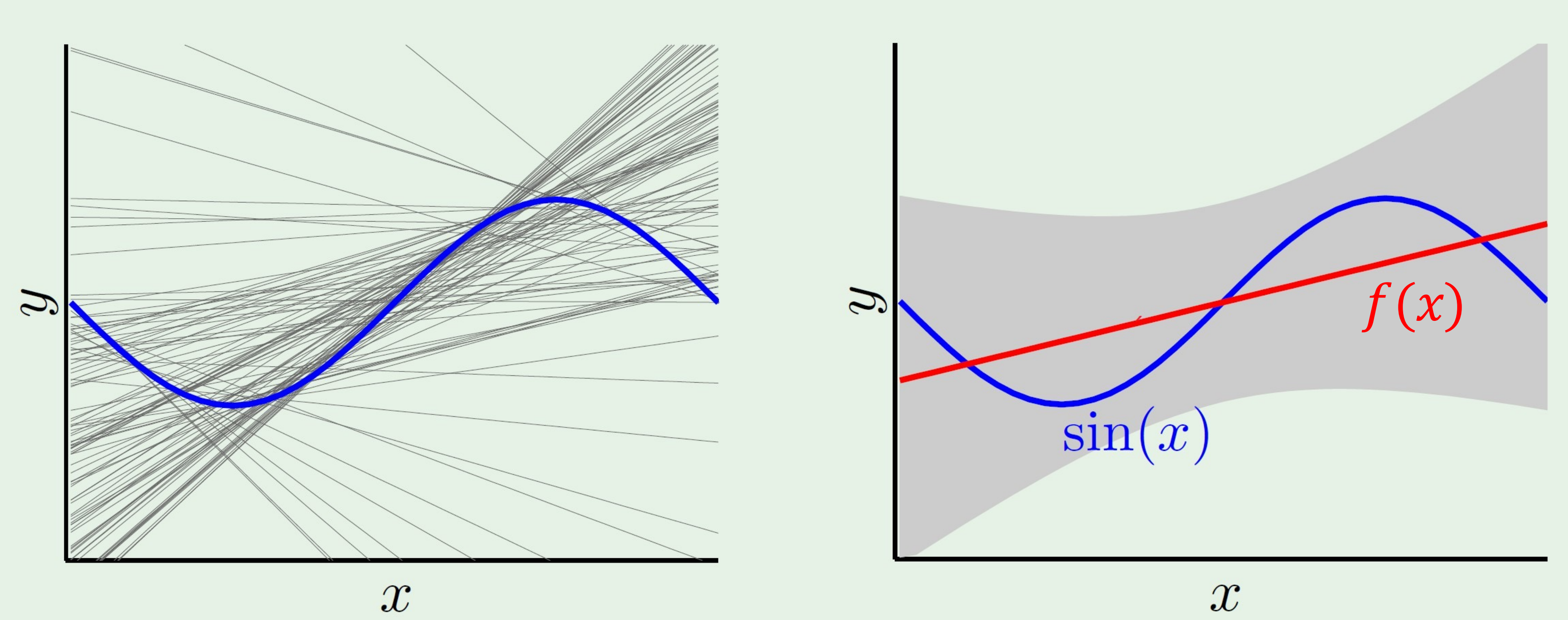

[Y.S Abou Mostafa, et. al]

# Variance $\mathcal{H}_0$



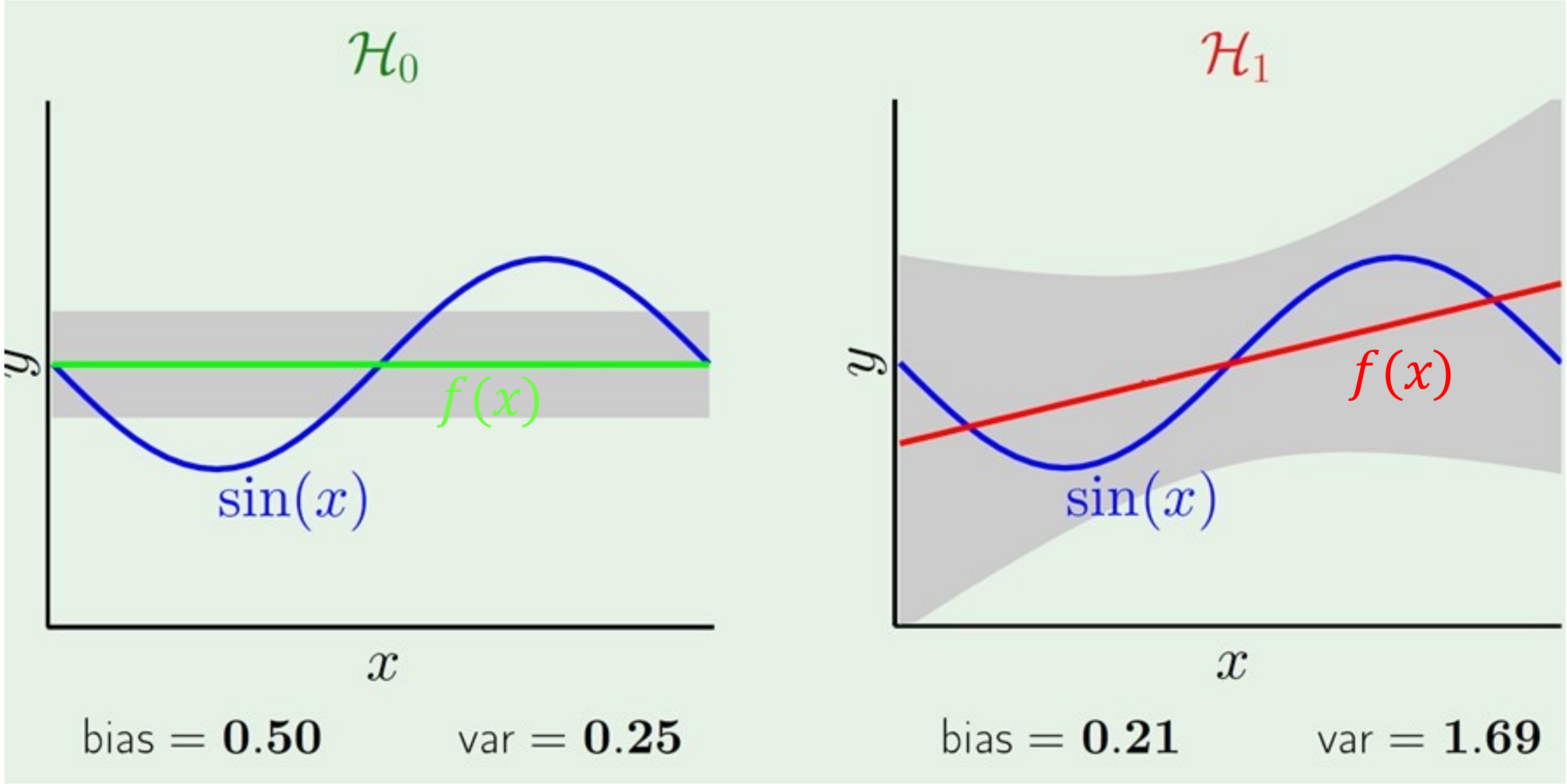

[Y.S Abou Mostafa, et. al]

# Variance $\mathcal{H}_1$




[Y.S Abou Mostafa, et. al]

# Which is better?



# Resource


1 C. M. Bishop, *Pattern Recognition and Machine Learning*.

2 Y. S. Abu-Mostafa, “Machine learning.” California Institute of Technology, 2012.

3 R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. 2001.