Deep Learning:

Estimation & Decision Theory

Robert Legenstein & Ozan Özdenizci
Institute of Theoretical Computer Science
robert.legenstein@tugraz.at
oezdenizci@tugraz.at

Deep Learning VO - WS 25/26
Lecture 2

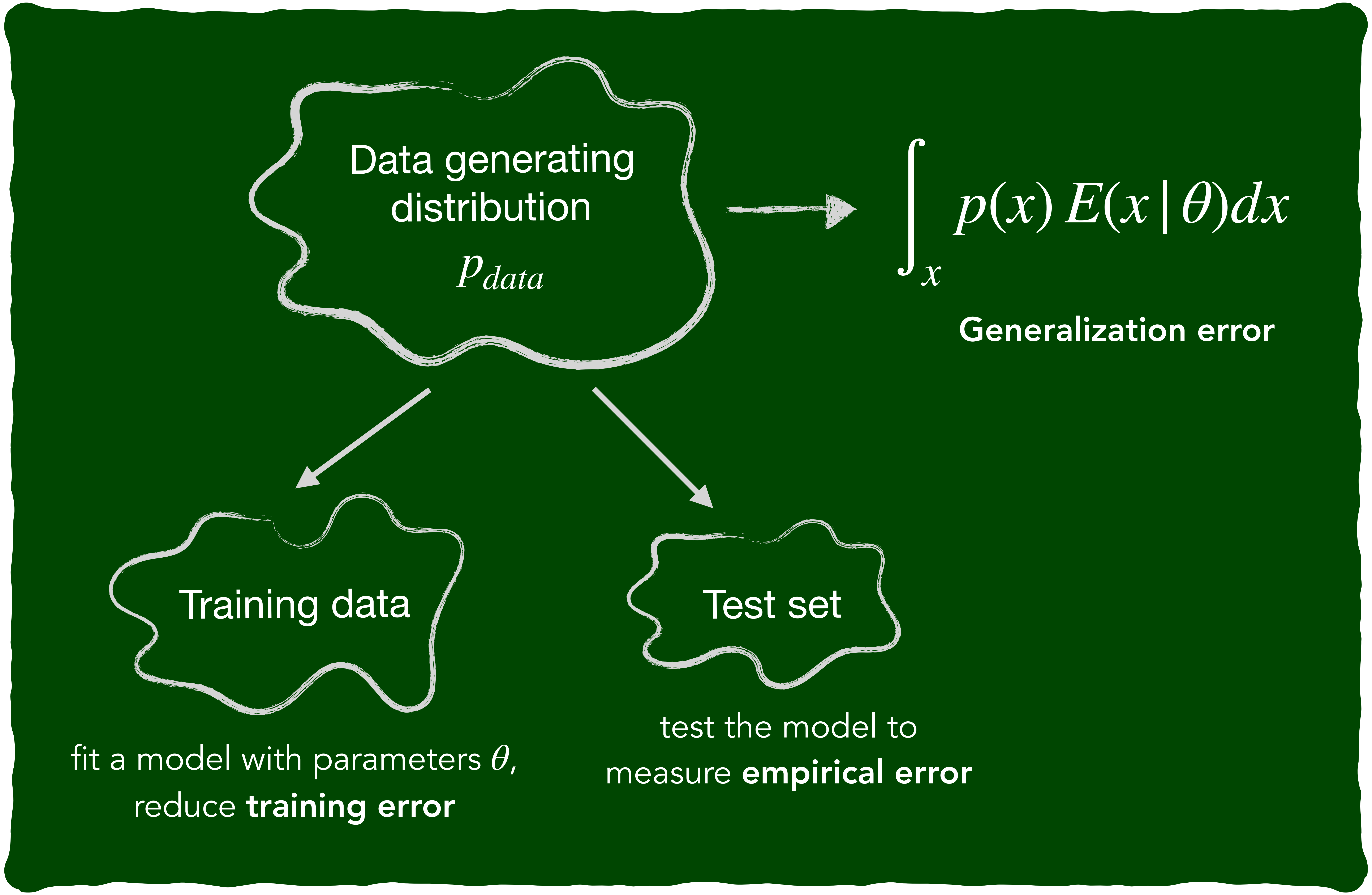
Today

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

Recap on Statistical Learning Theory

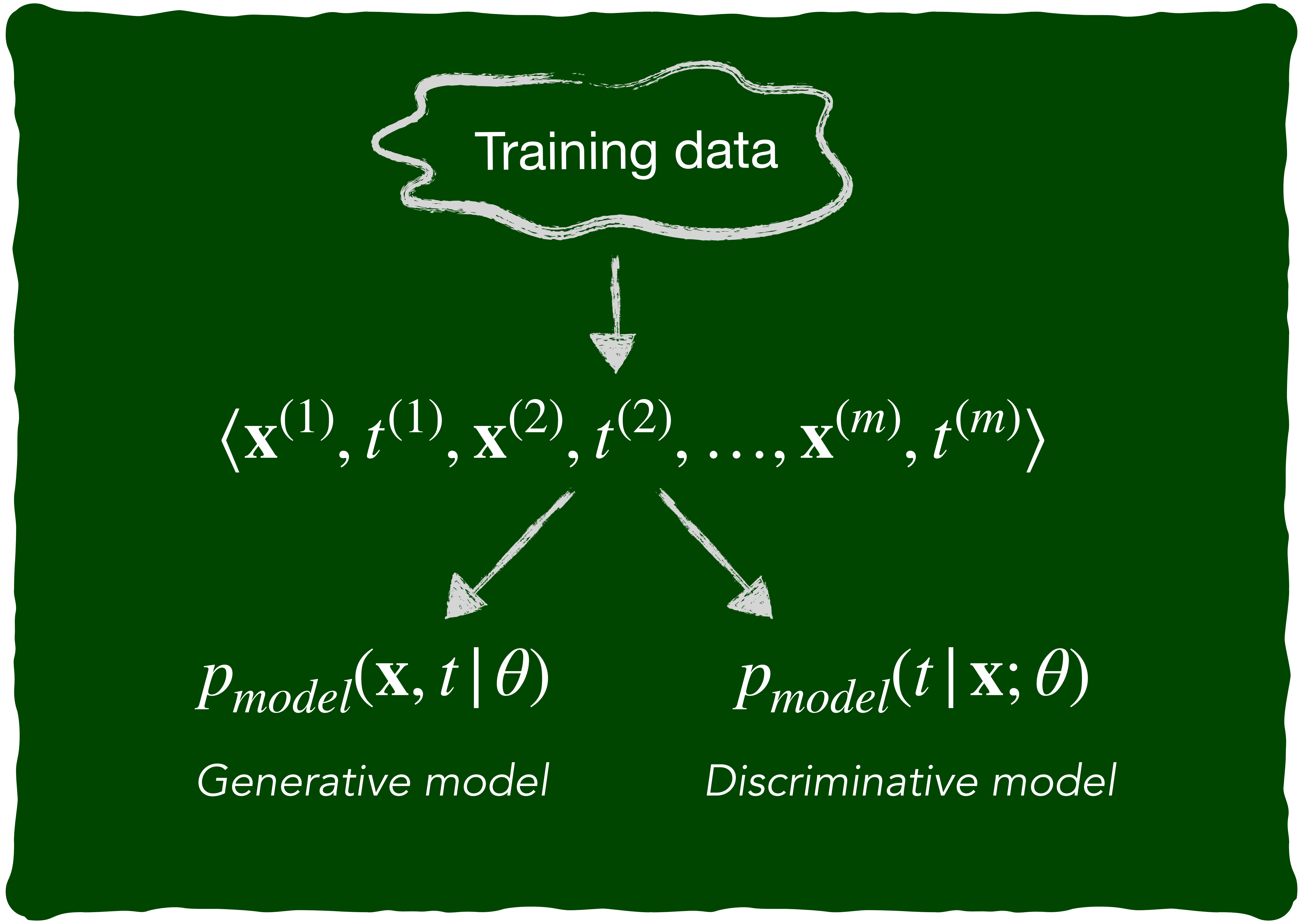
Data-generating distribution:

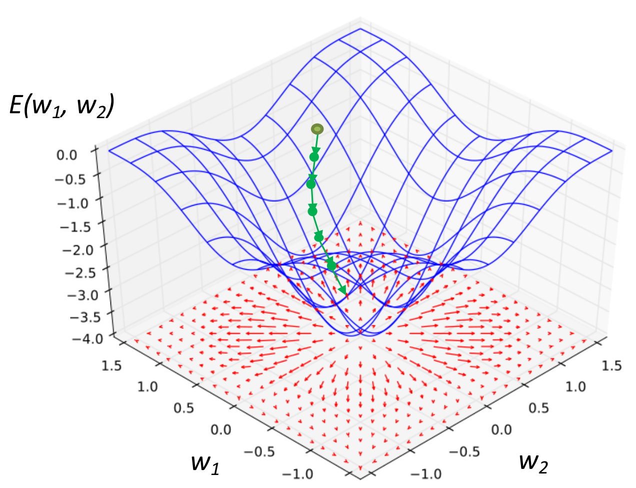
- We assume that training and test-data are generated from the *data-generating distribution* p_{data} .
- Our goal is to achieve minimal generalization error (i.e., error on randomly drawn samples from this distribution).

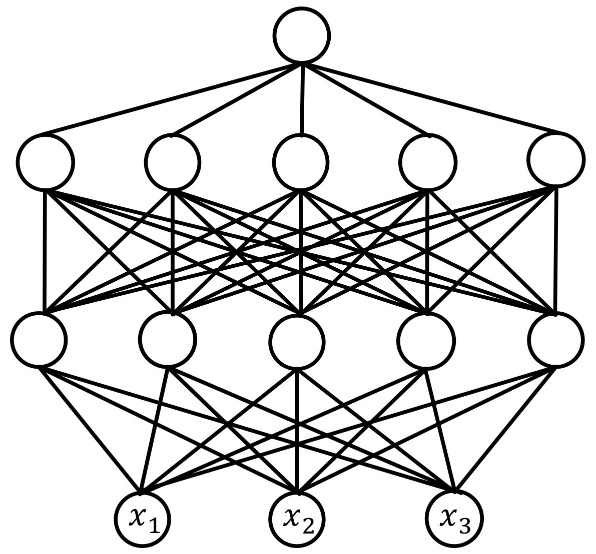


Why do we talk about "estimators"?

- 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., $p_{model}(t \mid \mathbf{x}; \theta)$ for a classification problem, where θ are the network weights.
- Training is the *estimation* of the parameters θ.







Point estimation

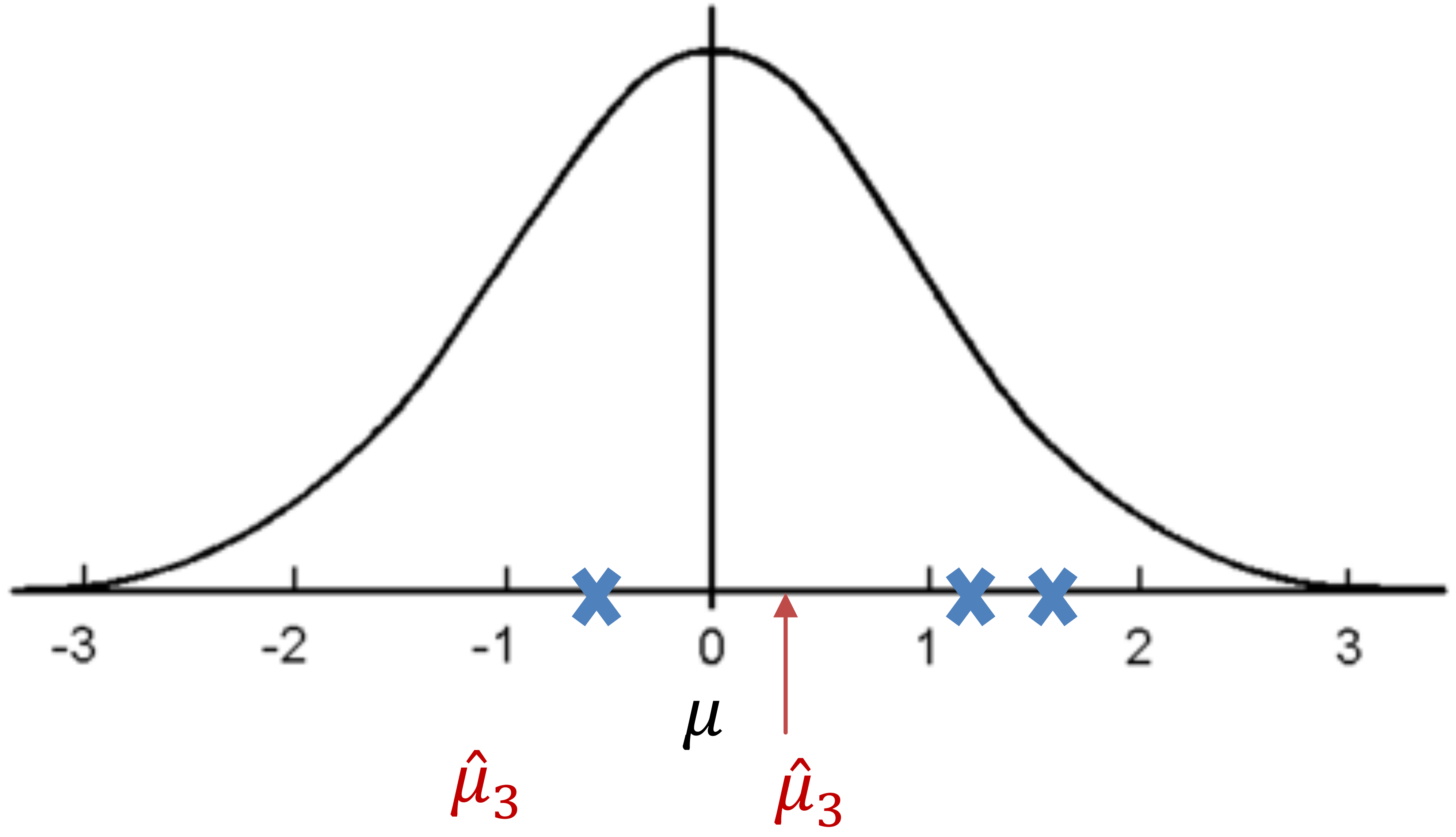
Point estimation is the attempt to provide the single "best" prediction of some quantity of interest.

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

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

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

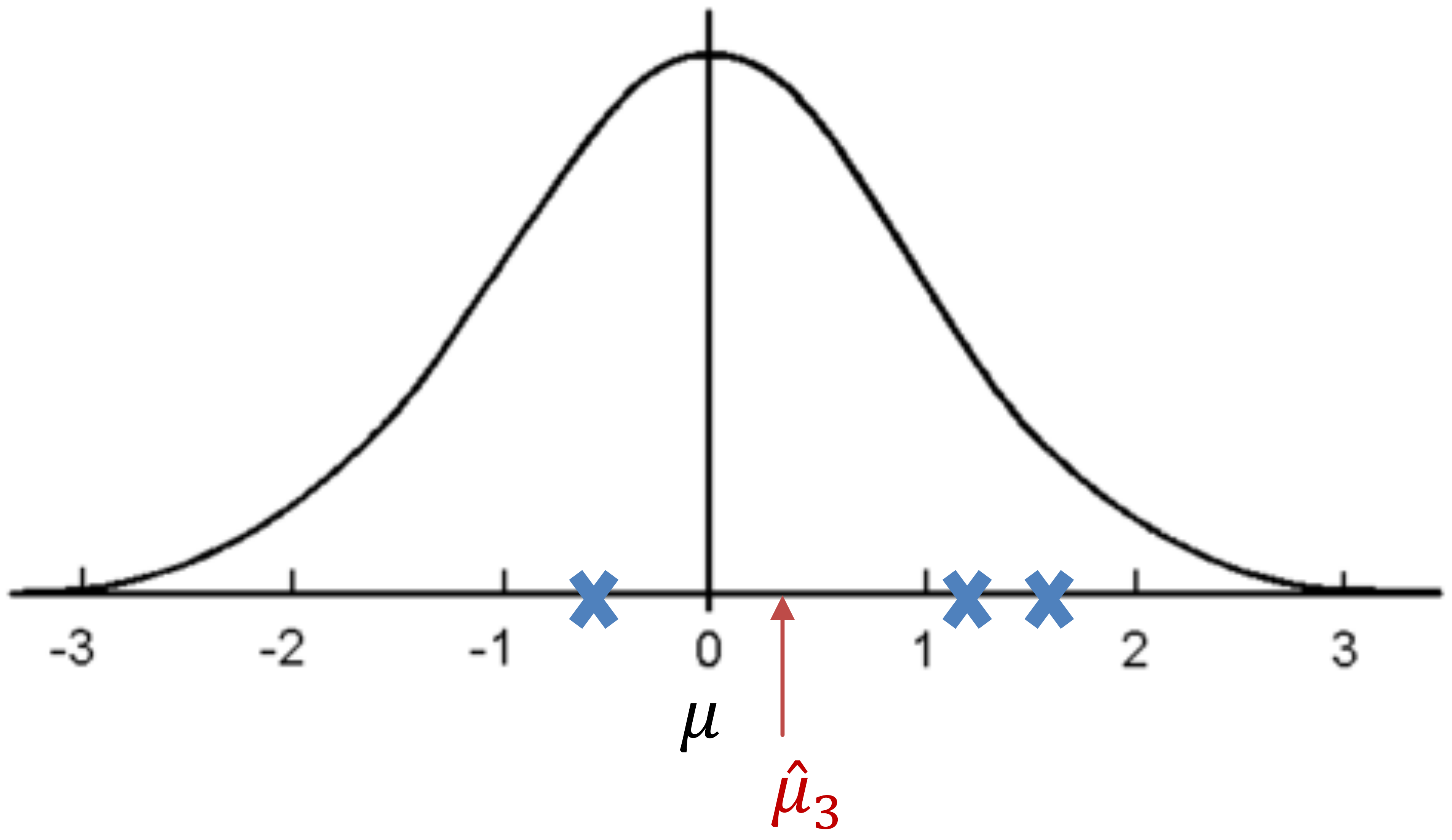
$$\hat{\mu}_m = g(x^{(1)}, \ldots, x^{(m)}) = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$$



Bias

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

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



<u>**Example:**</u> 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: $\mathrm{Var}(\hat{\theta}_m) = \mathbb{E}[(\hat{\theta}_m - \mathbb{E}[\hat{\theta}_m])^2]$.

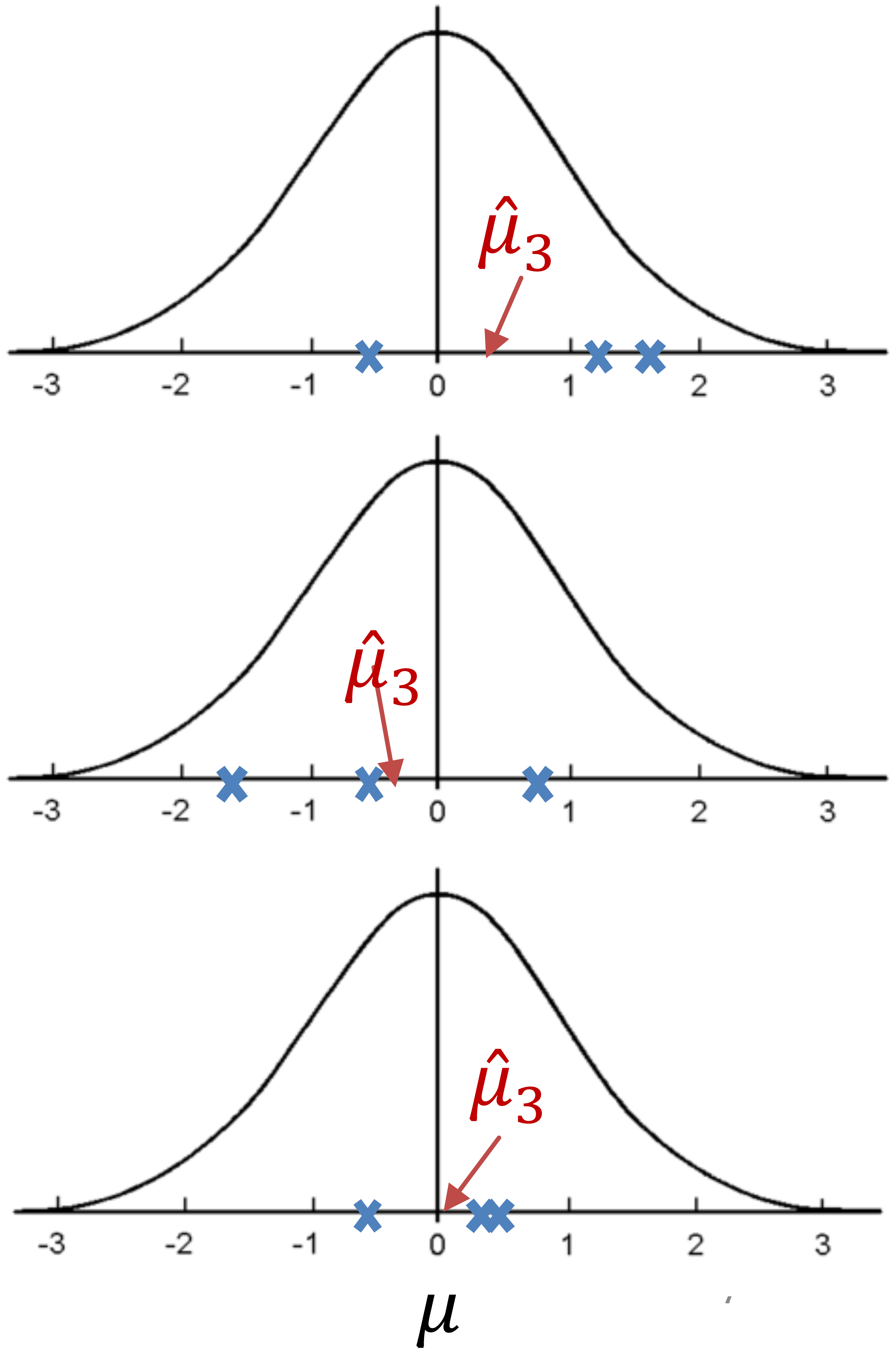
- How strongly does the estimator vary for different drawings of the data set.

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

<u>**Example:**</u> For a Gaussian with true variance σ^2, the standard error of the sample mean estimator is:

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

What does this imply?



Consistency

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

An estimator is (weakly) consistent if:

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

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

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

- [x] Estimators
- [ ] Maximum Likelihood & Maximum A Posteriori Estimation
- [ ] Classification & Decision Theory

Maximum Likelihood (ML) Estimation

- We observe some i.i.d. data $\mathbf{X} = \langle \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)} \rangle$ drawn from $p_{data}(\mathbf{x})$.
- Let $p_{model}(\mathbf{x}; \theta)$ be a parametric family of distributions.
- Maximum likelihood estimate for θ:

$$\begin{aligned}\theta_{ML} &= \arg\max_{\theta} p_{model}(\mathbf{X}; \theta) \\ &= \arg\max_{\theta} \prod_{i=1}^{m} p_{model}(\mathbf{x}^{(i)}; \theta) \\ &= \arg\max_{\theta} \sum_{i=1}^{m} \log p_{model}(\mathbf{x}^{(i)}; \theta)\end{aligned}$$

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 θ.

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

Example: Maximum likelihood estimate of the Gaussian mean

$$X = \langle x^{(1)}, x^{(2)}, \ldots, x^{(m)} \rangle \qquad p_{model}(x \,|\, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2} \right)$$

$$x \in \mathbb{R}, \;\; \mu \in \mathbb{R}, \;\; \sigma > 0$$

Likelihood

$$p_{model}(X \,|\, \mu, \sigma) = \prod_{i=1}^{m} p_{model}(x^{(i)} \,|\, \mu, \sigma)$$

$$\hat{\mu}_{ML} = \arg\max_{\mu} \sum_{i=1}^{m} \log p_{model}(x^{(i)} \,|\, \mu, \sigma)$$

$$= \arg\max_{\mu} \sum_{i=1}^{m} \left(\log \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{(x^{(i)}-\mu)^2}{2\sigma^2} \right)$$

$$= \arg\min_{\mu} \sum_{i=1}^{m} (x^{(i)} - \mu)^2$$

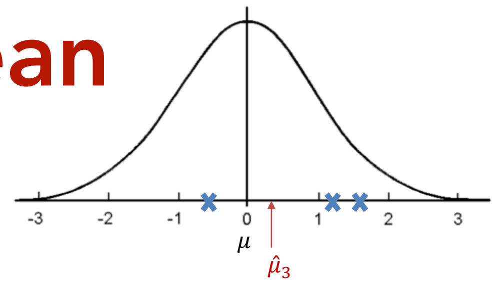
Compute the minimum:

$$\frac{\partial}{\partial \mu} \sum_{i=1}^{m} (x^{(i)} - \mu)^2 = -2 \sum_{i=1}^{m} (x^{(i)} - \mu)$$

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

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

sample mean



Conditional Log-Likelihood

- In regression or classification problems with $\mathbf{X} = \langle \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)} \rangle$ and targets $\mathbf{t} = (t^{(1)}, \ldots, t^{(m)})^T$ 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

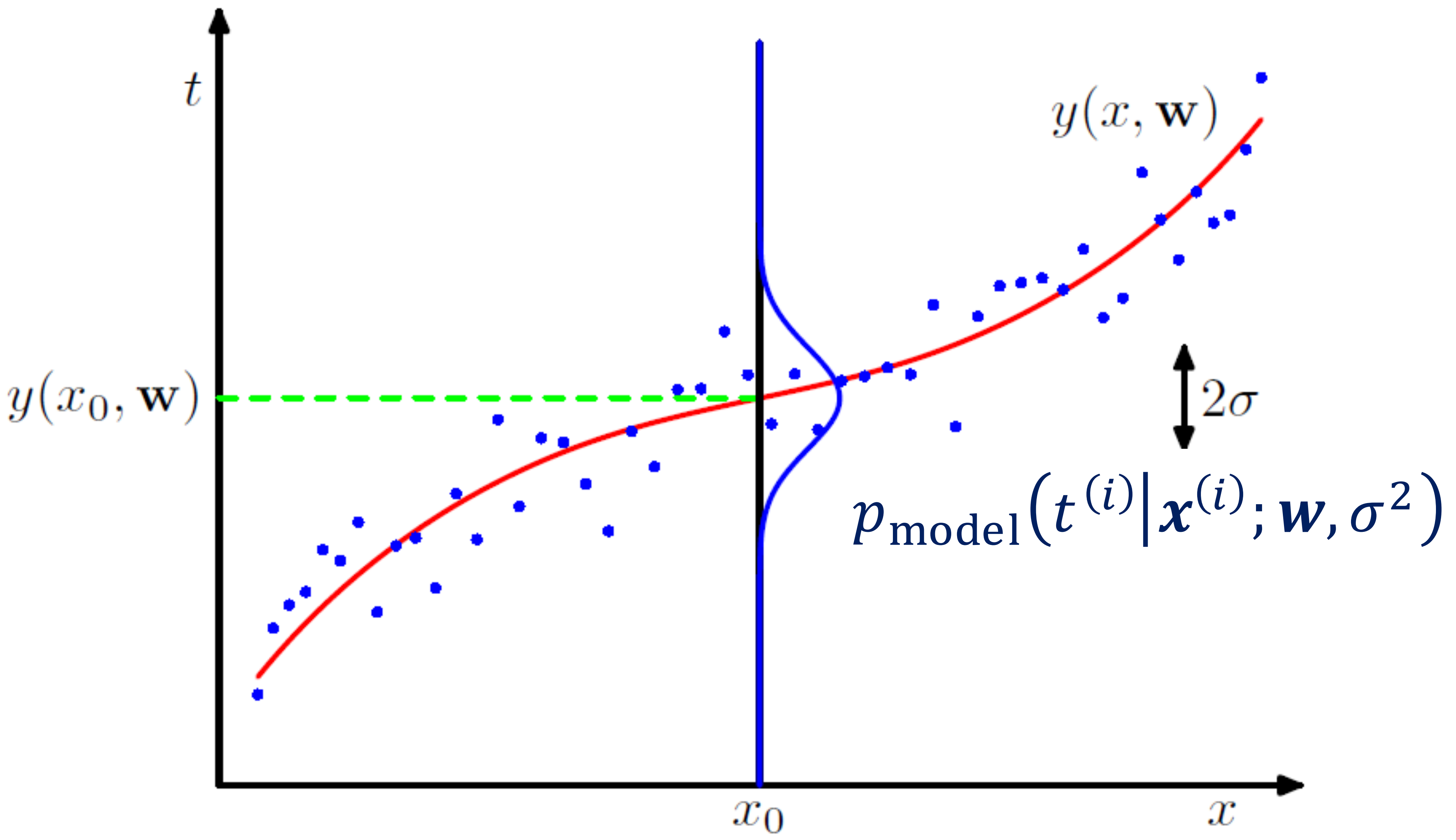
Given training examples with $\langle \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)} \rangle$ and $\mathbf{t} = (t^{(1)}, \ldots, t^{(m)})^T$.

Consider a probabilistic model for the data:

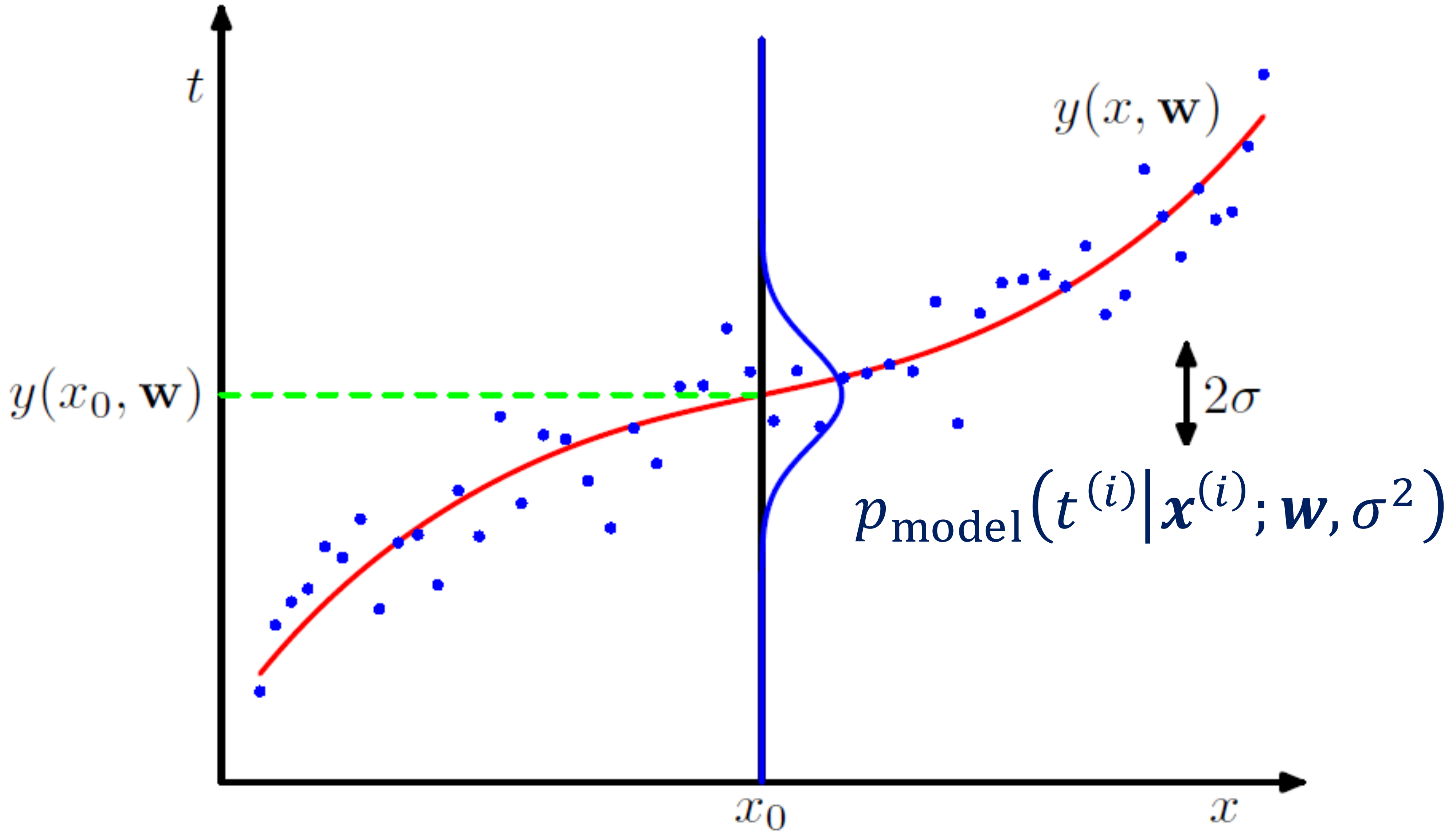
$$p_{model}(t^{(i)} \,|\, \mathbf{x}^{(i)}; \mathbf{w}, \sigma^2) = N(t^{(i)}; y_w(\mathbf{x}^{(i)}), \sigma^2).$$

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

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



Example: Regression with Gaussian noise



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

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

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

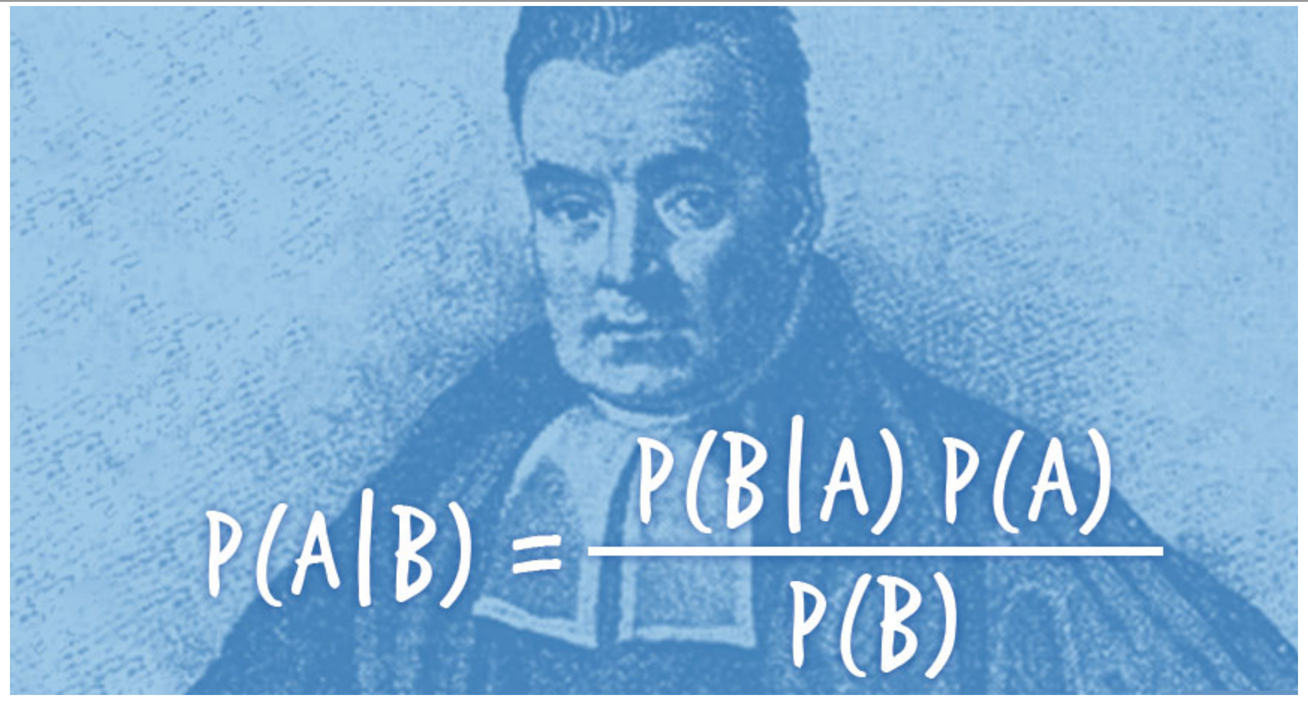
$$= \arg\max_{\mathbf{w}} - m \log\left(\sigma\sqrt{2\pi}\right) - \sum_{i=1}^{m} \frac{\left(t^{(i)} - y_w(\mathbf{x}^{(i)})\right)^2}{2\sigma^2}$$

$$= \arg\min_{\mathbf{w}} \sum_{i=1}^{m} \left(y_w(\mathbf{x}^{(i)}) - t^{(i)}\right)^2 \longrightarrow$$

Sum squared error

Maximum A Posteriori (MAP) Estimation

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



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

The MAP estimator uses the point of maximum posterior probability:

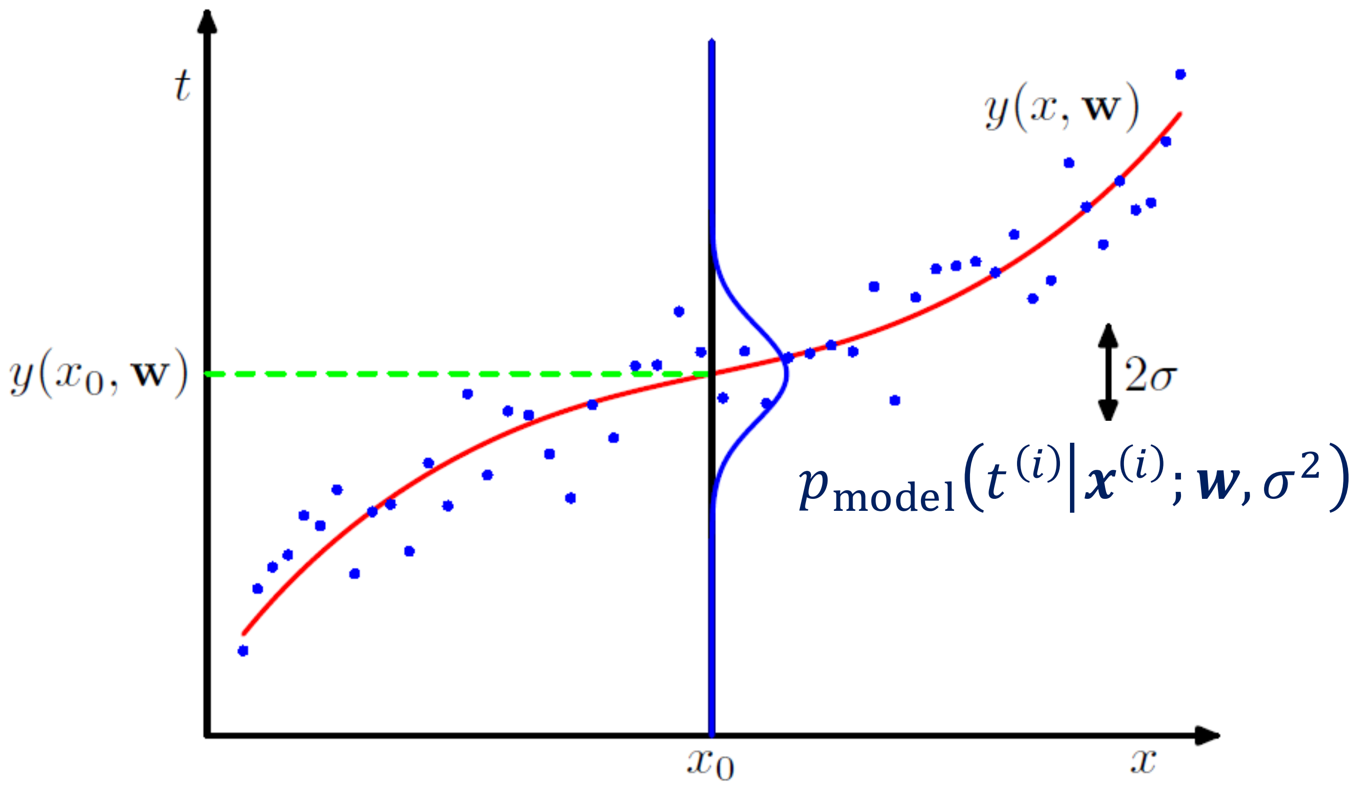
$$\begin{aligned}
\theta_{MAP} &= \arg\max_{\theta} p_{model}(\theta \,|\, \mathbf{X}) \\
&= \arg\max_{\theta} \left[\log p_{model}(\mathbf{X} \,|\, \theta) + \log p(\theta)\right] \\
&= \arg\max_{\theta} \left[\sum_{i=1}^{m} \log p_{model}(\mathbf{x}^{(i)} \,|\, \theta) + \log p(\theta)\right]
\end{aligned}$$

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

Example: Regression with Gaussian noise <u>and Gaussian Prior</u>

Consider a probabilistic model for the data:

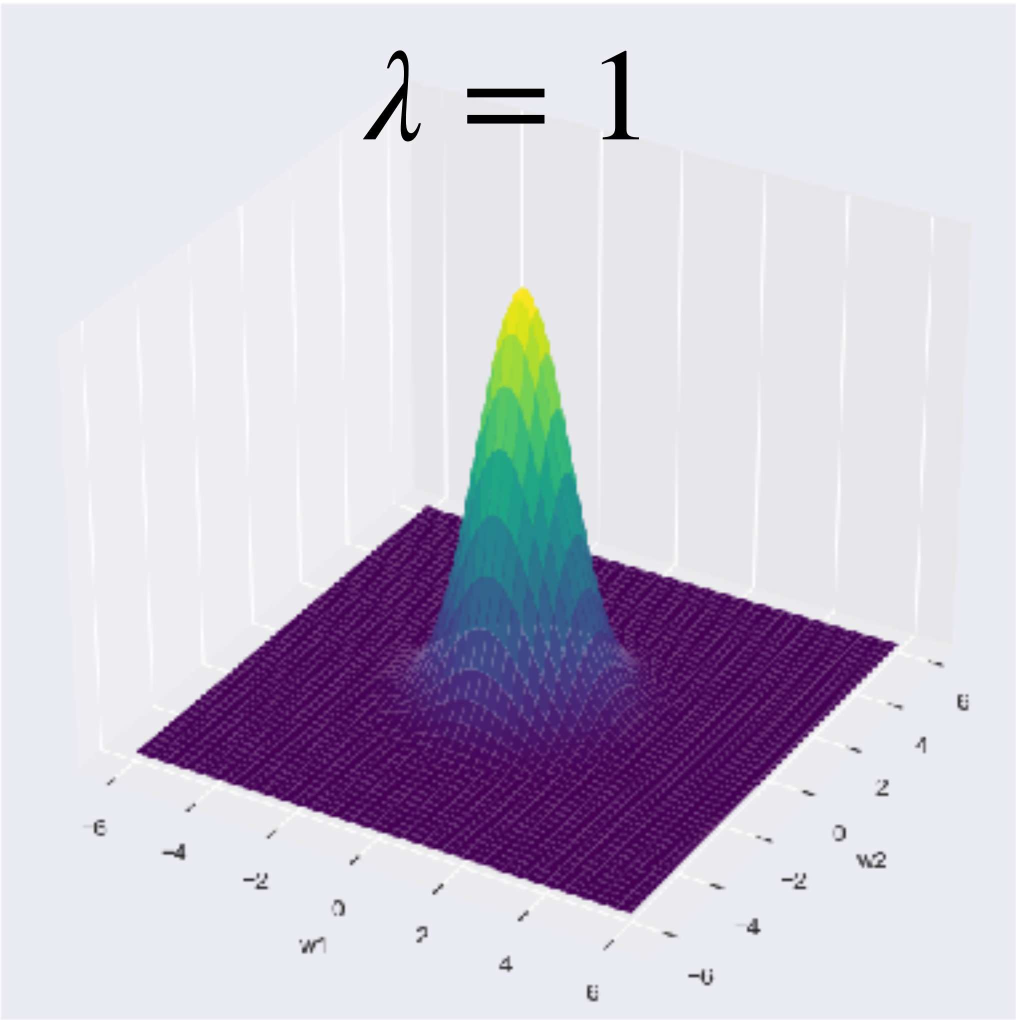
$$p_{model}(t^{(i)} \mid \mathbf{x}^{(i)}; \mathbf{w}, \sigma^2) = N(t^{(i)}; y_w(\mathbf{x}^{(i)}), \sigma^2)$$

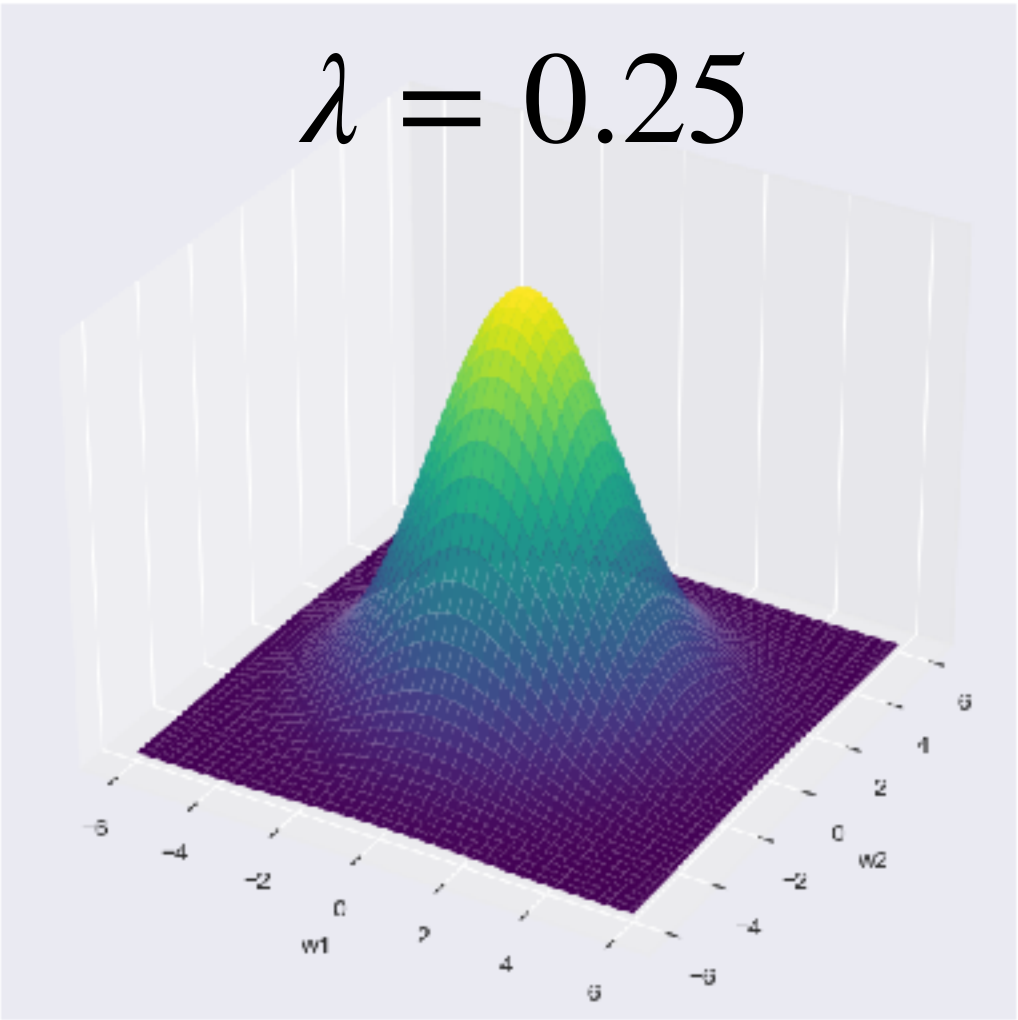


We assume a Gaussian prior on the parameters:

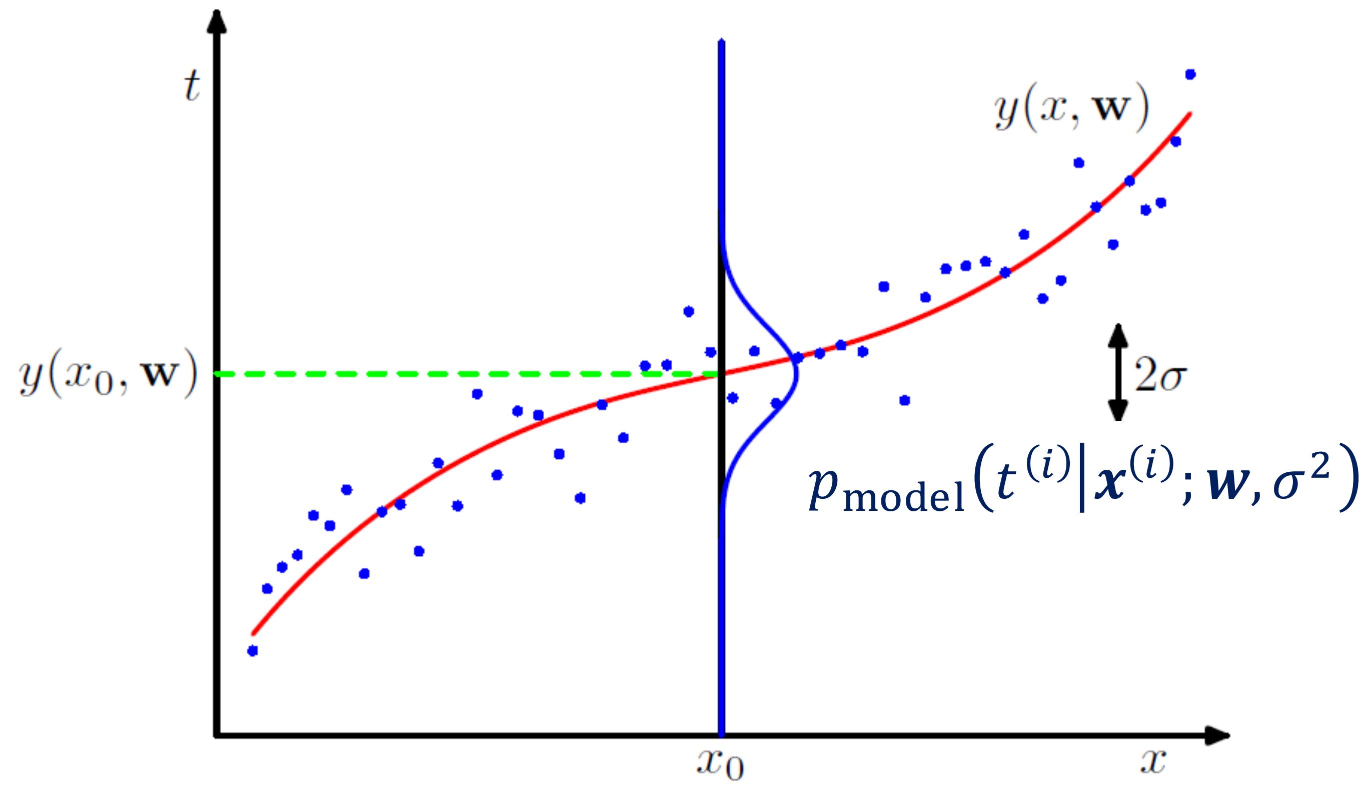
$$p(\mathbf{w}) = N(\mathbf{w}; 0, \frac{1}{\lambda}\mathbf{I}) = \left(\frac{\lambda}{2\pi}\right)^{\frac{D}{2}} \exp\left(-\frac{\lambda}{2}\mathbf{w}^T\mathbf{w}\right)$$

higher λ $\longrightarrow$ smaller parameters





Example: Regression with Gaussian noise <u>and Gaussian Prior</u>



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 λ → smaller parameters

We obtain the MAP estimator:

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

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

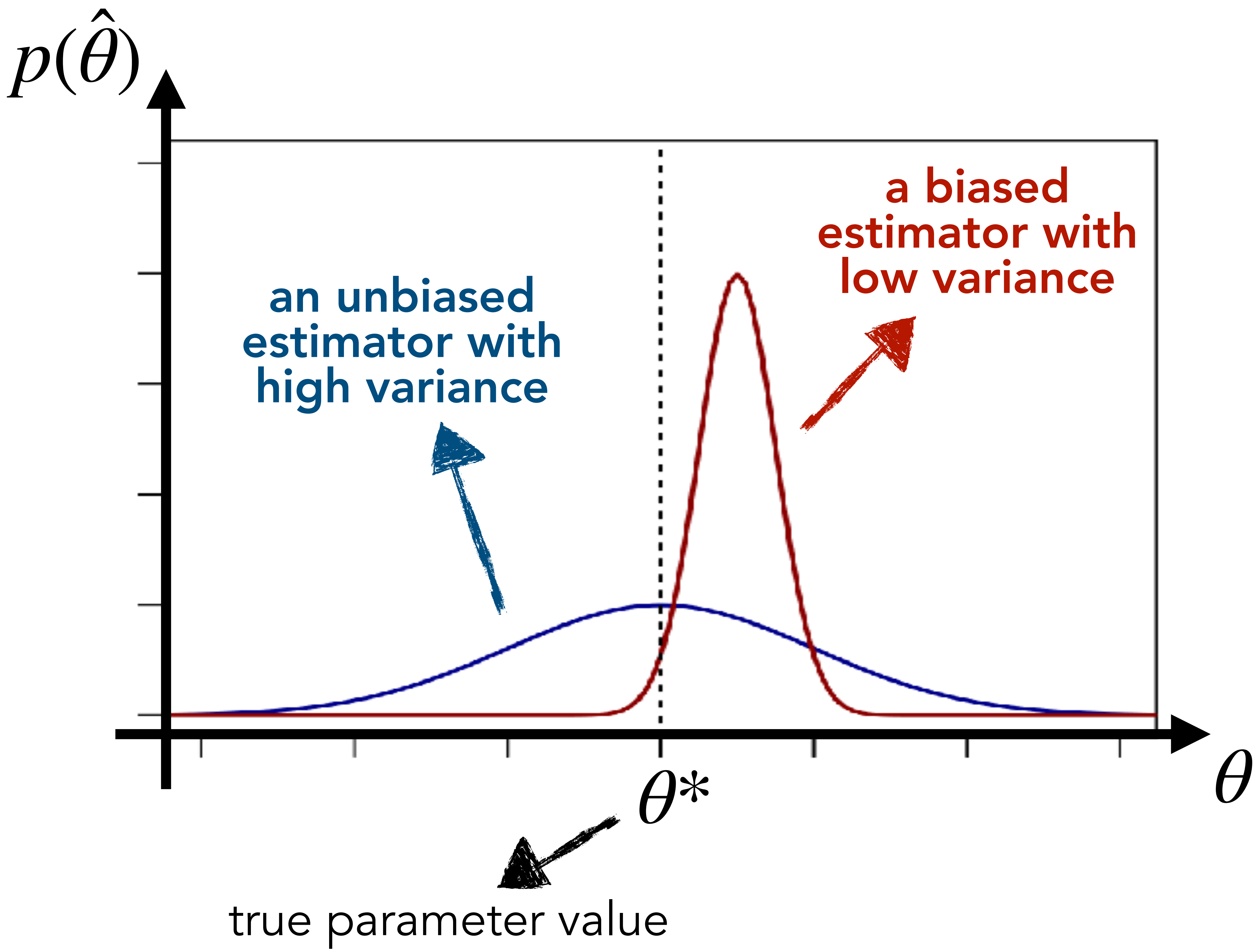
→ **Regularization**

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.

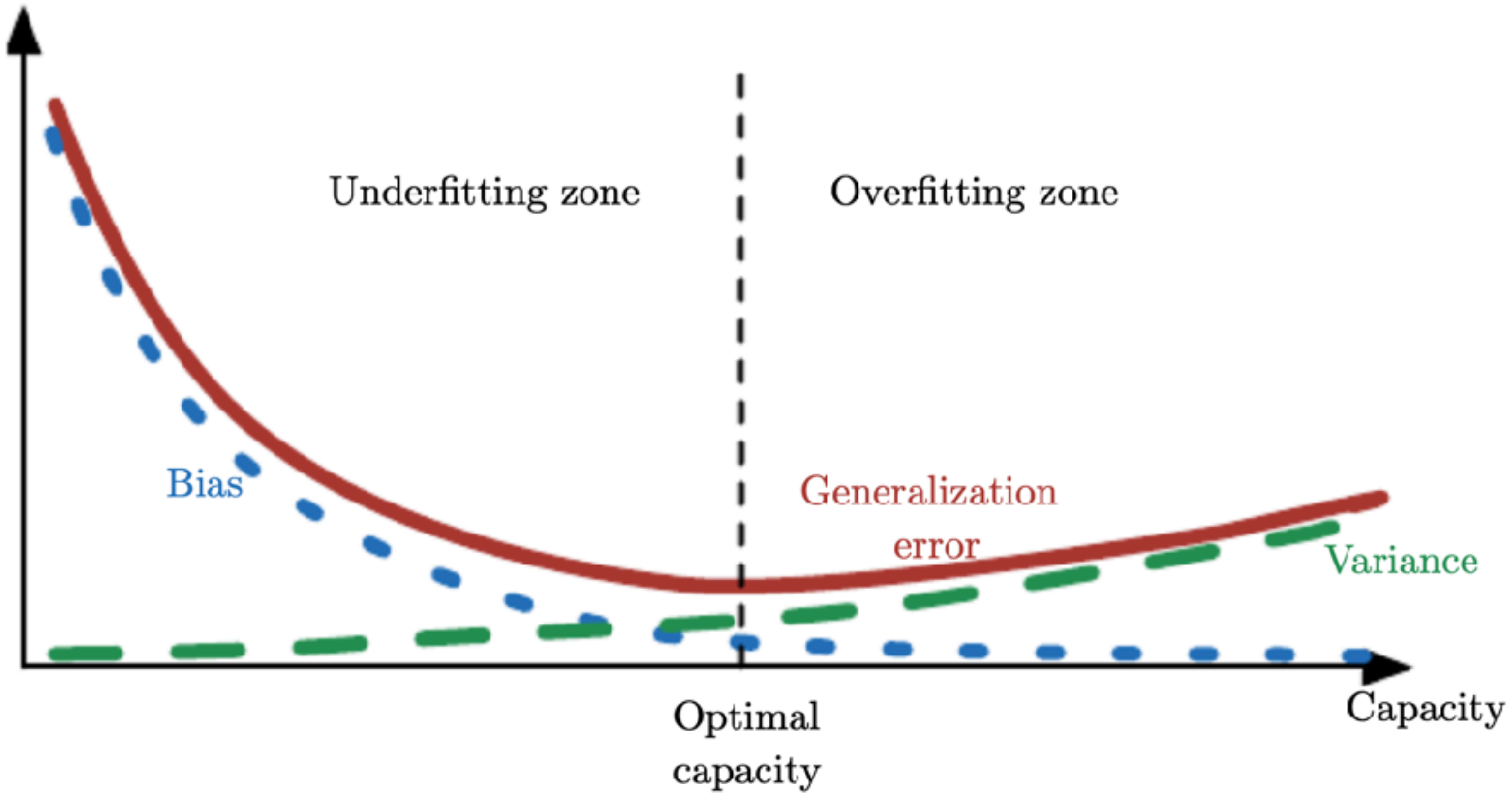
Mean squared error (MSE) of the estimate:

$$MSE = \mathbb{E}\left[\left(\hat{\theta}_m - \theta\right)^2\right]$$

$$= Bias\left(\hat{\theta}_m\right)^2 + Var\left(\hat{\theta}_m\right)$$

Desirable estimators have small MSE, i.e.,

estimators with **small bias and small variance**.



Today

- [x] Estimators
- [x] Maximum Likelihood & Maximum A Posteriori Estimation
- [ ] Classification & Decision Theory

Classification Problems

- We have K classes $C_1, \ldots, C_K$, 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: $y : \mathbb{R}^N \rightarrow \{1, \ldots, K\}$

Classification Problems

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

class-conditional class-prior

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

Decision: Decide for one class in some optimal way.

Decision rule: $y : \mathbb{R}^N \to \{1, \ldots, K\}$

An example: "dog" vs "cat"

C_1

C_2

$P(cat) = 0.8$

$P(dog) = 0.2$

$$\sum_{k=1}^{K} P(C_k) = 1$$

Probability of observing a cat. $\longleftarrow P(cat)$

Distribution over images of cats. $\longleftarrow P(\mathbf{x} \mid cat)$

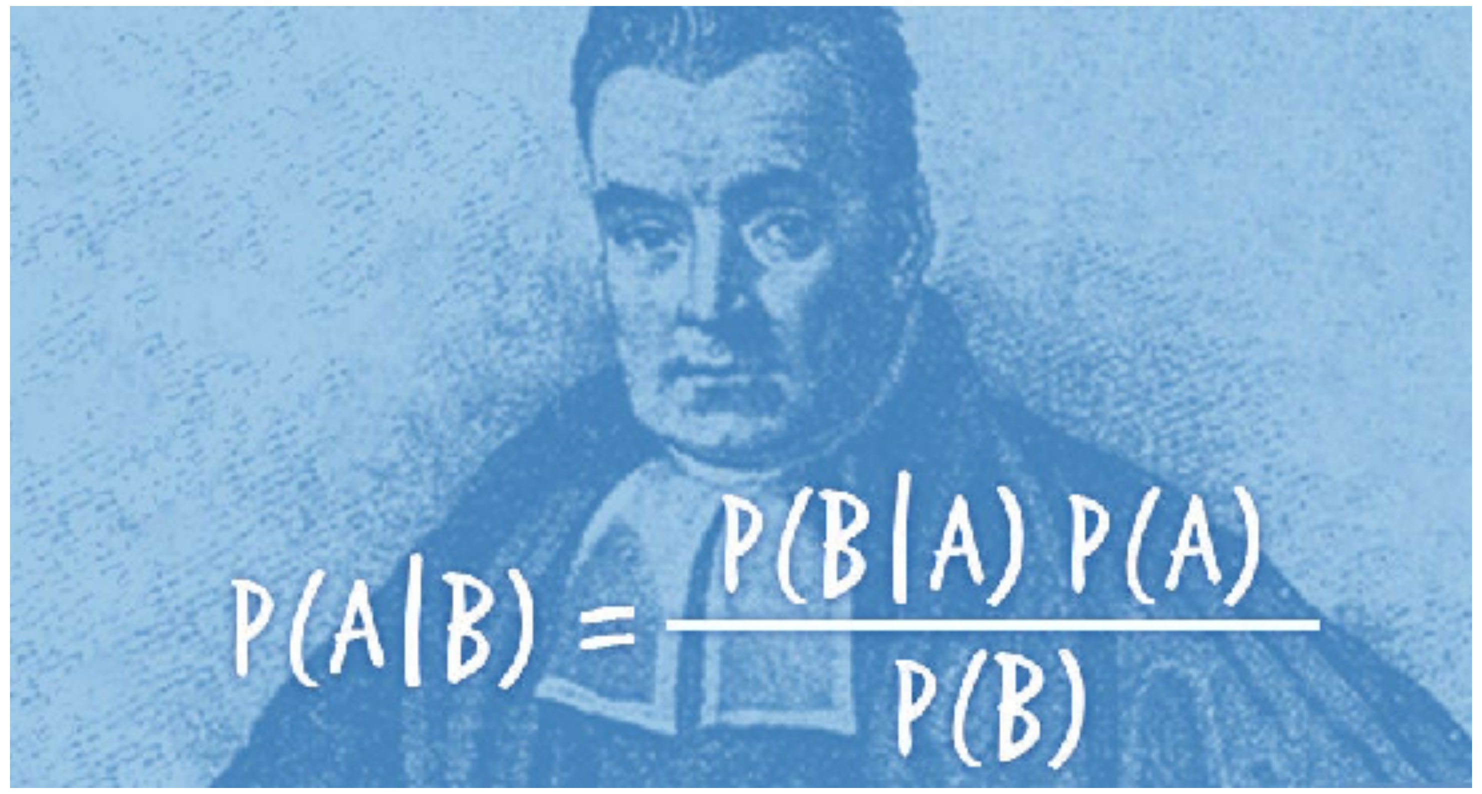
Probability that a given image is a cat. $\longleftarrow P(cat \mid \mathbf{x})$

Inference

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

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

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



<u>**Discriminative model**</u>

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

Sigmoid Function

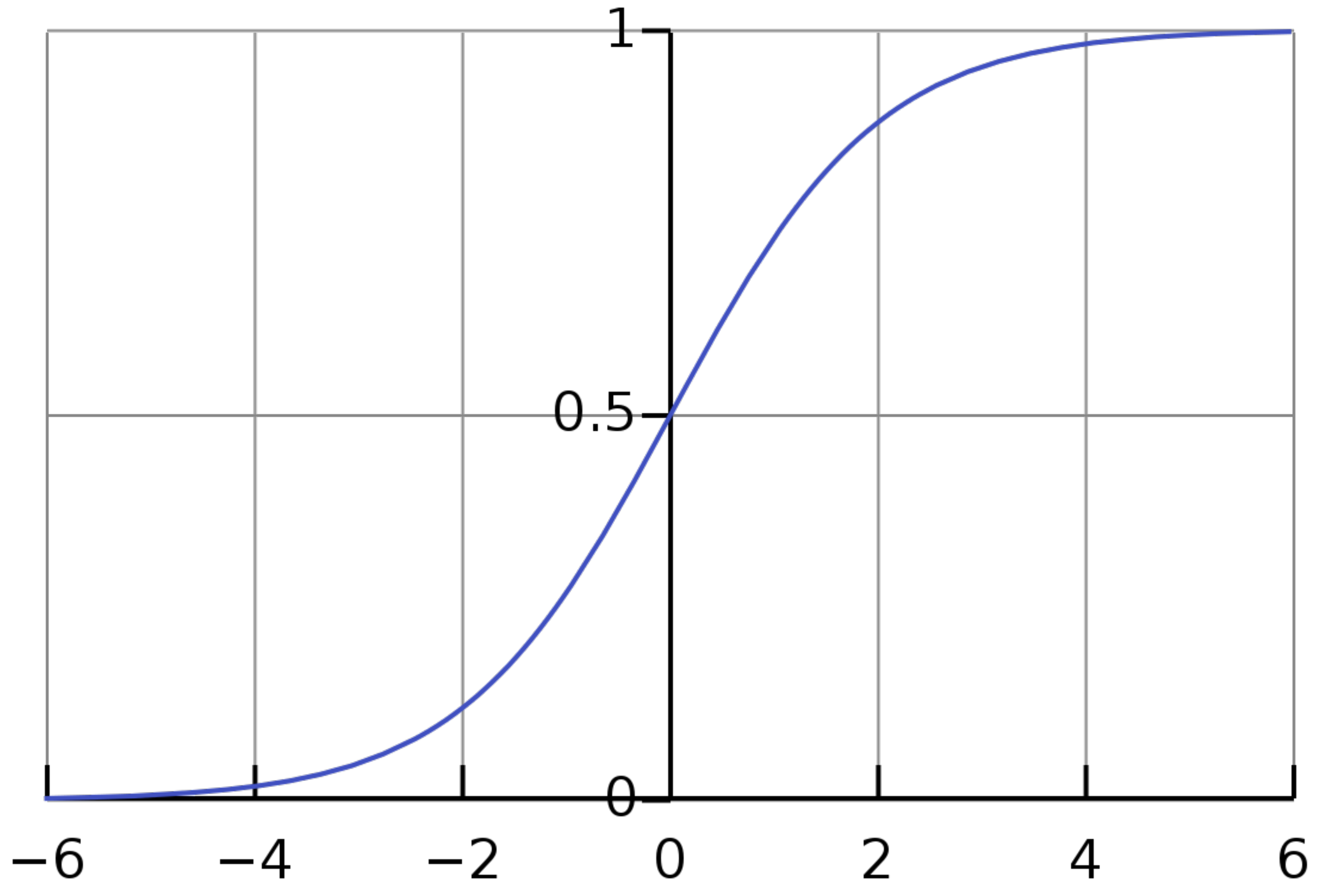
In binary classification:

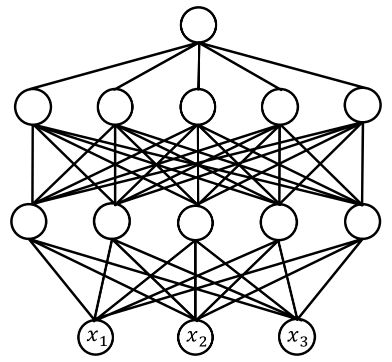
$$P(C_1 \mid \mathbf{x}) = \frac{P(\mathbf{x}, C_1)}{P(\mathbf{x})} = \frac{P(\mathbf{x}, C_1)}{P(\mathbf{x}, C_1) + P(\mathbf{x}, C_2)} = \frac{1}{1 + \frac{P(\mathbf{x}, C_2)}{P(\mathbf{x}, C_1)}} = \frac{1}{1 + \exp(-a)} = \sigma(a)$$

where

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

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



- 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).

Softmax Function

In multi-class classification:

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

where

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

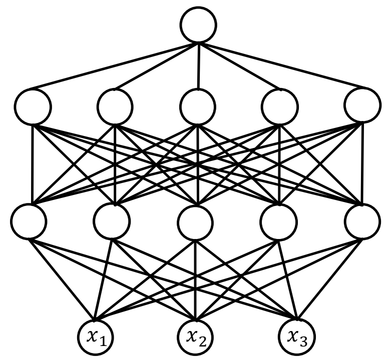
- The *softmax* can be interpreted as representing the posterior probabilities for a multi-class classification problem:

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

'logits' (or 'scores') **Probabilities**

$$\begin{bmatrix} 1.3 \\ 5.1 \\ 2.2 \\ 0.7 \\ 1.1 \end{bmatrix} \longrightarrow \boxed{\frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}} \longrightarrow \begin{bmatrix} 0.02 \\ 0.90 \\ 0.05 \\ 0.01 \\ 0.02 \end{bmatrix}$$

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



Maximum Likelihood (Cross-Entropy Error Function)

- We want to estimate the parameters of ***a discriminative model***.
- Given training examples with $\mathbf{X} = \langle \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(M)} \rangle$ and $\mathbf{t} = (t^{(1)}, \ldots, t^{(M)})^T$, 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:

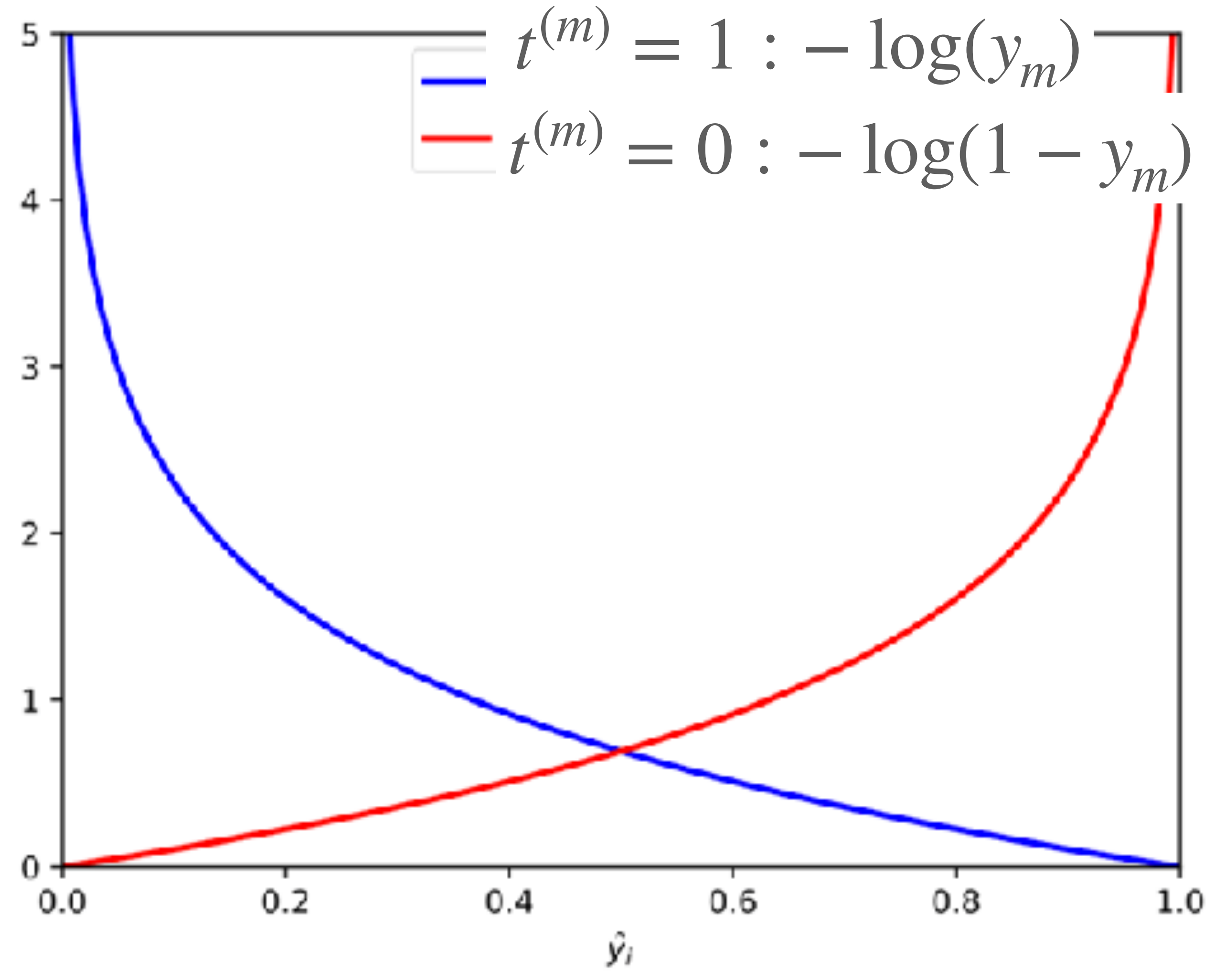
$$y_m := P(C_1 \,|\, \mathbf{x}^{(m)}) \quad (\text{ and thus } P(C_2 \,|\, \mathbf{x}^{(m)}) = 1 - y_m \,).$$

- **We are using Maximum Likelihood 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

Decision Theory: Minimizing Classification Rate

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

Goal 1: Make as few misclassifications as possible.

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

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

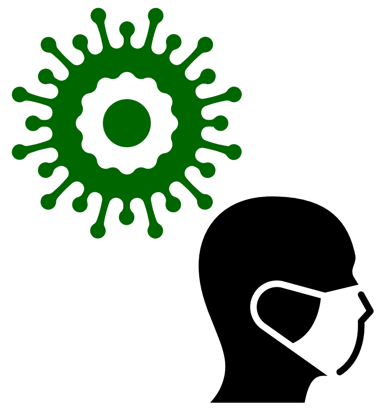
Choose the class with maximum posterior probability

Decision Theory: Minimizing Expected Loss

- We can introduce a loss matrix L.

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

Not all errors are equal!



Predicted class

True class

$$\begin{array}{ccc} & \text{normal} & \text{corona} \\ \text{normal} & \left[\begin{matrix} 0 \\ 100 \end{matrix}\right. & \left.\begin{matrix} 1 \\ 0 \end{matrix}\right] \\ \text{corona} & & \end{array}$$

Goal 2: Minimize expected loss.

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

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

Today

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

Questions?