

# Deep Learning: Estimation & Decision Theory

**Ozan Özdenizci**

Institute of Theoretical Computer Science

[ozan.ozdenizci@igi.tugraz.at](mailto:ozan.ozdenizci@igi.tugraz.at)

Deep Learning VO - WS 23/24  
Lecture 2 - October 9th, 2023

# Today

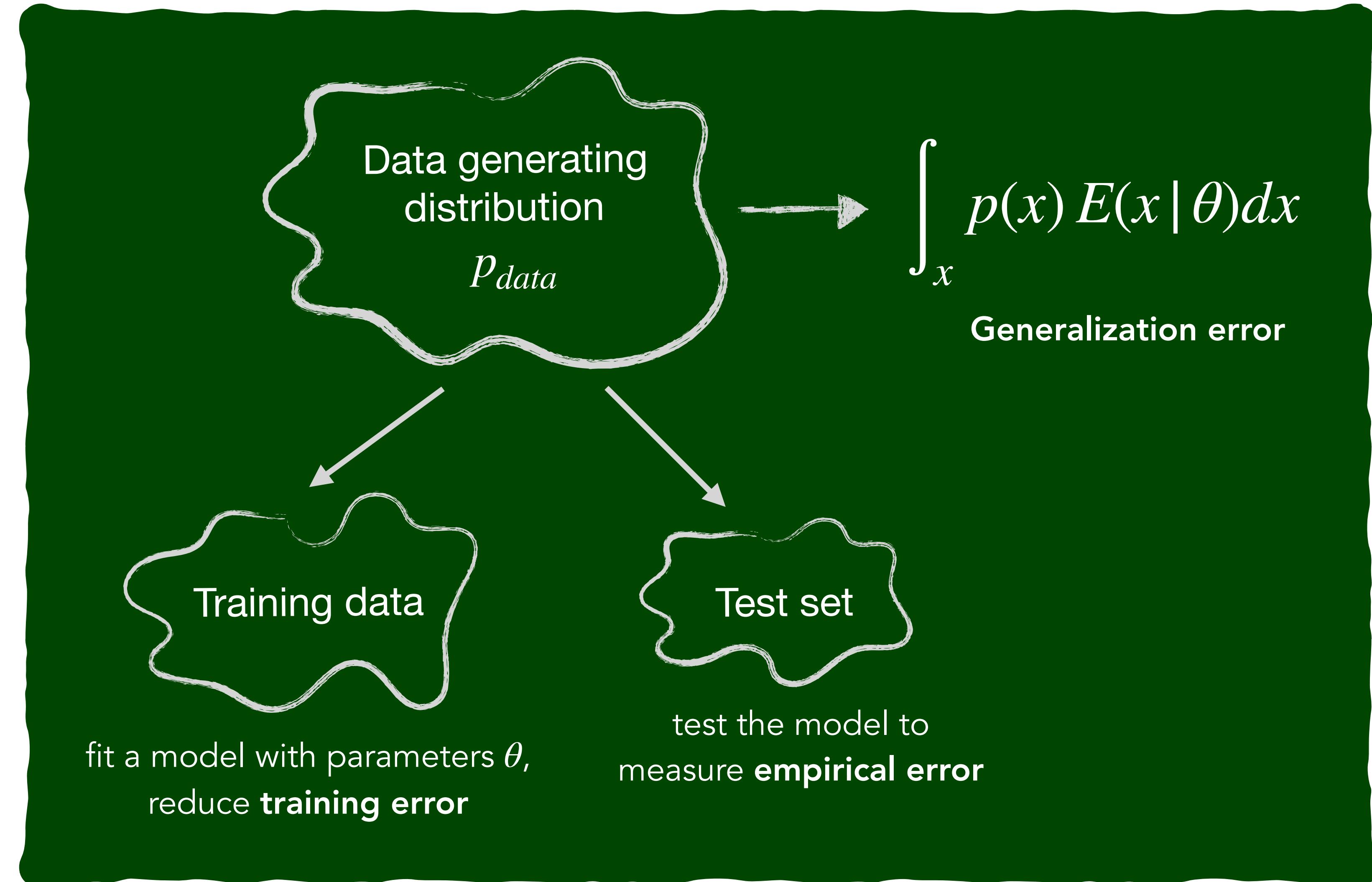
---

- Estimators
- Maximum Likelihood & Maximum A Posteriori Estimation
- Classification & Decision Theory

# Recap on Statistical Learning Theory

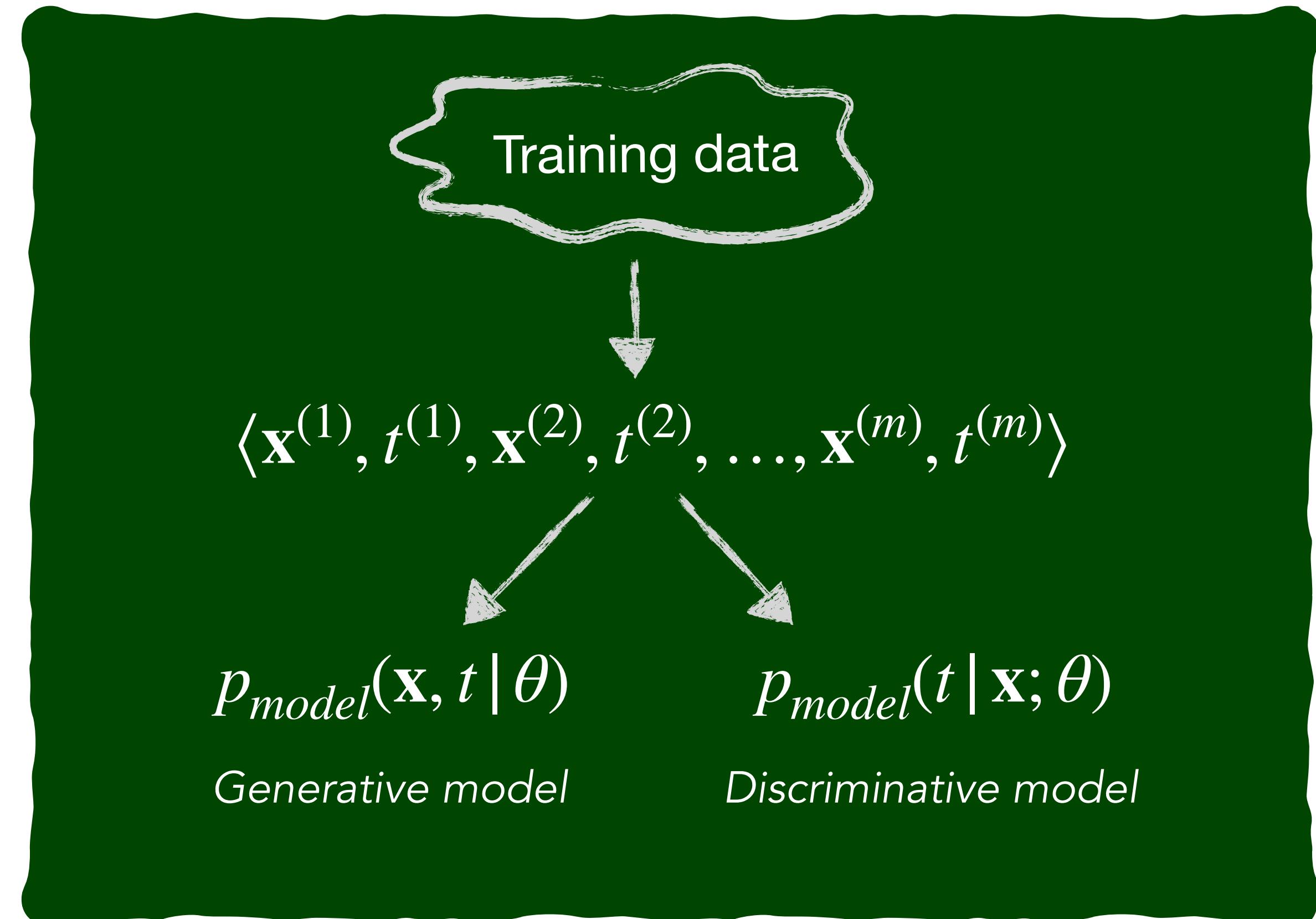
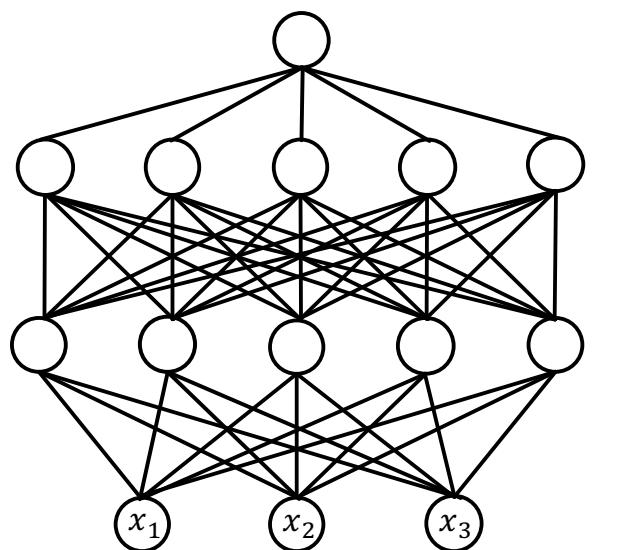
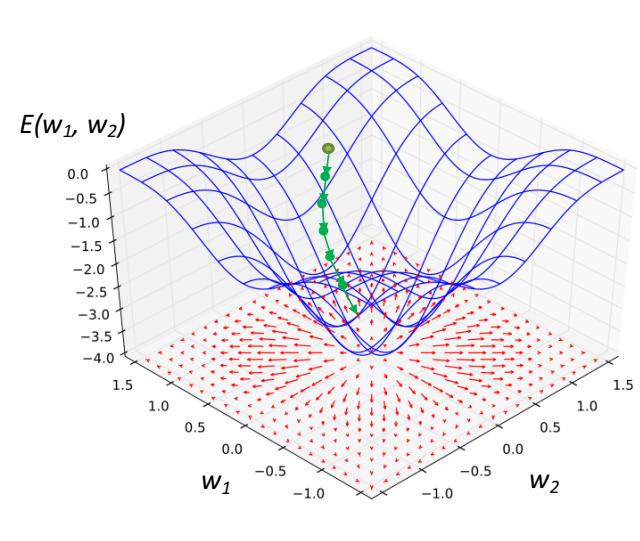
## Data-generating distribution:

- We assume that training and test-data are generated from the *data-generating distribution*  $p_{data}$ .
- Our goal is to achieve minimal generalization error (i.e., error on randomly drawn samples from this distribution).



# Why do we talk about “estimators”?

- We adopt a **probabilistic perspective** where the neural network models some aspect of the data generating distribution.
- Assume a data generating distribution over inputs and targets  $p_{data}(\mathbf{x}, t)$ .
- Our neural network is a probabilistic model, e.g.,  $p_{model}(t | \mathbf{x}; \theta)$  for a classification problem, where  $\theta$  are the network weights.
- Training is the estimation of the parameters  $\theta$ .



# Point estimation

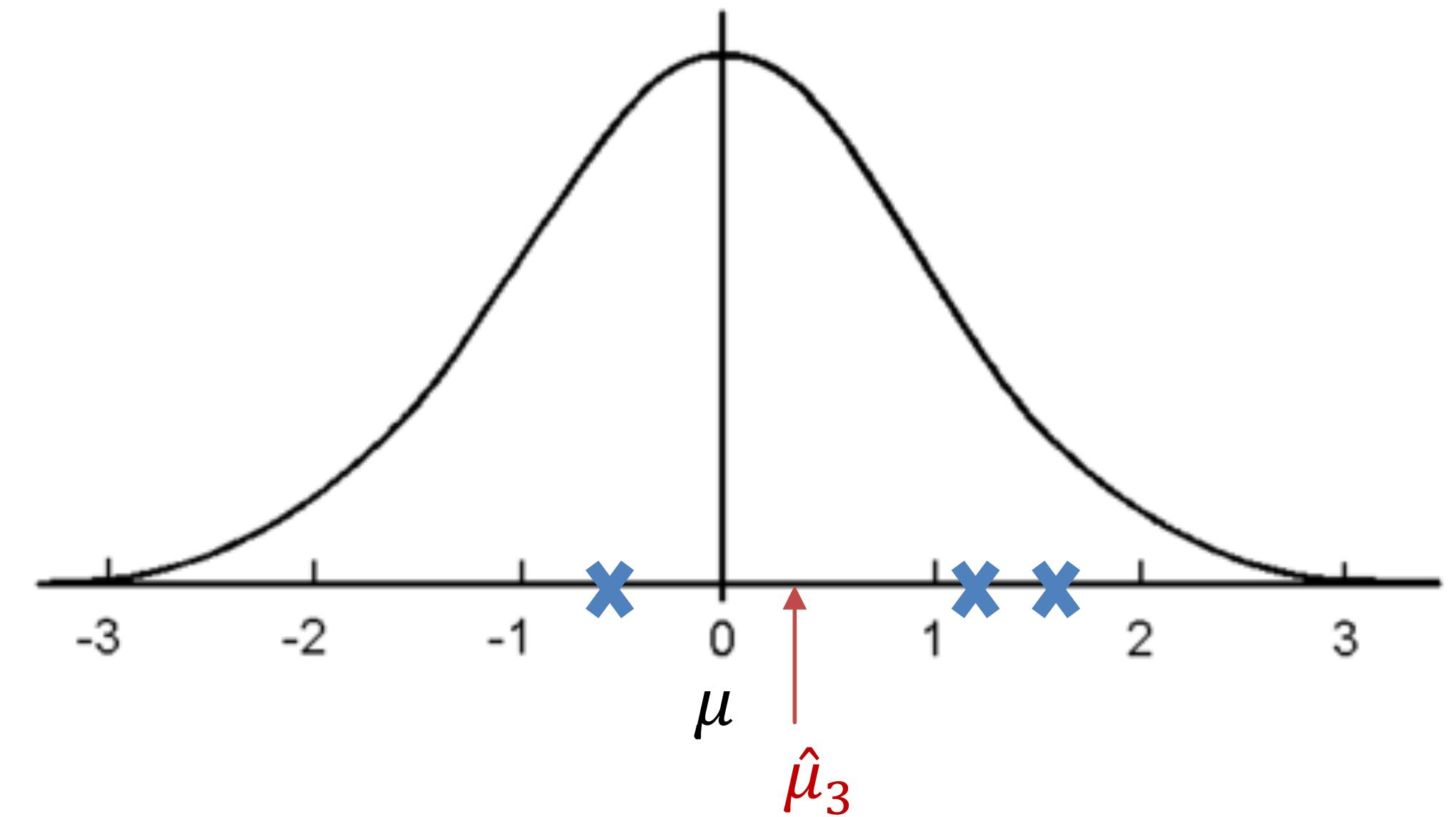
**Point estimation** is the attempt to provide the single “best” prediction of some quantity of interest.

- We assume a parameterized distribution  $p(\mathbf{x}; \theta)$  and want to estimate the parametric vector  $\theta$ .
- We have  $m$  i.i.d. samples  $\langle \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)} \rangle$ , i.e., data points, from  $p(\mathbf{x}; \theta)$ .
- A **point estimator** or **statistic** is any function of the data:  $\hat{\theta}_m = g(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)})$ .

**Example:** mean of a Gaussian:  $p(\mathbf{x}; \theta) = N(x; \mu, \sigma^2)$

- We want an estimate  $\hat{\mu}_m$  of the mean.
- A possible estimator: sample mean

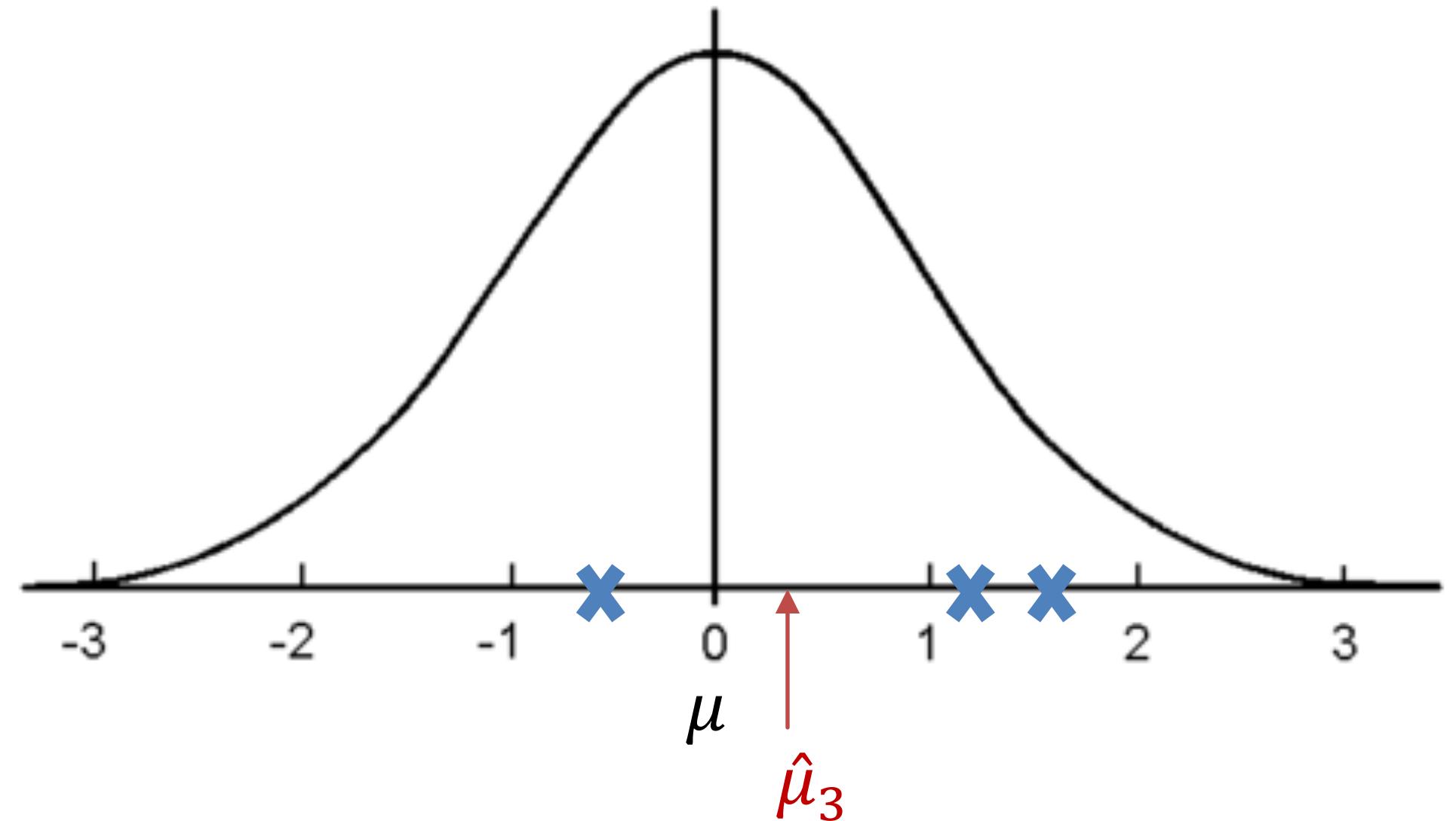
$$\hat{\mu}_m = g(x^{(1)}, \dots, x^{(m)}) = \frac{1}{m} \sum_{i=1}^m x^{(i)}$$



# Bias

**Bias of an estimator:**  $\text{bias}(\hat{\theta}_m) = \mathbb{E}[\hat{\theta}_m] - \theta$ .

- How much, in expectation, the point estimator deviates from the true parameter.
- An estimator  $\hat{\theta}_m$  is unbiased if:  $\text{bias}(\hat{\theta}_m) = 0$ , or equivalently:  $\mathbb{E}[\hat{\theta}_m] = \theta$ .



**Example:** The sample mean:  $\hat{\mu}_m = \frac{1}{m} \sum_{i=1}^m x^{(i)}$  of the Gaussian mean is an *unbiased* estimator.

$$\text{bias}(\hat{\mu}_m) = \mathbb{E}[\hat{\mu}_m] - \mu = \mathbb{E} \left[ \frac{1}{m} \sum_{i=1}^m x^{(i)} \right] - \mu = \left( \frac{1}{m} \sum_{i=1}^m \mathbb{E}[x^{(i)}] \right) - \mu = \left( \frac{1}{m} \sum_{i=1}^m \mu \right) - \mu = 0$$

# Variance

**Variance of an estimator:**  $\text{Var}(\hat{\theta}_m) = \mathbb{E}[(\hat{\theta}_m - \mathbb{E}[\hat{\theta}_m])^2]$ .

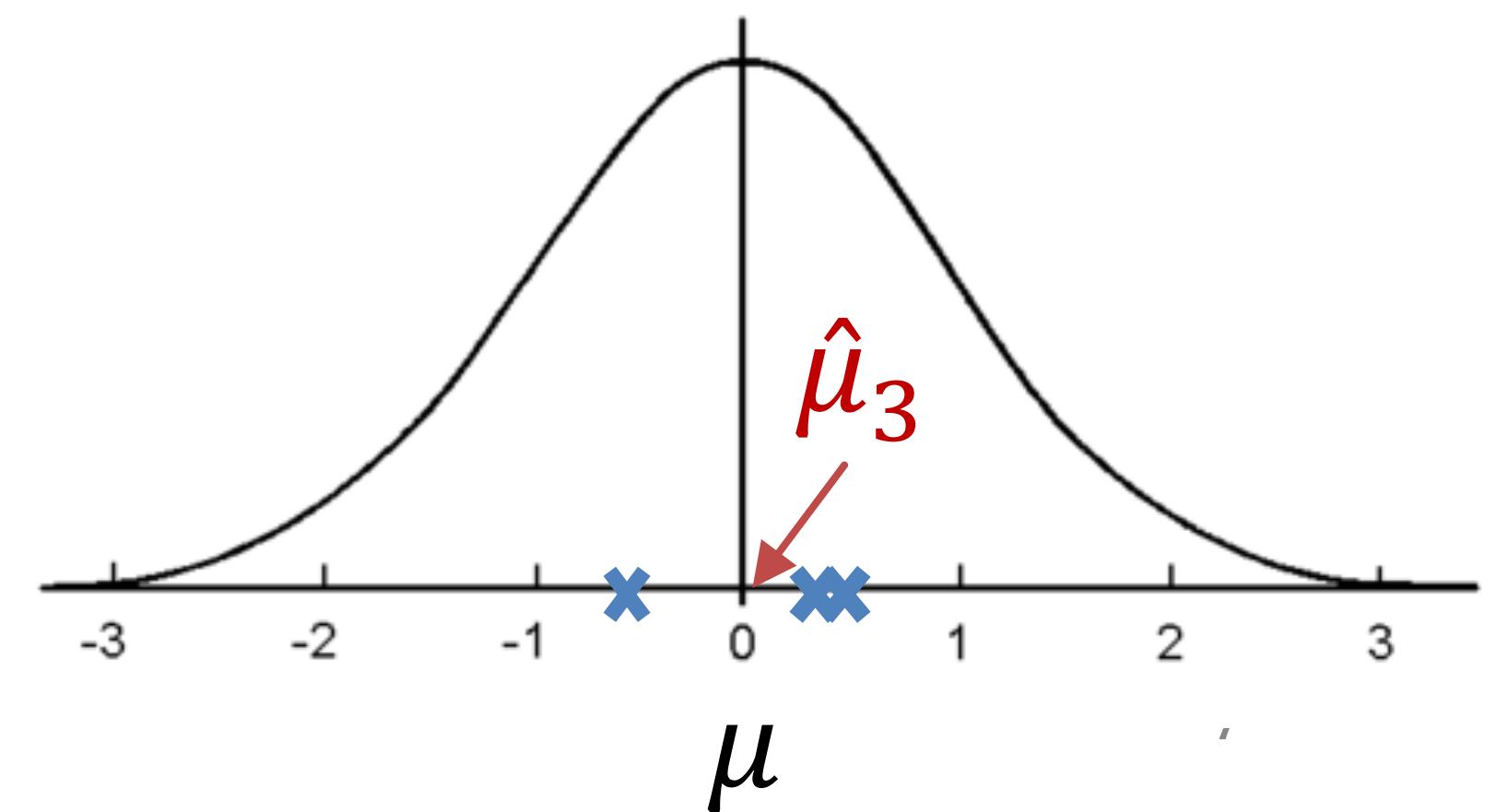
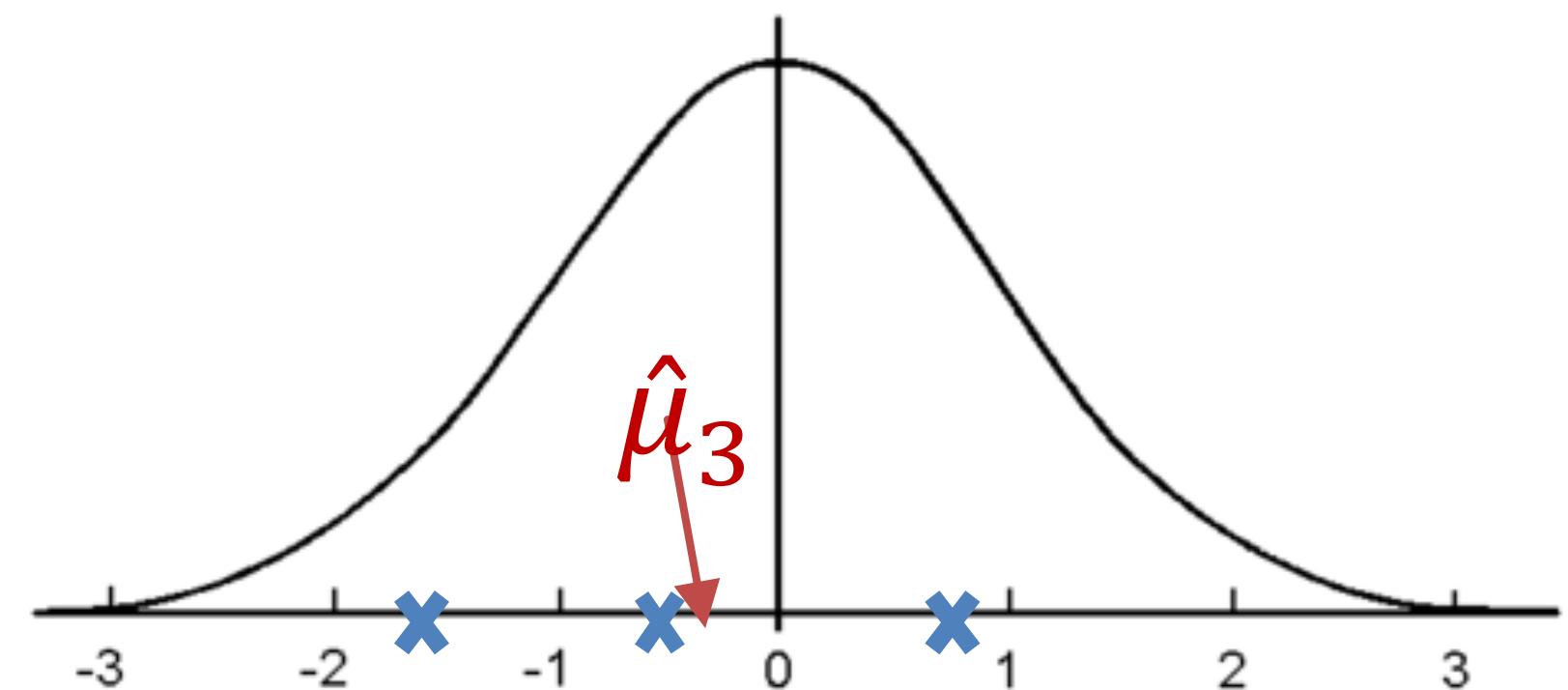
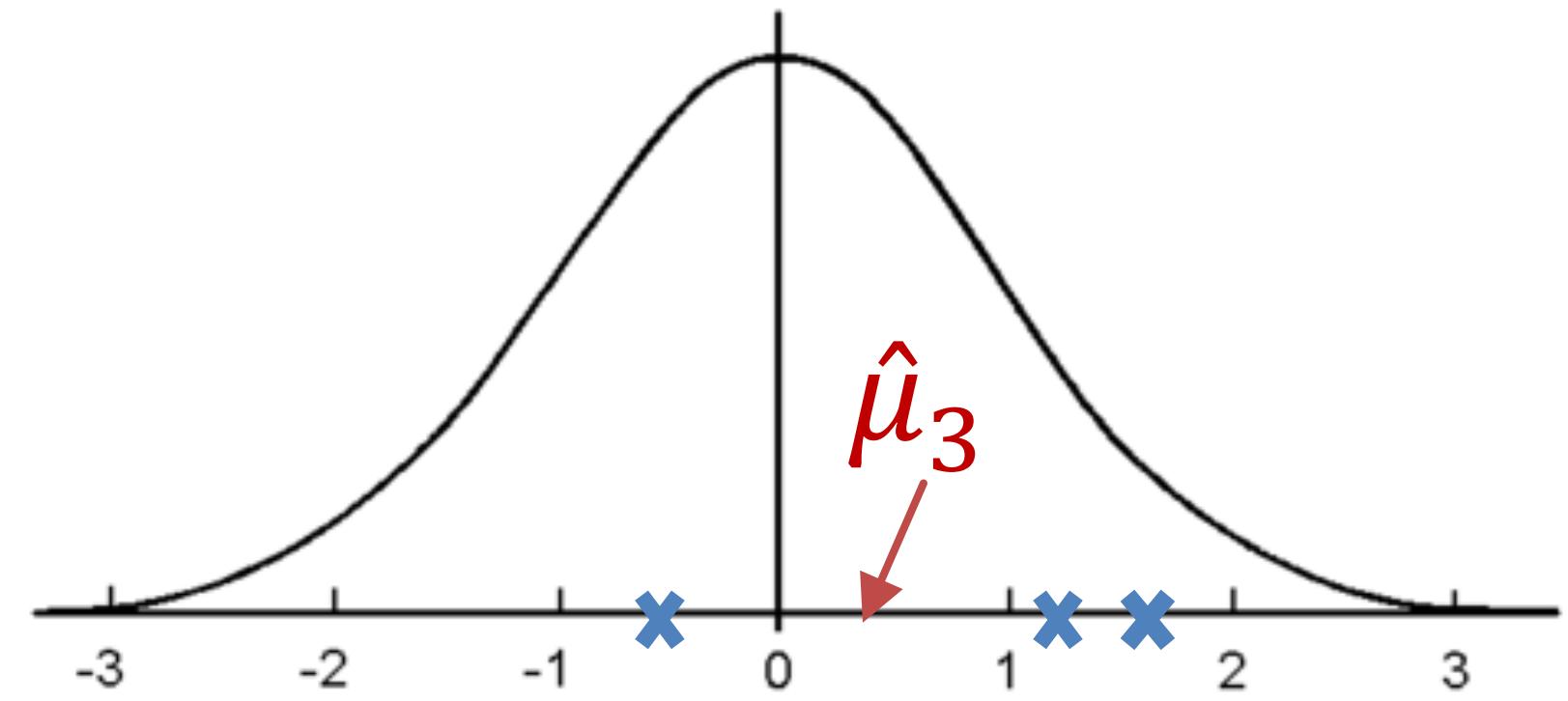
- How strongly does the estimator vary for different drawings of the data set.

**Standard error of an estimator:**  $\text{SE}(\hat{\theta}_m) = \sqrt{\text{Var}(\hat{\theta}_m)}$ .

**Example:** For a Gaussian with true variance  $\sigma^2$ , the standard error of the sample mean estimator is:

$$\text{SE}(\hat{\mu}_m) = \sqrt{\text{Var}\left[\frac{1}{m} \sum_{i=1}^m x^{(i)}\right]} = \frac{\sigma}{\sqrt{m}}$$

What does this imply?



# Consistency

---

- Behavior of the estimator as the amount of data grows.

An estimator is (weakly) consistent if:

$$\text{plim}_{m \rightarrow \infty} \hat{\theta}_m = \theta$$

i.e., (convergence in probability):  $\forall \epsilon > 0 : P(|\hat{\theta}_m - \theta| > \epsilon) \rightarrow 0 \text{ as } m \rightarrow \infty .$

This ensures that the bias diminishes (i.e., estimate gets closer to the true value of the parameter) as the number of data grows.

- An unbiased estimator is not necessarily consistent.
- A consistent estimator is not necessarily unbiased.

# Today

---

Estimators

Maximum Likelihood & Maximum A Posteriori Estimation

Classification & Decision Theory

# Maximum Likelihood (ML) Estimation

- We observe some i.i.d. data  $\mathbf{X} = \langle \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)} \rangle$  drawn from  $p_{data}(\mathbf{x})$ .
- Let  $p_{model}(\mathbf{x}; \theta)$  be a parametric family of distributions.
- Maximum likelihood estimate for  $\theta$ :  
$$\begin{aligned}\hat{\theta}_{ML} &= \arg \max_{\theta} p_{model}(\mathbf{X}; \theta) \\ &= \arg \max_{\theta} \prod_{i=1}^m p_{model}(\mathbf{x}^{(i)}; \theta) \\ &= \arg \max_{\theta} \sum_{i=1}^m \log p_{model}(\mathbf{x}^{(i)}; \theta)\end{aligned}$$

The maximum likelihood estimator is **consistent** given that:

- The true distribution  $p_{data}(\mathbf{x})$  lies within the model family  $p_{model}(\mathbf{x}; \theta)$
- The true distribution  $p_{data}(\mathbf{x})$  corresponds to exactly one value of  $\theta$ .

What is the ML estimate  
of the mean of a  
Gaussian, i.e.,  $\hat{\mu}_{ML}$ ?

# Example: Maximum likelihood estimate of the Gaussian mean

$$X = \langle x^{(1)}, x^{(2)}, \dots, x^{(m)} \rangle \quad p_{model}(x | \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)$$
$$x \in \mathbb{R}, \mu \in \mathbb{R}, \sigma > 0$$

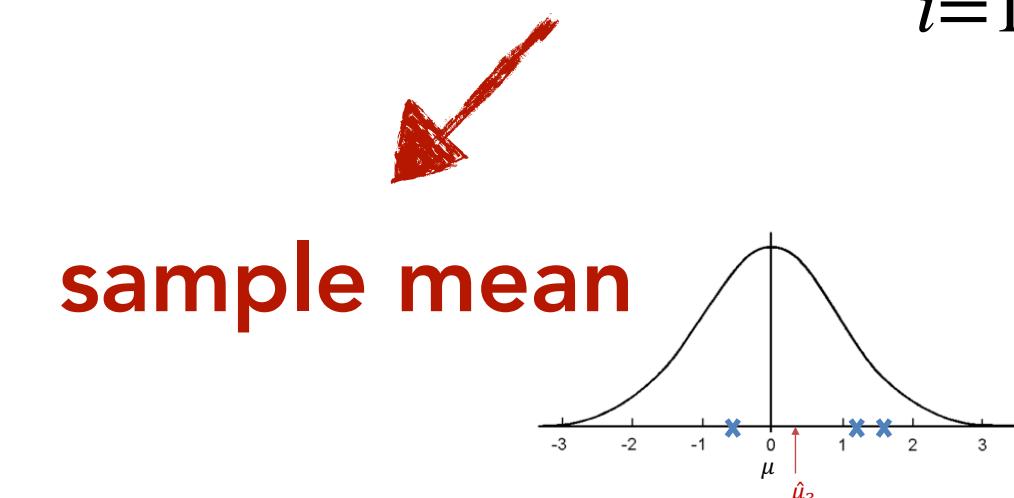
Likelihood

$$p_{model}(X | \mu, \sigma) = \prod_{i=1}^m p_{model}(x^{(i)} | \mu, \sigma)$$
$$\hat{\mu}_{ML} = \arg \max_{\mu} \sum_{i=1}^m \log p_{model}(x^{(i)} | \mu, \sigma)$$
$$= \arg \max_{\mu} \sum_{i=1}^m \left( \log \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{(x^{(i)} - \mu)^2}{2\sigma^2} \right)$$
$$= \arg \min_{\mu} \sum_{i=1}^m (x^{(i)} - \mu)^2$$

Compute the minimum:

$$\frac{\partial}{\partial \mu} \sum_{i=1}^m (x^{(i)} - \mu)^2 = -2 \sum_{i=1}^m (x^{(i)} - \mu)$$
$$-2 \sum_{i=1}^m x^{(i)} + 2m \hat{\mu}_{ML} = 0$$

$$\hat{\mu}_{ML} = \frac{1}{m} \sum_{i=1}^m x^{(i)}$$



# Conditional Log-Likelihood

---

- In regression or classification problems with  $\mathbf{X} = \langle \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)} \rangle$  and targets  $\mathbf{t} = (t^{(1)}, \dots, t^{(m)})^T$  we are usually interested in estimating the conditional distribution  $p_{data}(\mathbf{t} | \mathbf{X})$ .
- We can also use the maximum likelihood estimation principle:

$$\theta_{ML} = \arg \max_{\theta} p_{model}(\mathbf{t} | \mathbf{X}; \theta)$$

$$\theta_{ML} = \arg \max_{\theta} \sum_{i=1}^m \log p_{model}(t^{(i)} | \mathbf{x}^{(i)}; \theta)$$

# Example: Regression with Gaussian noise

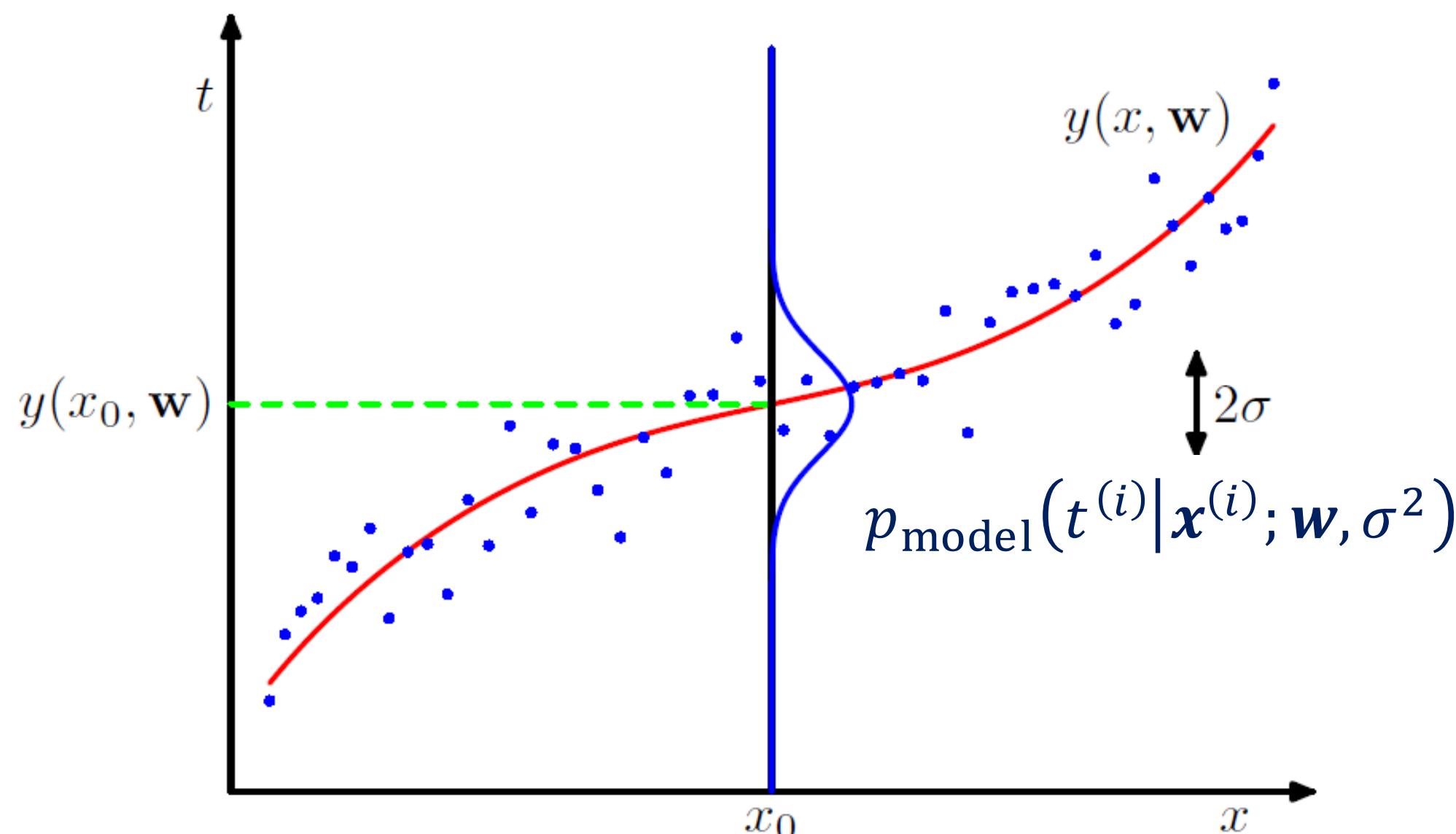
Given training examples with  $\langle \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)} \rangle$  and  $\mathbf{t} = (t^{(1)}, \dots, t^{(m)})^T$ .

Consider a probabilistic model for the data:

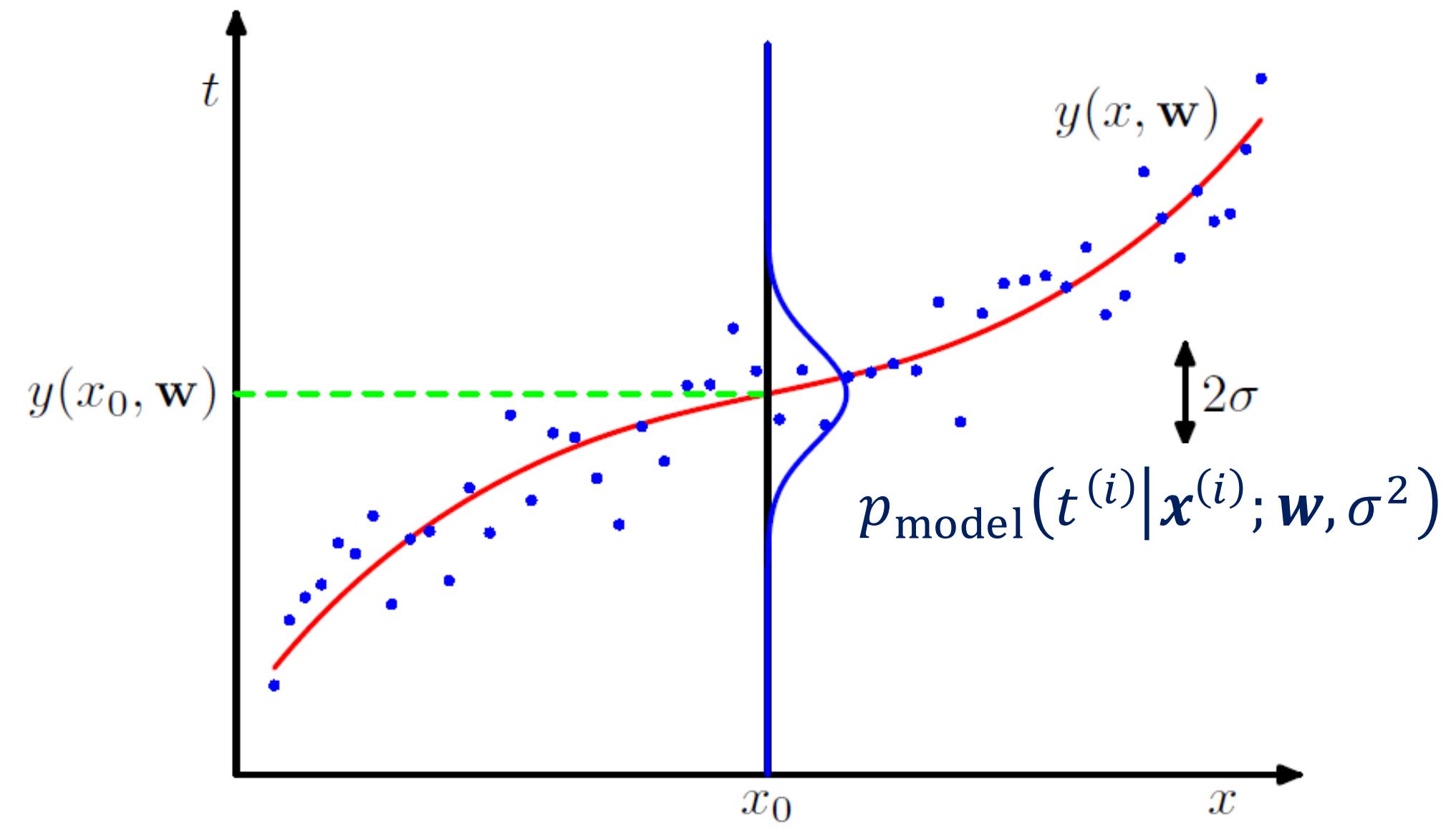
$$p_{model}(t^{(i)} | \mathbf{x}^{(i)}; \mathbf{w}, \sigma^2) = N(t^{(i)}; y_w(\mathbf{x}^{(i)}), \sigma^2).$$

$N(t; \mu, \sigma^2)$   $\longrightarrow$  Normal distribution with mean  $\mu$  and variance  $\sigma^2$

$y_w(\mathbf{x}^{(i)})$   $\longrightarrow$  A parameterized model for the mean (e.g., a polynomial)



# Example: Regression with Gaussian noise



$$\mathbf{w}_{ML} = \arg \max_{\mathbf{w}} \sum_{i=1}^m \log p_{model}(t^{(i)} | \mathbf{x}^{(i)}; \mathbf{w}, \sigma^2)$$

$$= \arg \max_{\mathbf{w}} -m \log (\sigma \sqrt{2\pi}) - \sum_{i=1}^m \frac{(t^{(i)} - y_w(\mathbf{x}^{(i)}))^2}{2\sigma^2}$$

$$= \arg \min_{\mathbf{w}} \sum_{i=1}^m (y_w(\mathbf{x}^{(i)}) - t^{(i)})^2 \rightarrow \text{Sum squared error}$$

*Estimating the parameters of the model with **maximum likelihood**.*

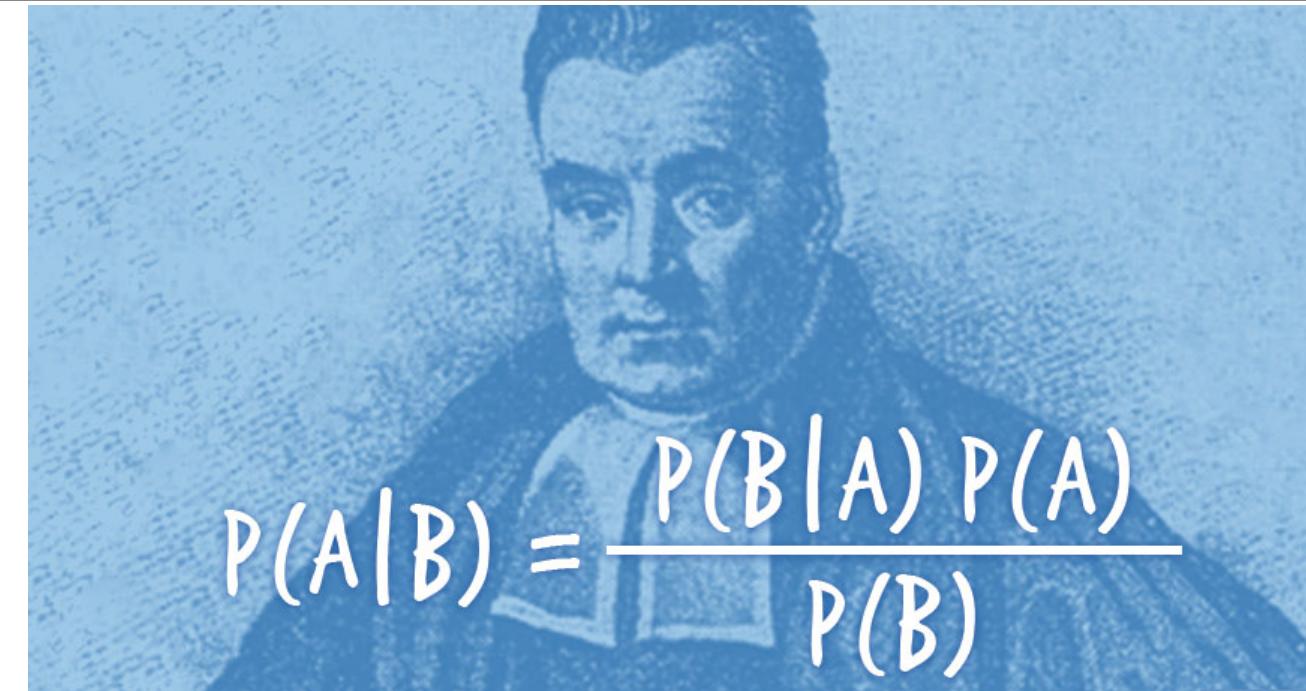
$$N(t^{(i)}; y_w(\mathbf{x}^{(i)}), \sigma^2) = \frac{1}{\sigma \sqrt{2\pi}} \exp \left( -\frac{(t^{(i)} - y_w(\mathbf{x}^{(i)}))^2}{2\sigma^2} \right)$$

# Maximum A Posteriori (MAP) Estimation

**Bayes' Rule:**  $p(\theta | \mathbf{X}) \propto p(\mathbf{X} | \theta) p(\theta)$

The MAP estimator uses the point of maximum posterior probability:

$$\begin{aligned}\theta_{MAP} &= \arg \max_{\theta} p_{model}(\theta | \mathbf{X}) \\ &= \arg \max_{\theta} [\log p_{model}(\mathbf{X} | \theta) + \log p(\theta)] \\ &= \arg \max_{\theta} \left[ \sum_{i=1}^m \log p_{model}(\mathbf{x}^{(i)} | \theta) + \log p(\theta) \right]\end{aligned}$$



$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$

$$p(\theta | \mathbf{X}) = \frac{p(\mathbf{X} | \theta)p(\theta)}{p(\mathbf{X})}$$

The MAP estimator can **decrease the variance** at the cost of **increasing bias**.

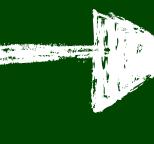
# Example: Regression with Gaussian noise and Gaussian Prior

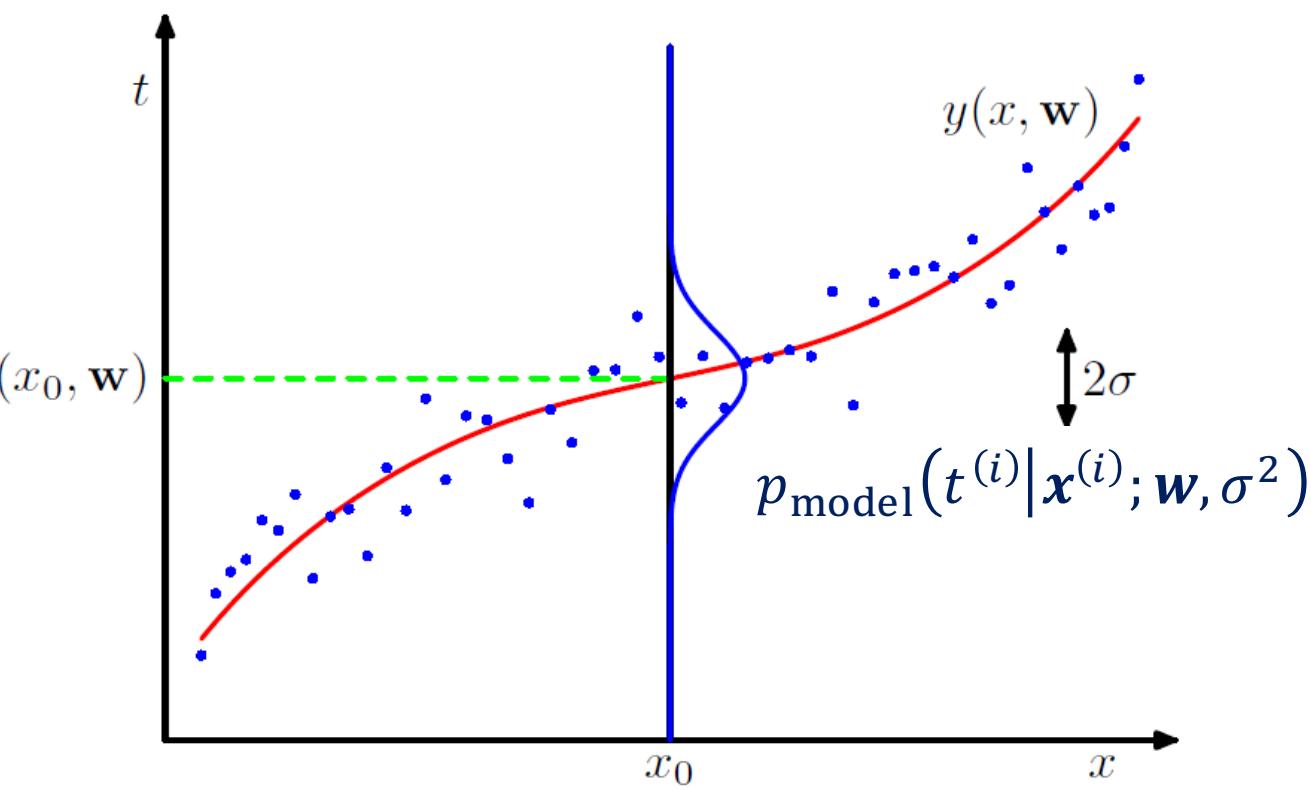
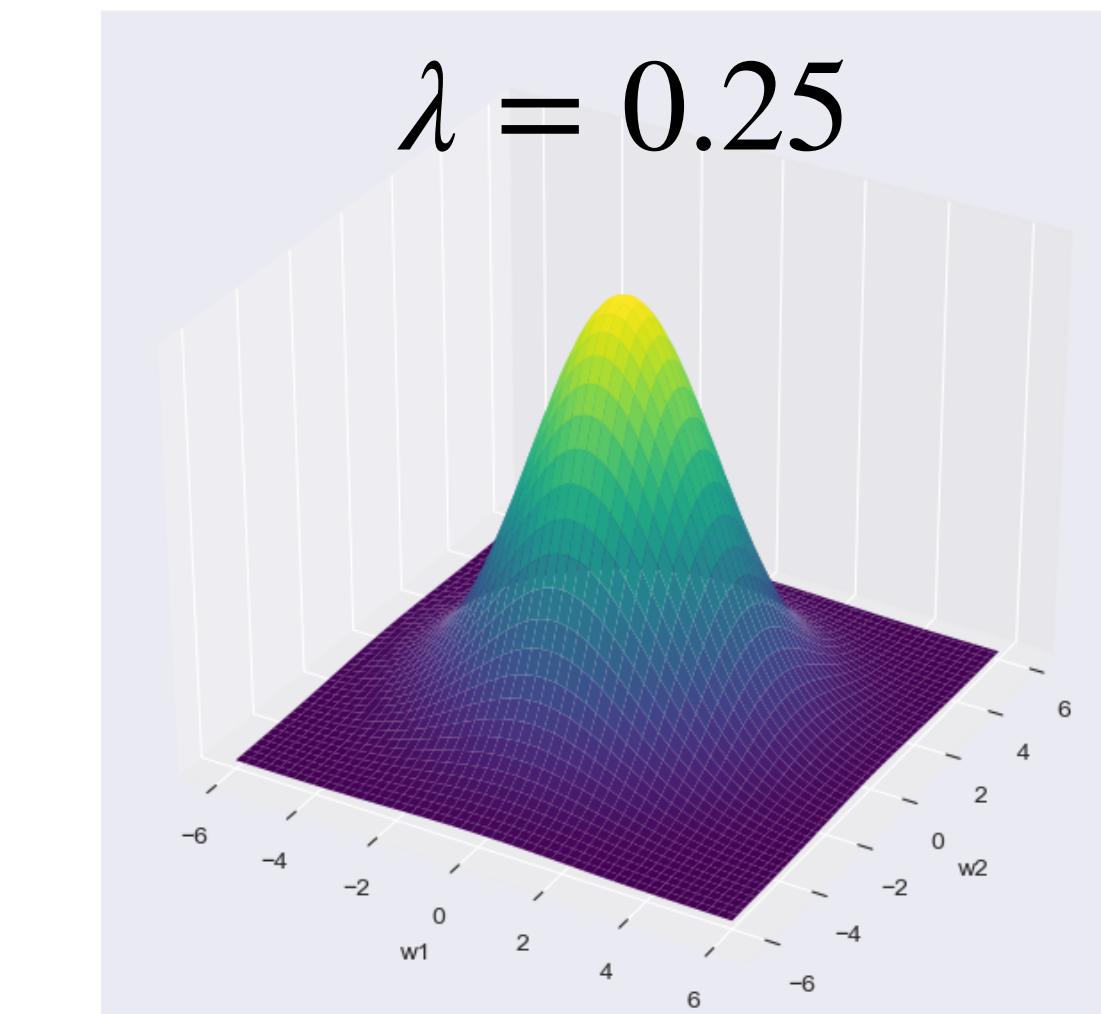
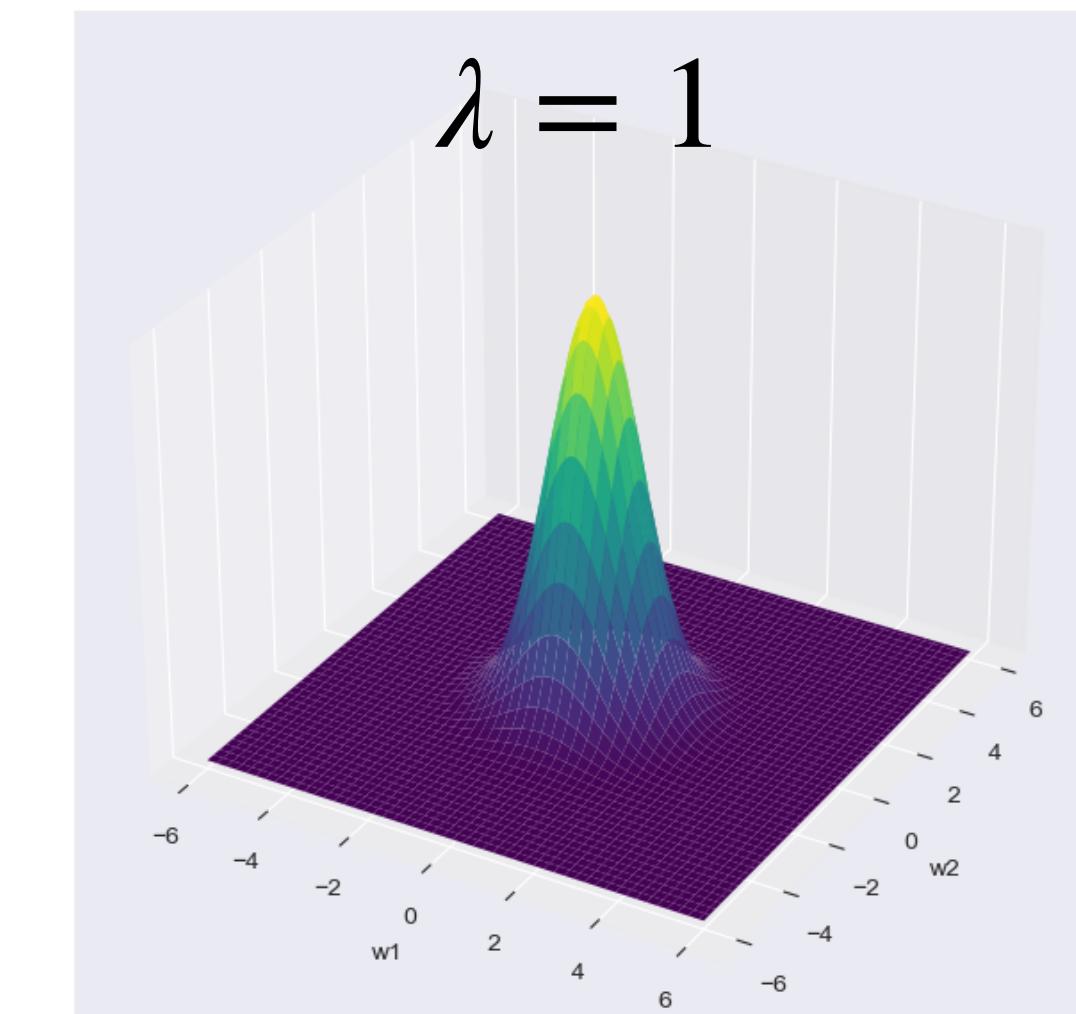
Consider a probabilistic model for the data:

$$p_{model}(t^{(i)} | \mathbf{x}^{(i)}; \mathbf{w}, \sigma^2) = N(t^{(i)}; y_w(\mathbf{x}^{(i)}), \sigma^2)$$

We assume a Gaussian prior on the parameters:

$$p(\mathbf{w}) = N(\mathbf{w}; 0, \frac{1}{\lambda} \mathbf{I}) = \left( \frac{\lambda}{2\pi} \right)^{\frac{D}{2}} \exp \left( -\frac{\lambda}{2} \mathbf{w}^T \mathbf{w} \right)$$

higher  $\lambda$   smaller parameters



# Example: Regression with Gaussian noise and Gaussian Prior

Consider a probabilistic model for the data:

$$p_{model}(t^{(i)} | \mathbf{x}^{(i)}; \mathbf{w}, \sigma^2) = N(t^{(i)}; y_w(\mathbf{x}^{(i)}), \sigma^2)$$

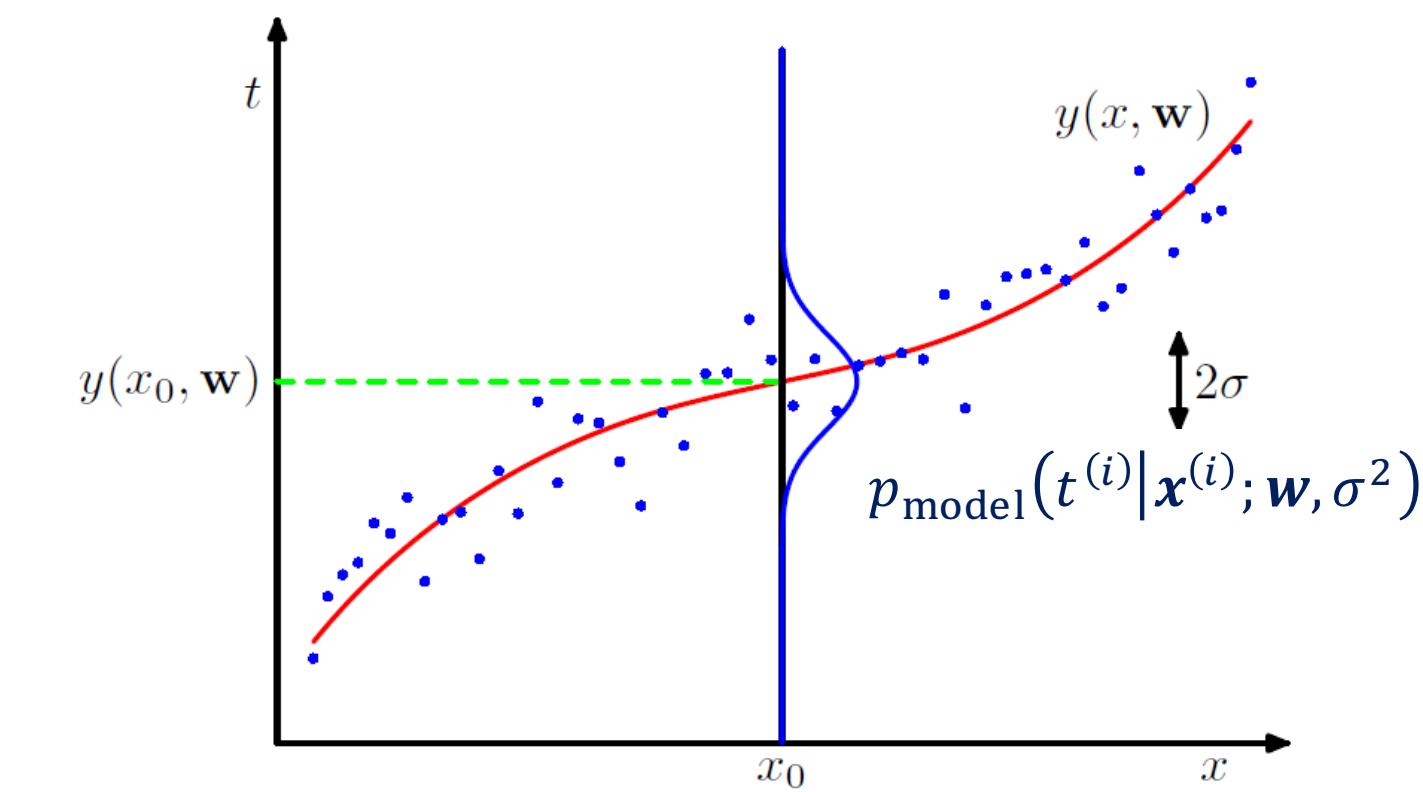
We assume a Gaussian prior on the parameters:

$$p(\mathbf{w}) = N(\mathbf{w}; 0, \frac{1}{\lambda} \mathbf{I}) = \left( \frac{\lambda}{2\pi} \right)^{\frac{D}{2}} \exp \left( -\frac{\lambda}{2} \mathbf{w}^T \mathbf{w} \right)$$

We obtain the MAP estimator:

$$\mathbf{w}_{MAP} = \arg \max_{\mathbf{w}} [\log p_{model}(\mathbf{t} | \mathbf{X}, \mathbf{w}) + \log p(\mathbf{w})]$$

$$= \arg \min_{\mathbf{w}} \left[ \sum_{i=1}^m (y_w(\mathbf{x}^{(i)}) - t^{(i)})^2 + \lambda \sigma^2 \mathbf{w}^T \mathbf{w} \right] \rightarrow \text{Regularization}$$



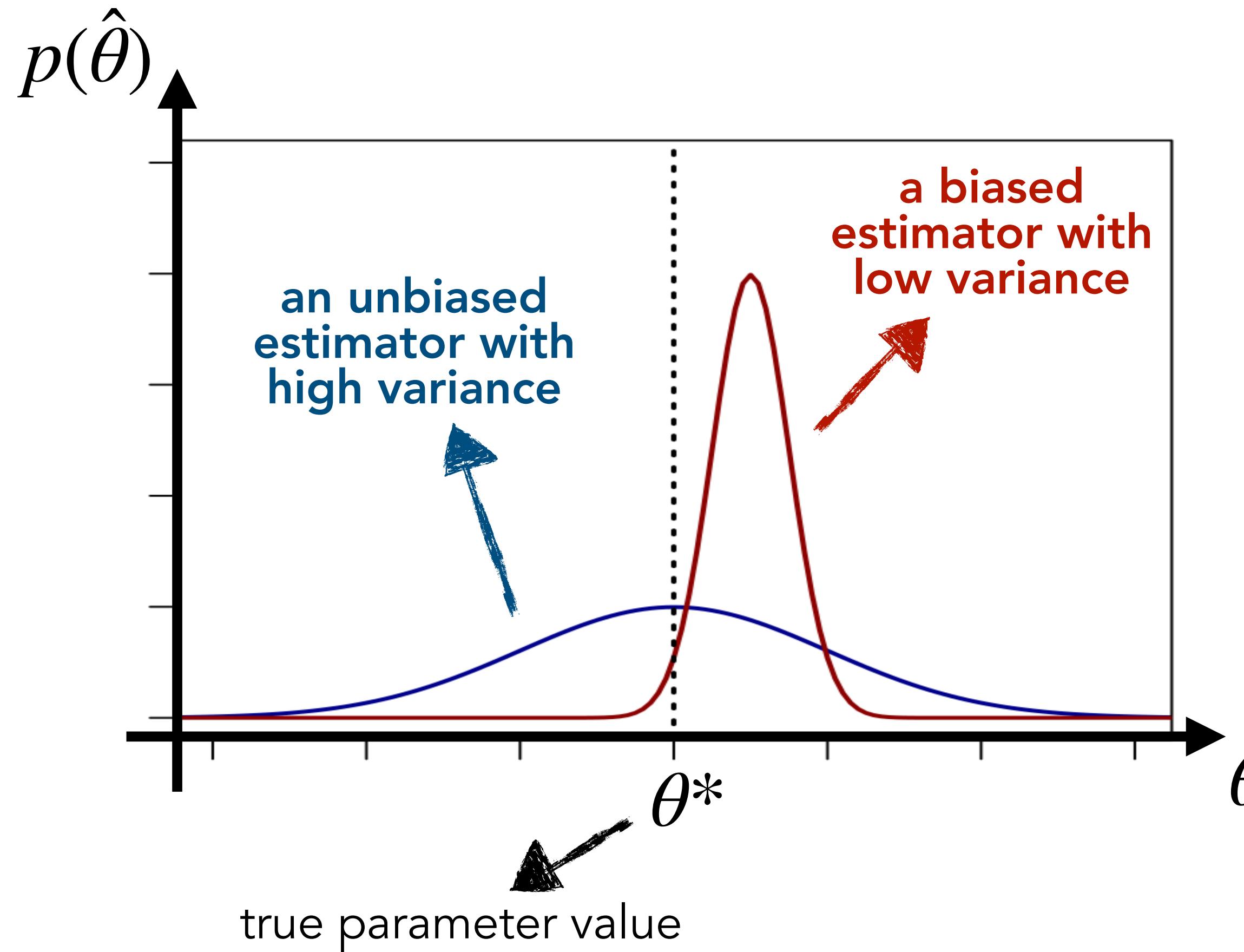
higher  $\lambda \rightarrow$  smaller parameters

Note:  $\mathbf{w}^T \mathbf{w} = \sum_i w_i^2 = ||\mathbf{w}||^2$

# Bias-Variance trade-off

**Bias:** measures the deviation from the true value of the parameter.

**Variance:** measures deviation from the expected estimator value for some data set.



(For each drawing of the data samples, the estimate  $\hat{\theta}$  will vary according to our model.)

# Bias-Variance trade-off

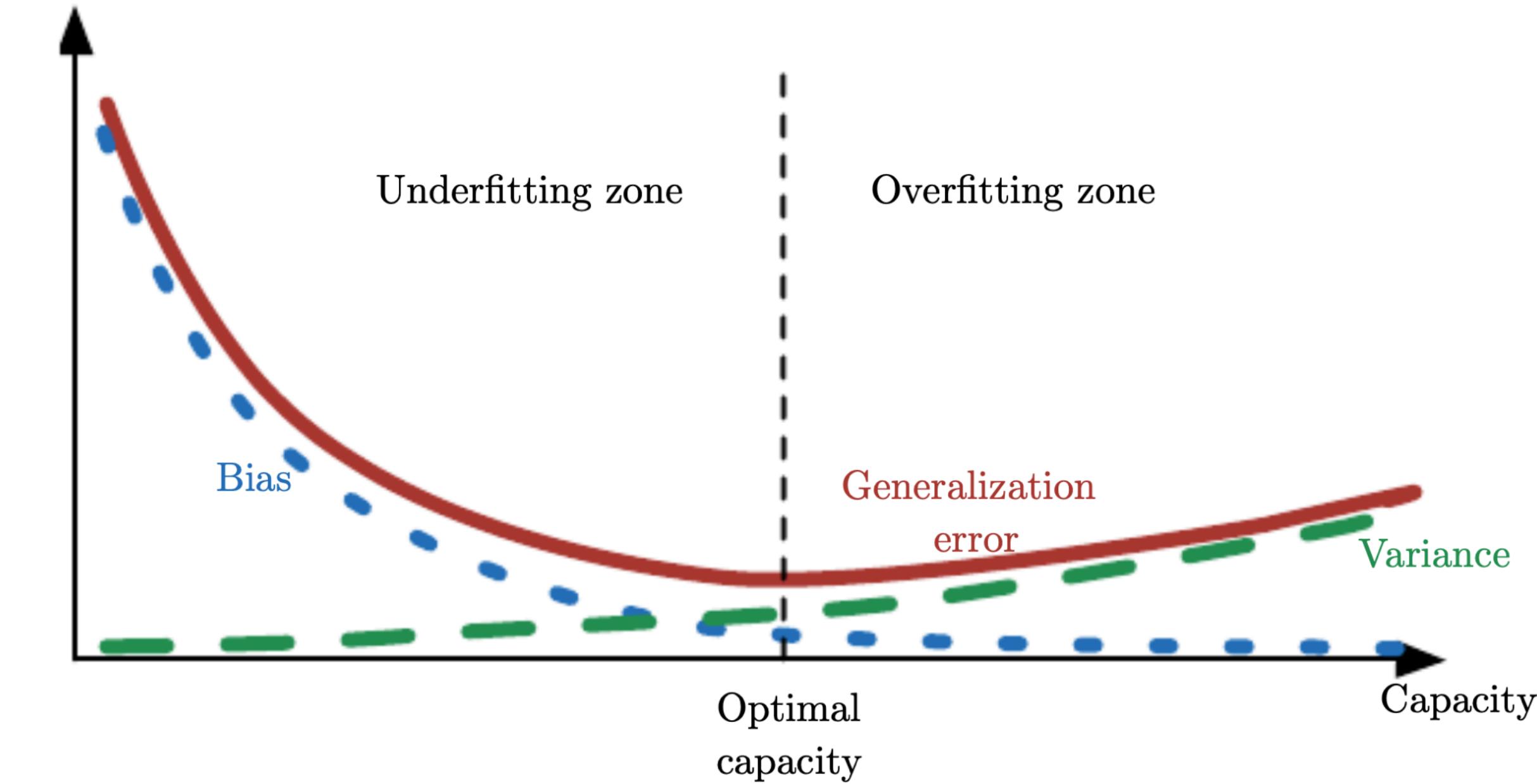
**Bias:** measures the deviation from the true value of the parameter.

**Variance:** measures deviation from the expected estimator value for some data set.

Mean squared error (MSE) of the estimate:

$$\begin{aligned} MSE &= \mathbb{E} \left[ (\hat{\theta}_m - \theta)^2 \right] \\ &= Bias \left( \hat{\theta}_m \right)^2 + Var \left( \hat{\theta}_m \right) \end{aligned}$$

Desirable estimators have small MSE, i.e.,  
estimators with **small bias and small variance**.



# Today

---

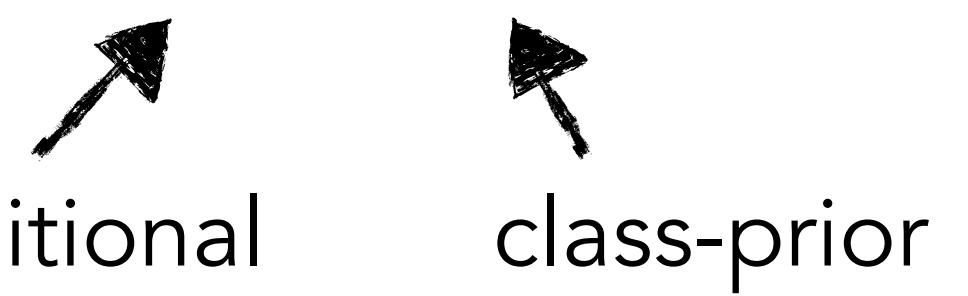
Estimators

Maximum Likelihood & Maximum A Posteriori Estimation

Classification & Decision Theory

# Classification Problems

- We have  $K$  classes  $C_1, \dots, C_K$ , and the random variable  $C$  indicates the class.
- We get an input  $\mathbf{x} \in \mathbb{R}^N$  from the data distribution:  $P(\mathbf{x}, C) = P(\mathbf{x} | C) P(C)$ .



class-conditional      class-prior

**Inference:** Compute the **posterior**  $P(C | \mathbf{x})$ , where  $P(C_k | \mathbf{x})$  is the probability that  $\mathbf{x}$  belongs to class  $C_k$ .

**Decision:** Decide for one class in some optimal way.

**Decision rule:**  $y : \mathbb{R}^N \rightarrow \{1, \dots, K\}$

# Classification Problems

- We have  $K$  classes  $C_1, \dots, C_K$ , and the random variable  $C$  indicates the class.
- We get an input  $\mathbf{x} \in \mathbb{R}^N$  from the data distribution:  $P(\mathbf{x}, C) = P(\mathbf{x} | C) P(C)$ .

↑  
class-conditional      ↑  
class-prior

**Inference:** Compute the **posterior**  $P(C | \mathbf{x})$ , where

$P(C_k | \mathbf{x})$  is the probability that  $\mathbf{x}$  belongs to class  $C_k$ .

**Decision:** Decide for one class in some optimal way.

**Decision rule:**  $y : \mathbb{R}^N \rightarrow \{1, \dots, K\}$



$C_1$        $C_2$

Probability of observing a cat.       $\leftarrow P(cat)$   
Distribution over images of cats.       $\leftarrow P(\mathbf{x} | cat)$   
Probability that a given image is a cat.       $\leftarrow P(cat | \mathbf{x})$

$$P(cat) = 0.8 \quad \sum_{k=1}^K P(C_k) = 1$$
$$P(dog) = 0.2$$

# Inference

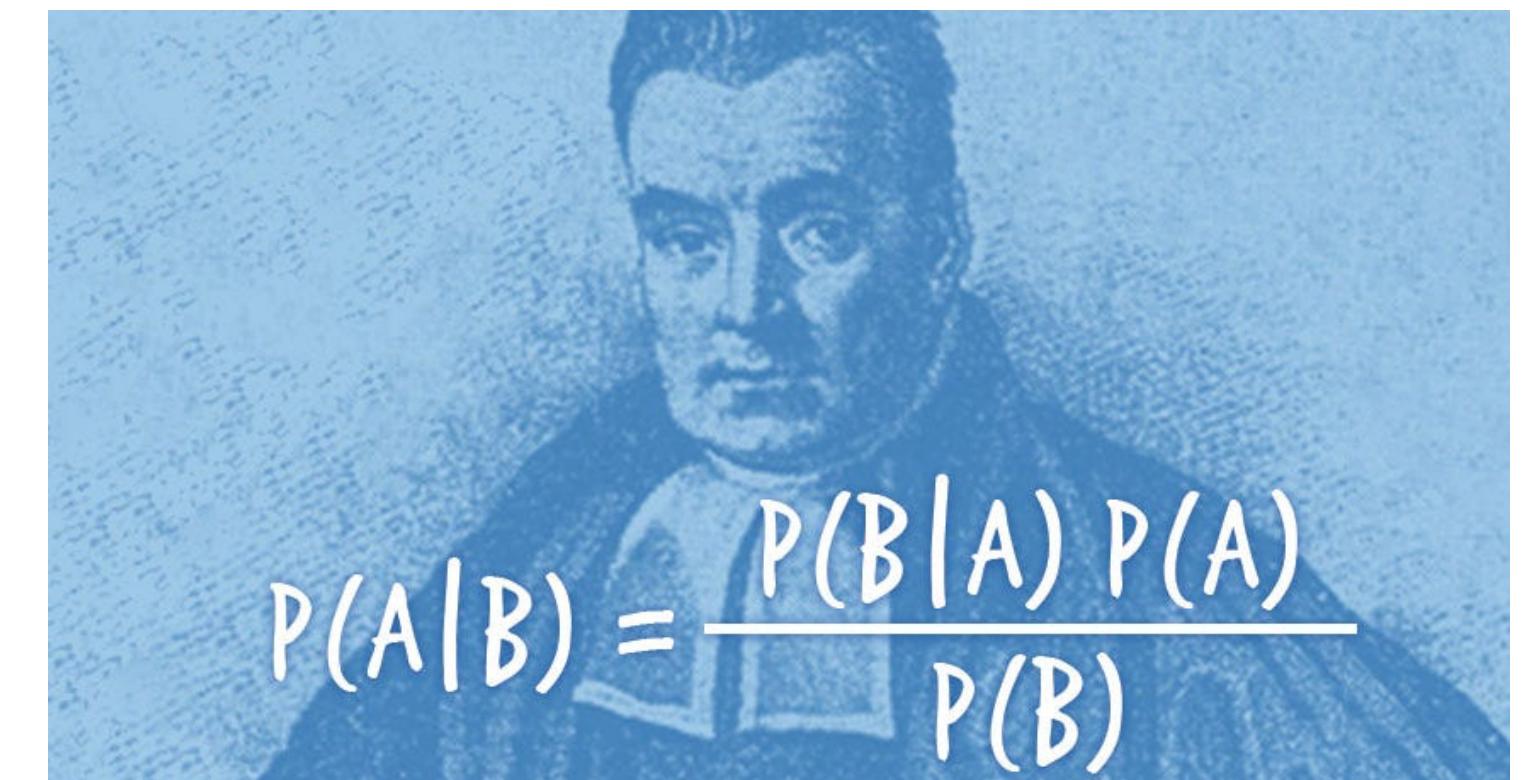
- We have  $K$  classes  $C_1, \dots, C_K$ , and the random variable  $C$  indicates the class.
- We get an input  $\mathbf{x} \in \mathbb{R}^N$  from the data distribution:  $P(\mathbf{x}, C) = P(\mathbf{x} | C) P(C)$ .
- The class  $C$  is however not observed.

**Inference:** Compute the **posterior**  $P(C | \mathbf{x})$ , where  $P(C_k | \mathbf{x})$  is the probability that  $\mathbf{x}$  belongs to class  $C_k$ .

$$P(C_k | \mathbf{x}) = \frac{P(\mathbf{x}, C_k)}{P(\mathbf{x})} = \frac{P(\mathbf{x}, C_k)}{\sum_j P(\mathbf{x}, C_j)}$$

or

$$P(C_k | \mathbf{x}) = \frac{P(\mathbf{x} | C_k) P(C_k)}{P(\mathbf{x})} = \frac{P(\mathbf{x} | C_k) P(C_k)}{\sum_j P(\mathbf{x} | C_j) P(C_j)}$$



$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$

# Modeling the distribution

Given examples  $(\mathbf{x}^{(m)}, C^{(m)})$  from the data distribution.

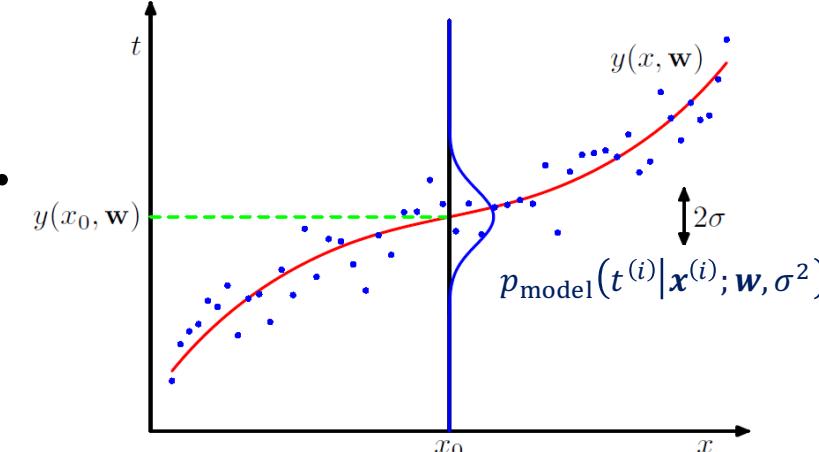
## Generative model

- We fit a model distribution  $P_{model}(\mathbf{x}, C)$ .
- We can then obtain the posterior distribution by:

$$P_{model}(C_k | \mathbf{x}) = \frac{P_{model}(\mathbf{x}, C_k)}{P_{model}(\mathbf{x})}.$$

## Discriminative model

- We fit directly a model for the posterior distribution  $P_{model}(C | \mathbf{x})$ .
- Recall: The conditional log-likelihood example.



# Cross-Entropy Error Function

- We want to estimate the parameters of **a discriminative model**.
- Given training examples with  $\mathbf{X} = \langle \mathbf{x}^{(1)}, \dots, \mathbf{x}^{(M)} \rangle$  and  $\mathbf{t} = (t^{(1)}, \dots, t^{(M)})^T$ , where  $t^{(m)} \in \{0,1\}$  and **two classes**:  $C_1$  ( $t^{(m)} = 1$ ) and  $C_2$  ( $t^{(m)} = 0$ ).
- The output  $y_m$  of the model for input  $\mathbf{x}^{(m)}$  is interpreted as the posterior for class  $C_1$ :

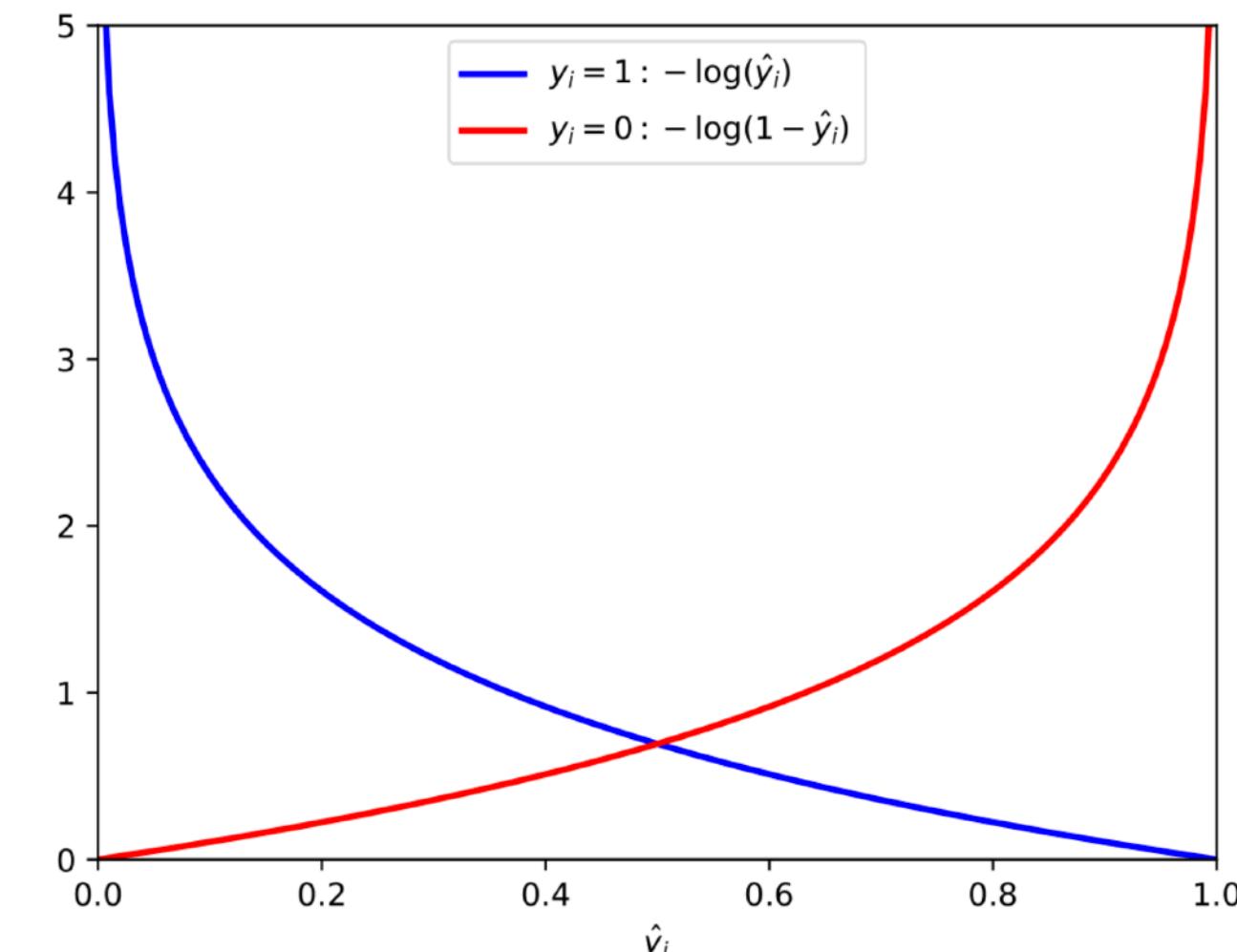
$$P(C_1 | \mathbf{x}^{(m)}) = y_m \quad (\text{and thus } P(C_2 | \mathbf{x}^{(m)}) = 1 - y_m).$$

- **We are using ML to estimate the parameters:**

$$P(\mathbf{t} | \mathbf{X}, \theta) = \prod_{m=1}^M P(t^{(m)} | \mathbf{x}^{(m)}, \theta) = \prod_{m=1}^M (y_m)^{t^{(m)}} (1 - y_m)^{(1-t^{(m)})}$$

- We minimize the negative log-likelihood:

$$E = -\log p(\mathbf{t} | \mathbf{X}, \theta) = \sum_{m=1}^M \left[ -t^{(m)} \log(y_m) - (1 - t^{(m)}) \log(1 - y_m) \right].$$



Cross-entropy error

# Sigmoid Function

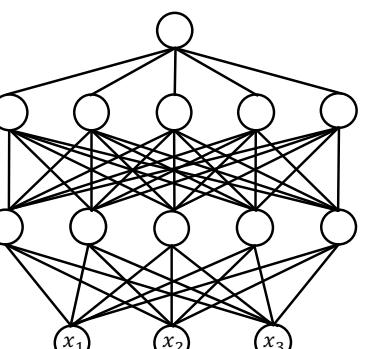
In binary classification:

$$P(C_1 | \mathbf{x}) = \frac{P(\mathbf{x}, C_1)}{P(\mathbf{x})} = \frac{P(\mathbf{x}, C_1)}{P(\mathbf{x}, C_1) + P(\mathbf{x}, C_2)} = \frac{1}{1 + \frac{P(\mathbf{x}, C_2)}{P(\mathbf{x}, C_1)}} = \frac{1}{1 + \exp(-a)} = \sigma(a)$$

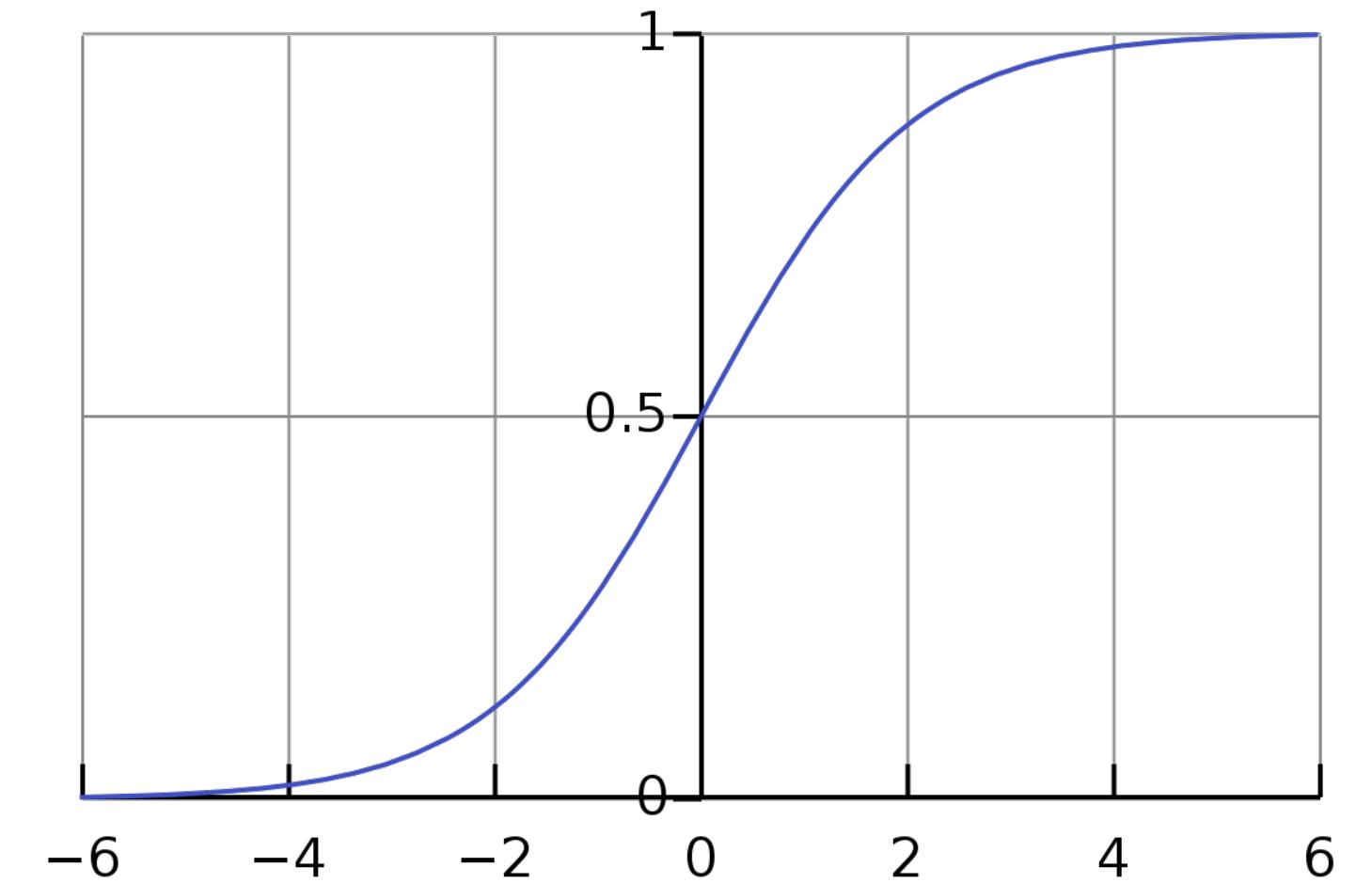
where

$$a = \ln \left( \frac{P(\mathbf{x}, C_1)}{P(\mathbf{x}, C_2)} \right) = \ln \left( \frac{P(\mathbf{x} | C_1)P(C_1)}{P(\mathbf{x} | C_2)P(C_2)} \right)$$

- It is natural to write the posterior as the logistic sigmoid of some quantity.
- Often used in neural networks.
- The logsig function is a saturating nonlinearity that maps  $a \in \mathbb{R}$  to  $(0, 1)$ .



$$\sigma(a) = \frac{1}{1 + \exp(-a)}$$



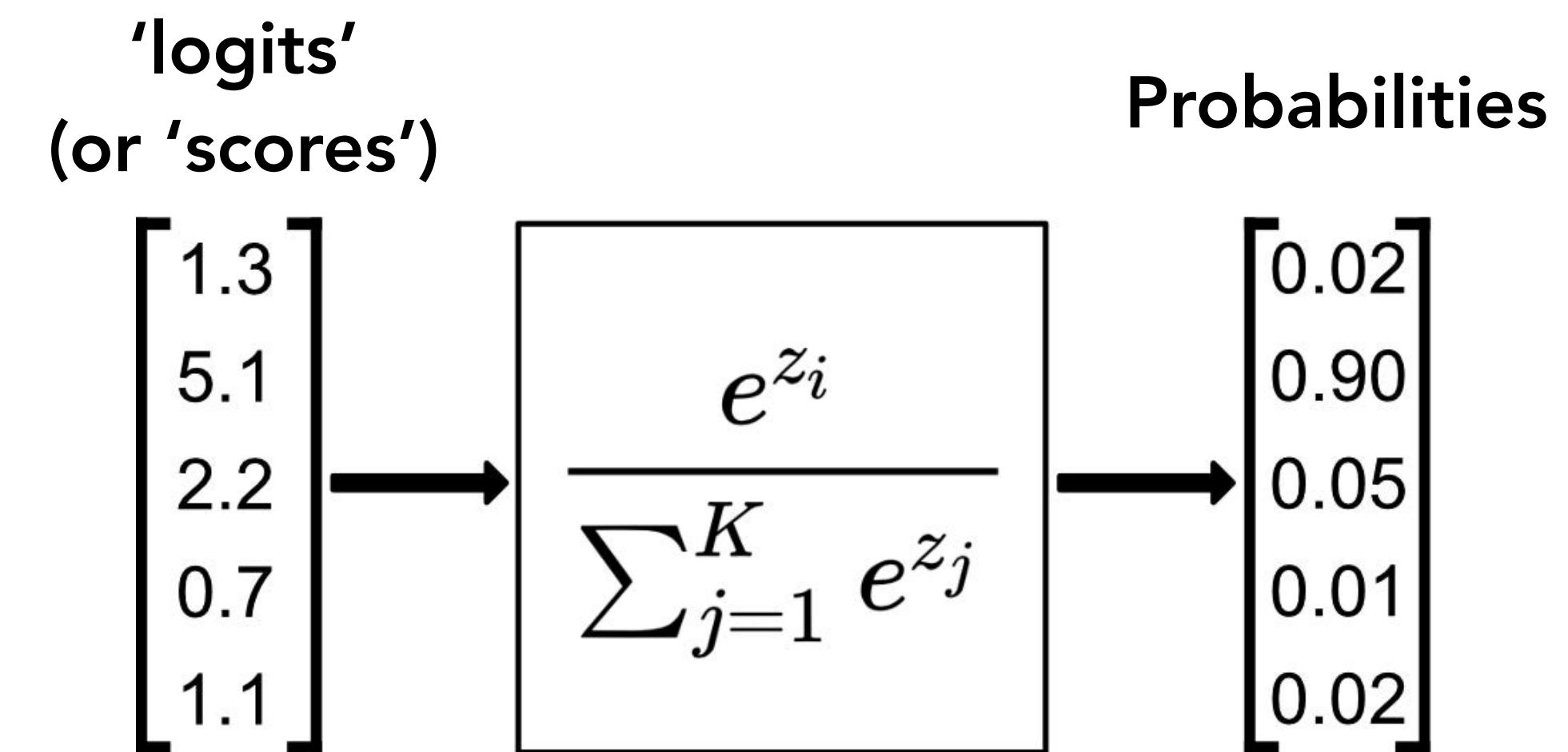
# Softmax Function

In multi-class classification:

$$P(C_k | \mathbf{x}) = \frac{P(\mathbf{x} | C_k)P(C_k)}{\sum_j P(\mathbf{x} | C_j)P(C_j)} = \frac{\exp(a_k)}{\sum_j \exp(a_j)}$$

where

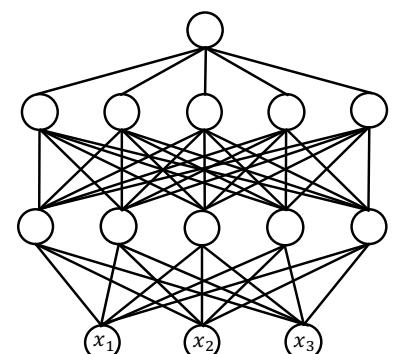
$$a_k = \ln(P(\mathbf{x} | C_k)P(C_k))$$



- The softmax can be interpreted as representing the posterior probabilities for a multi-class classification problem:

$$\sigma_k(a_1, \dots, a_k) = \frac{\exp(a_k)}{\sum_{j=1}^K \exp(a_j)}$$

- Obtains normalized probabilities.
- Often used in neural networks.
- A generalization of the sigmoid function.



# Decision Theory: Minimizing Classification Rate

- Assume we can compute or estimate the posterior  $P(C_k | \mathbf{x})$ .
- How should we decide? What is the best decision rule  $y : \mathbb{R}^N \rightarrow \{1, \dots, K\}$ ?

**Goal 1:** Make as few misclassifications as possible.

$$\mathbb{E}[\text{Correct}] = \int_{\mathbb{R}^N} P(\mathbf{x})P(C_{y(\mathbf{x})} | \mathbf{x})d\mathbf{x}$$

This is maximized for:  $y(\mathbf{x}) = \arg \max_k P(C_k | \mathbf{x})$



Choose the class with  
maximum posterior  
probability

# Decision Theory: Minimizing Expected Loss

- We can introduce a loss matrix  $L$ .

$Loss = L_{k,j}$  if true class is  $C_k$  and  $y(\mathbf{x}) = j$ .

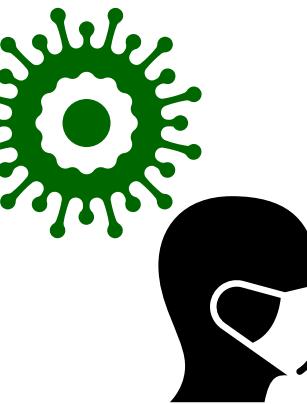
**Goal 2:** Minimize expected loss.

$$\mathbb{E}[\text{Loss}] = \int_{\mathbb{R}^N} P(\mathbf{x}) \sum_k L_{k,y(\mathbf{x})} P(C_k | \mathbf{x}) d\mathbf{x}$$

This is minimized for:  $y(\mathbf{x}) = \arg \min_j \sum_k L_{k,j} P(C_k | \mathbf{x})$

**Not all errors are equal!**

		Predicted class	
		normal	corona
True class	normal	0	1
	corona	100	0



# Today

---

Estimators

Maximum Likelihood & Maximum A Posteriori Estimation

Classification & Decision Theory

# Questions?