ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

inf | Informatik
Computer Science

Introduction to Machine Learning

Generalization and Model Validation

Dr. Ilija Bogunovic
Learning and Adaptive Systems (las.ethz.ch)

Recall: Least-squares linear regression optimization [Legendre 1805, Gauss 1809]

- Given data set $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$

$$\boxed{\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_{i=1}^{n} (y_i - \mathbf{w}^T \mathbf{x}_i)^2}$$

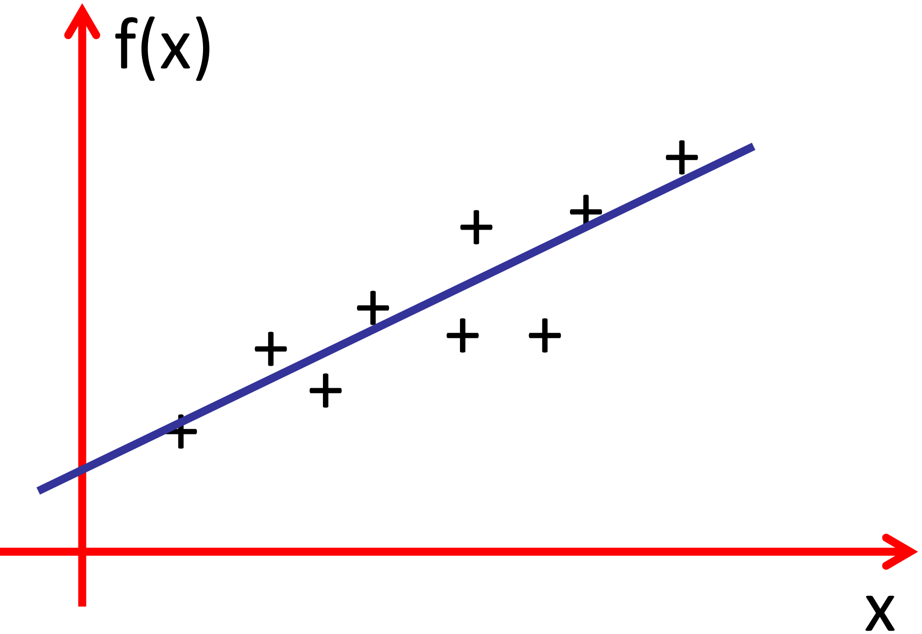
- Last lecture, discussed how to solve using closed form & gradient descent

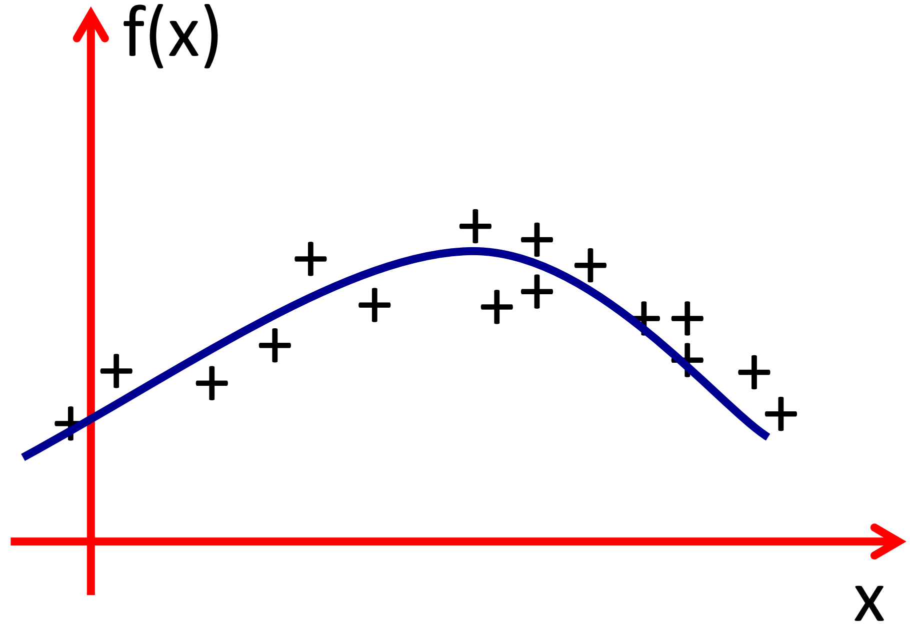
Supervised learning summary so far

Representation/ features	Linear hypotheses
Model/ objective:	Loss-function Squared loss, l_p loss
Method:	Exact solution, Gradient Descent
Evaluation metric:	Empirical risk = (mean) squared error

Recall: Important choices in regression

- What types of functions f should we consider? Examples





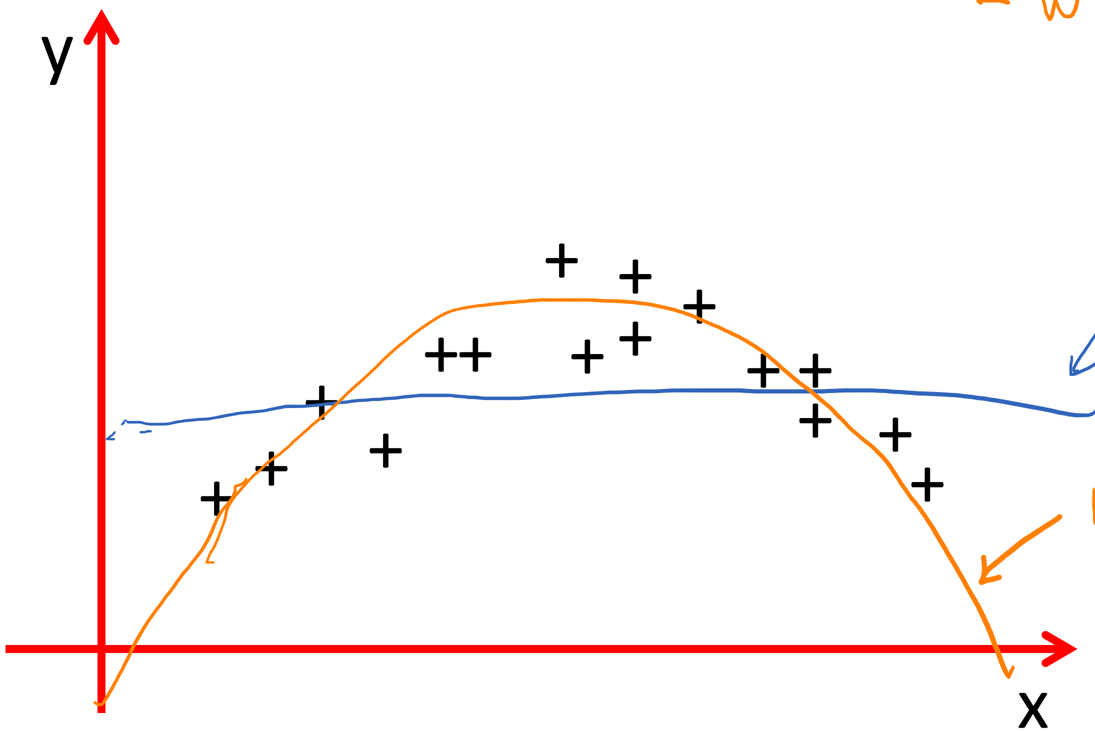
- How should we measure goodness of fit?

Fitting nonlinear functions

- How about functions like this:

$$f(x) = ax^2 + bx + c$$
$$= w^T \tilde{x}$$

$$w = [a, b, c]^T$$
$$\tilde{x} = [x^2, x, 1]$$



Least squares fit with Linear model

Least squares fit with "quadratic" model (Linear model on features $x^2, x, 1$)

Linear regression for polynomials

We can fit non-linear functions via linear regression, using nonlinear features of our data (basis functions)

$$f(\mathbf{x}) = \sum_{i=1}^{d} w_i \phi_i(\mathbf{x})$$

$x \in \mathbb{R}^p$

$x \mapsto \tilde{x} = \phi(x) \in \mathbb{R}^d$

$w \in \mathbb{R}^d$

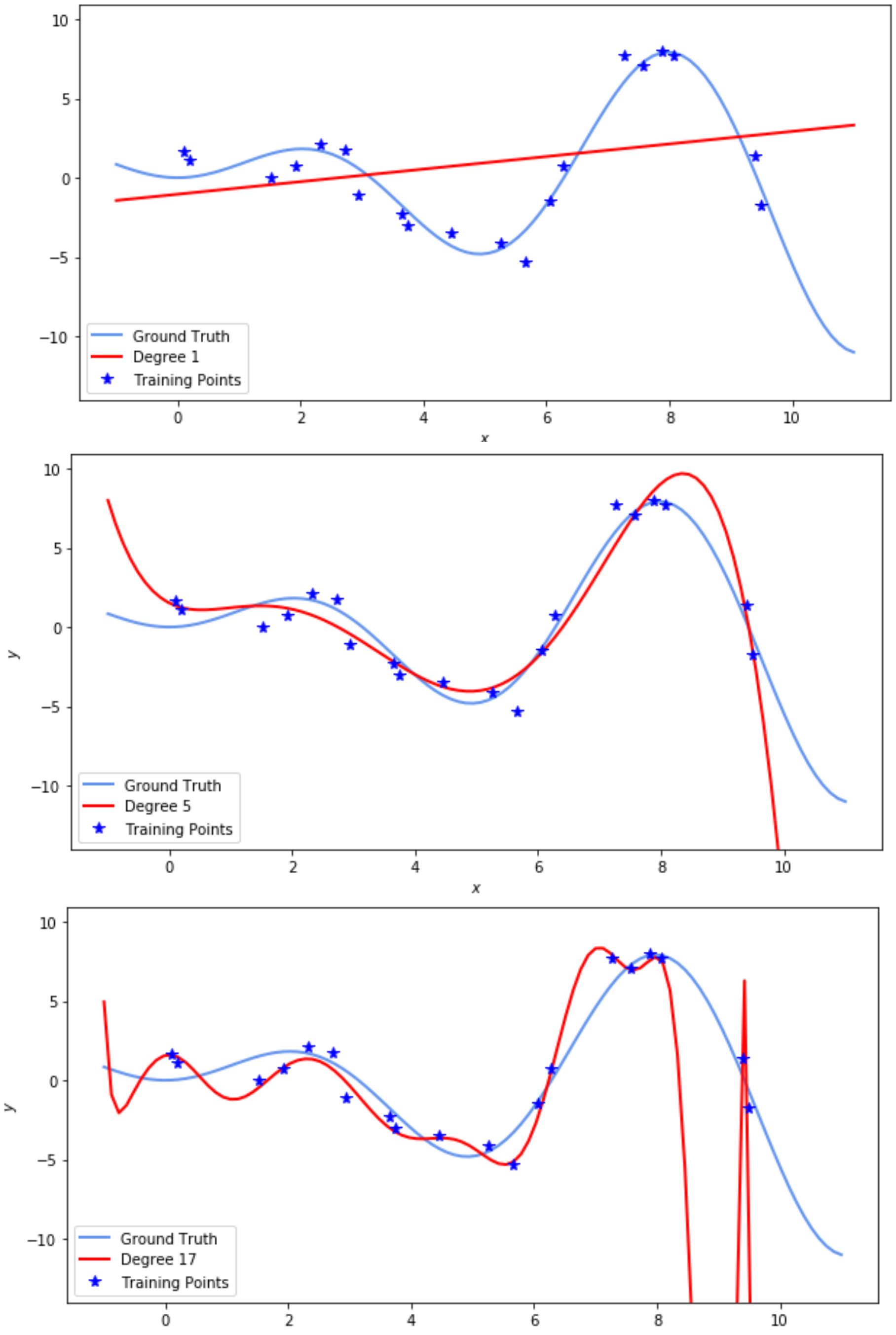
1 dim. : $\phi(x) = [1, x, x^2, \ldots, x^k]$

2 dim. : $\phi(x) = [1, x_1, x_2, x_1 x_2, x_1^2, x_2^2 \ldots]$

⋮

p dim : $\phi(x)$ vector of all monomials in $x_1 \ldots x_p$ of degree up to k.

Demo: Linear regression on polynomials



Underfitting

Overfitting

Supervised learning summary so far

Representation/ features

Linear hypotheses, nonlinear hypotheses through feature transformations

Model/ objective:

Loss-function

Squared loss, l_p-loss

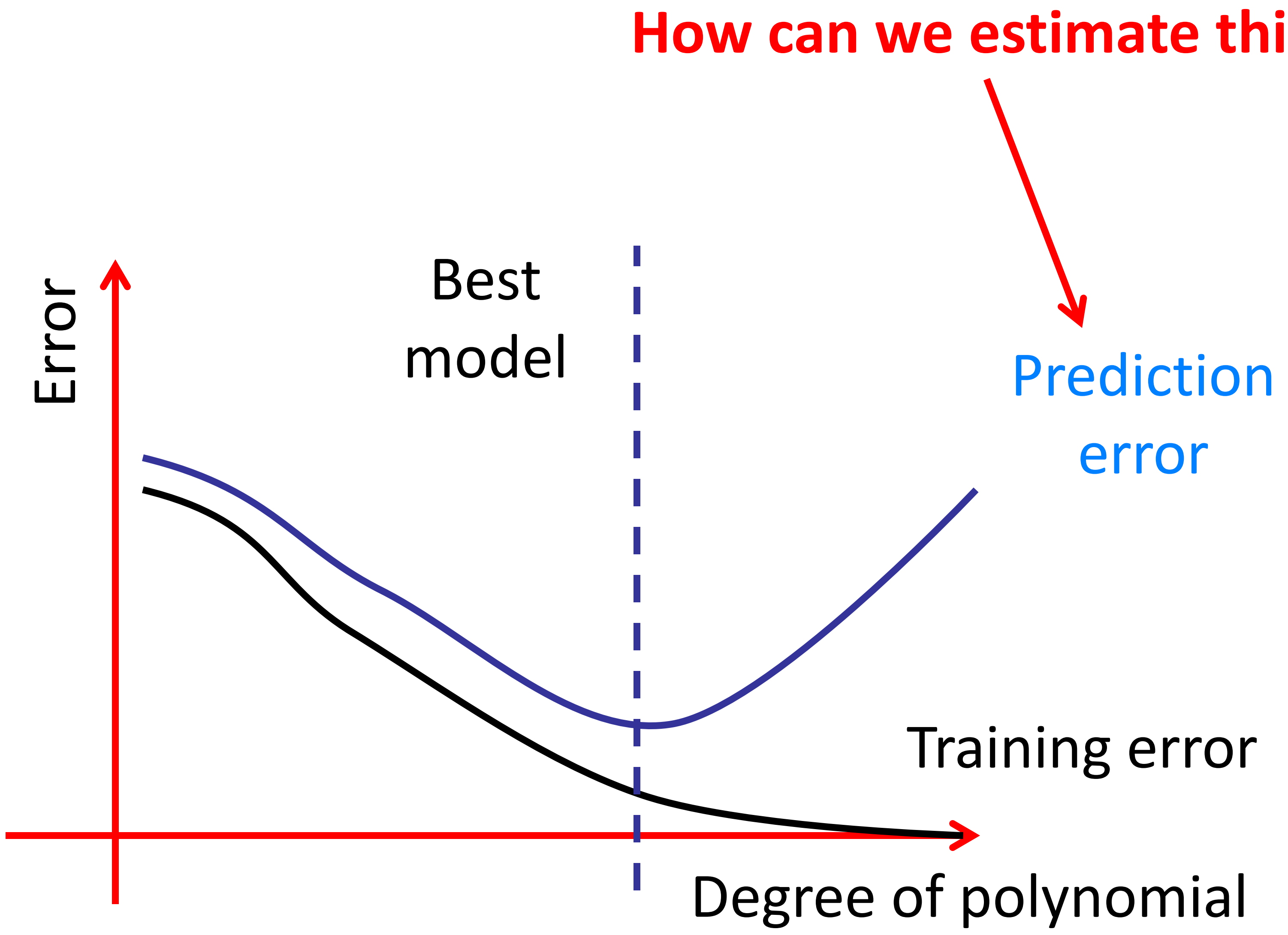
Method:

Exact solution, Gradient Descent

Evaluation metric:

Mean squared error

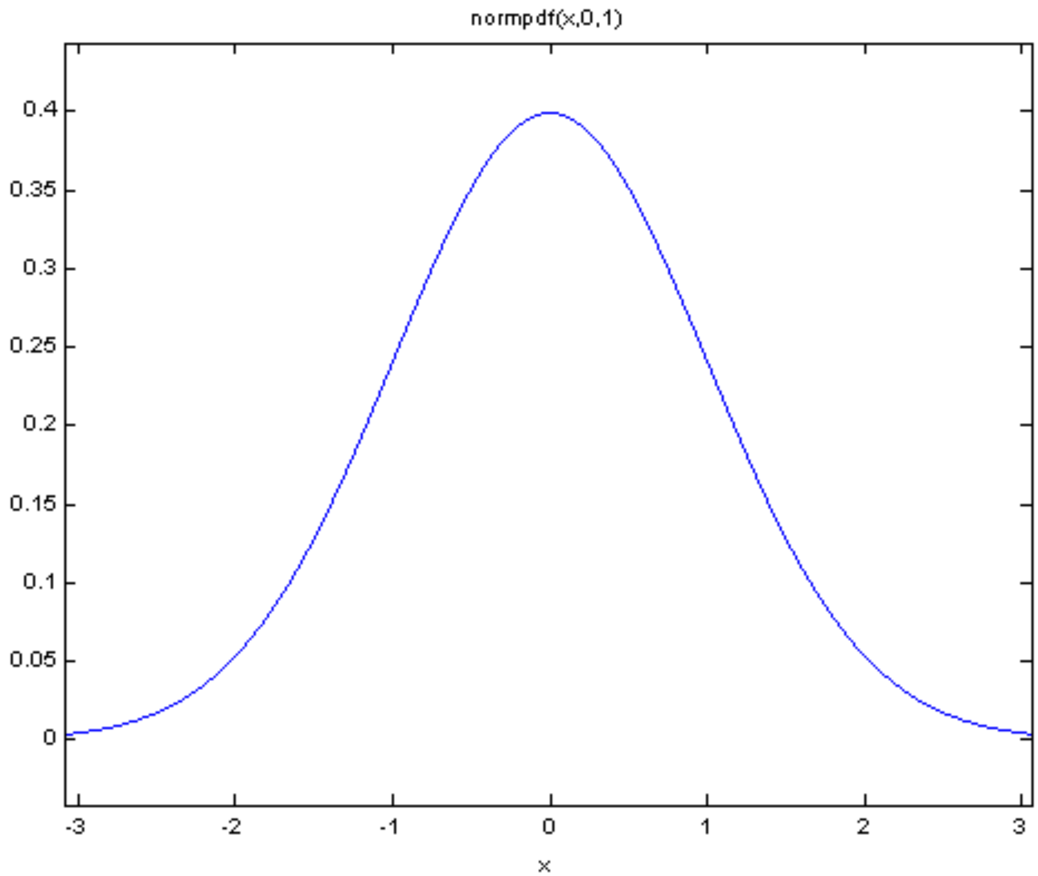
Model selection for linear regression with polynomials



Interlude: A note on probability

- You‘ll need to know about basic concepts in probability:
  - Random variables
  - Expectations (Mean, Variance etc.)
  - Independence (i.i.d. samples from a distribution, ...)
  - ...

Example: Gaussian distribution

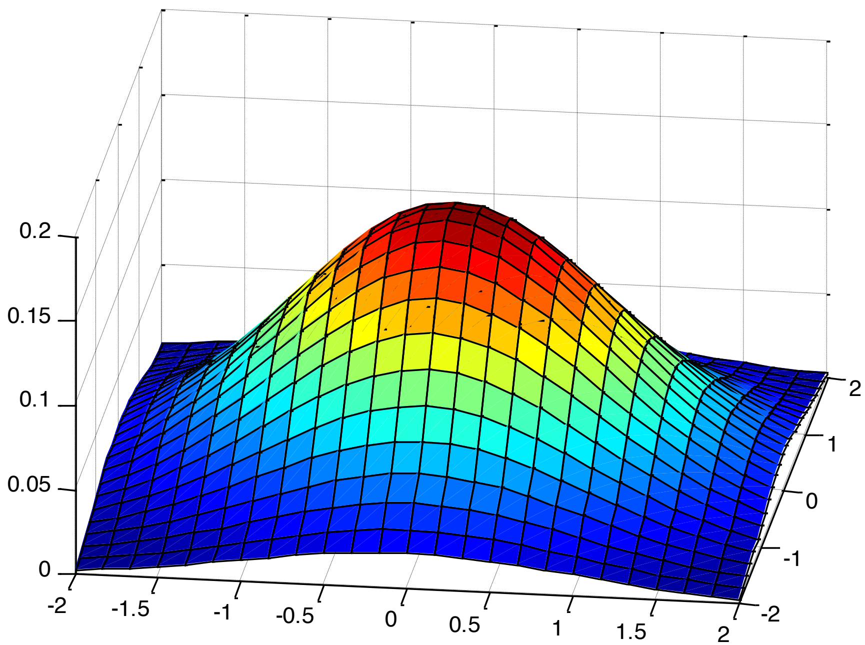


- σ = Standard deviation
- μ = mean

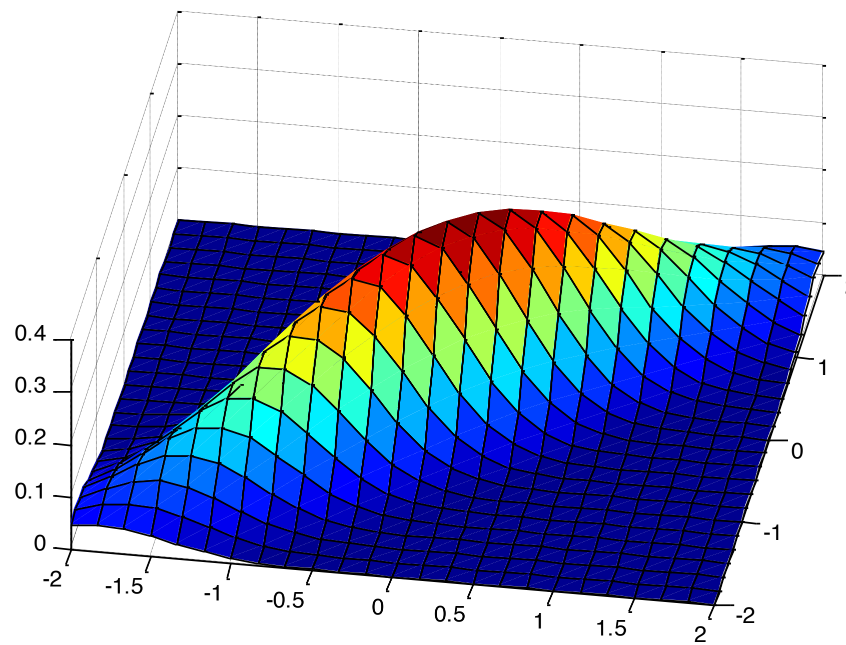
$$\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Example: Multivariate Gaussian

$$\frac{1}{2\pi\sqrt{|\Sigma|}}\exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right) \qquad \Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{pmatrix} \qquad \mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}$$



$$\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$



$$\Sigma = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}$$

Interlude: Expectations

- Expected value of random variable X

$$\mathbb{E}[X] = \begin{cases} \sum_x x\,p(x) & , X \text{ is discrete} \\ \int x\,p(x)\,dx & , X \text{ is continuous} \end{cases}$$

- Expected value of some function of X

$$\mathbb{E}[f(X)] = \begin{cases} \sum_x f(x)\,p(x) \\ \int f(x)\,p(x)\,dx \end{cases}$$

e.g.: $\mathrm{Var}[X] = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$

- Linearity of expectation X, Y RVs , $a, b \in \mathbb{R}$

$$\mathbb{E}[aX + bY] = a\,\mathbb{E}[X] + b\,\mathbb{E}[Y]$$

Achieving generalization

- Fundamental assumption: Our data set is generated independently and identically distributed (iid) from some unknown distribution *P*

$$(\mathbf{x}_i, y_i) \sim P(\mathbf{X}, Y)$$

- Our goal is to minimize the *expected error (true risk)* under *P*

$$R(\mathbf{w}) = \int P(\mathbf{x}, y)(y - \mathbf{w}^T\mathbf{x})^2 d\mathbf{x}dy$$

$$= \mathbb{E}_{\mathbf{x},y}[(y - \mathbf{w}^T\mathbf{x})^2]$$

Side note on iid assumption

- When is iid assumption invalid?
  - Time series data
  - Spatially correlated data
  - Correlated noise
  - …
- Often, can still use machine learning, but one has to be careful in interpreting results.
- Most important: Choose train/test to assess the desired generalization

Estimating the generalization error

- Estimate the true risk by the empirical risk on a sample data set D

$$\hat{R}_D(\mathbf{w}) = \frac{1}{|D|} \sum_{(\mathbf{x},y)\in D} (y - \mathbf{w}^T\mathbf{x})^2$$

- Why might this work?

Law of large numbers $\hat{R}_D(\mathbf{w}) \to R(\mathbf{w})$

for any fixed **w** almost surely as $|D| \to \infty$

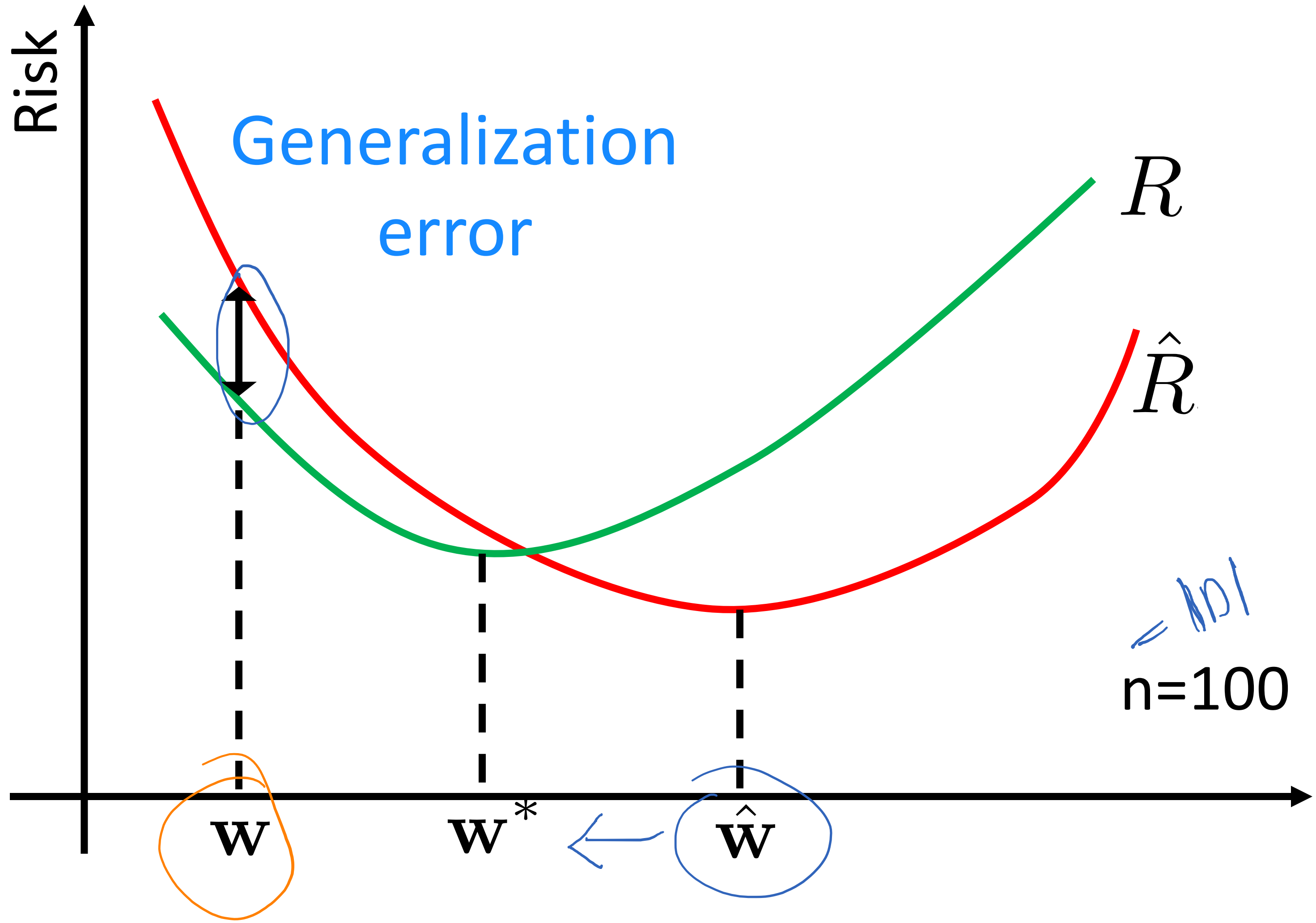
What happens if we optimize on training data?

- Suppose we are given training data D

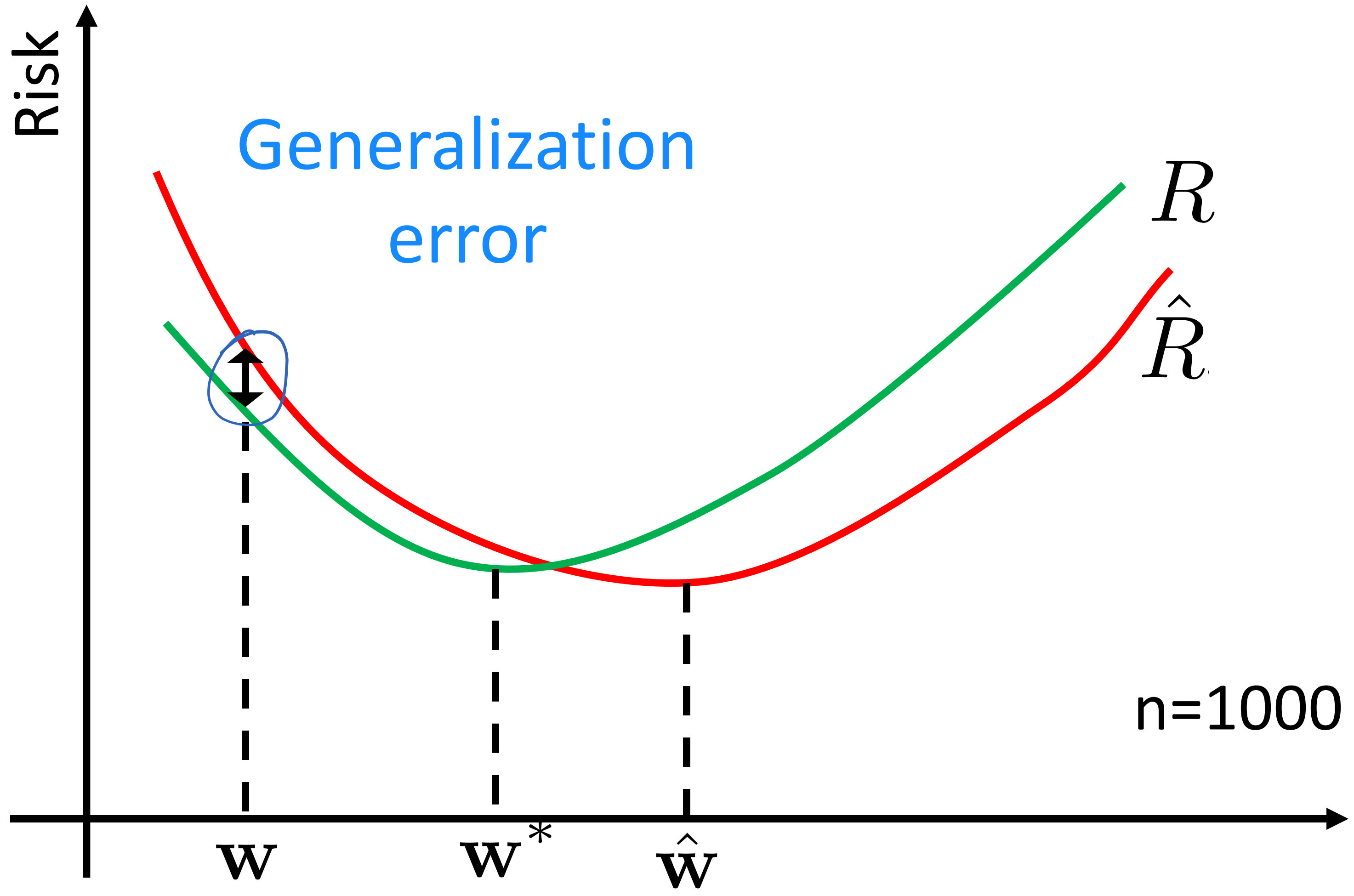
  Empirical Risk Minimization: $$\hat{\mathbf{w}}_D = \operatorname*{argmin}_{\mathbf{w}} \hat{R}_D(\mathbf{w})$$

- Ideally, we wish to solve $$\mathbf{w}^* = \operatorname*{argmin}_{\mathbf{w}} R(\mathbf{w})$$

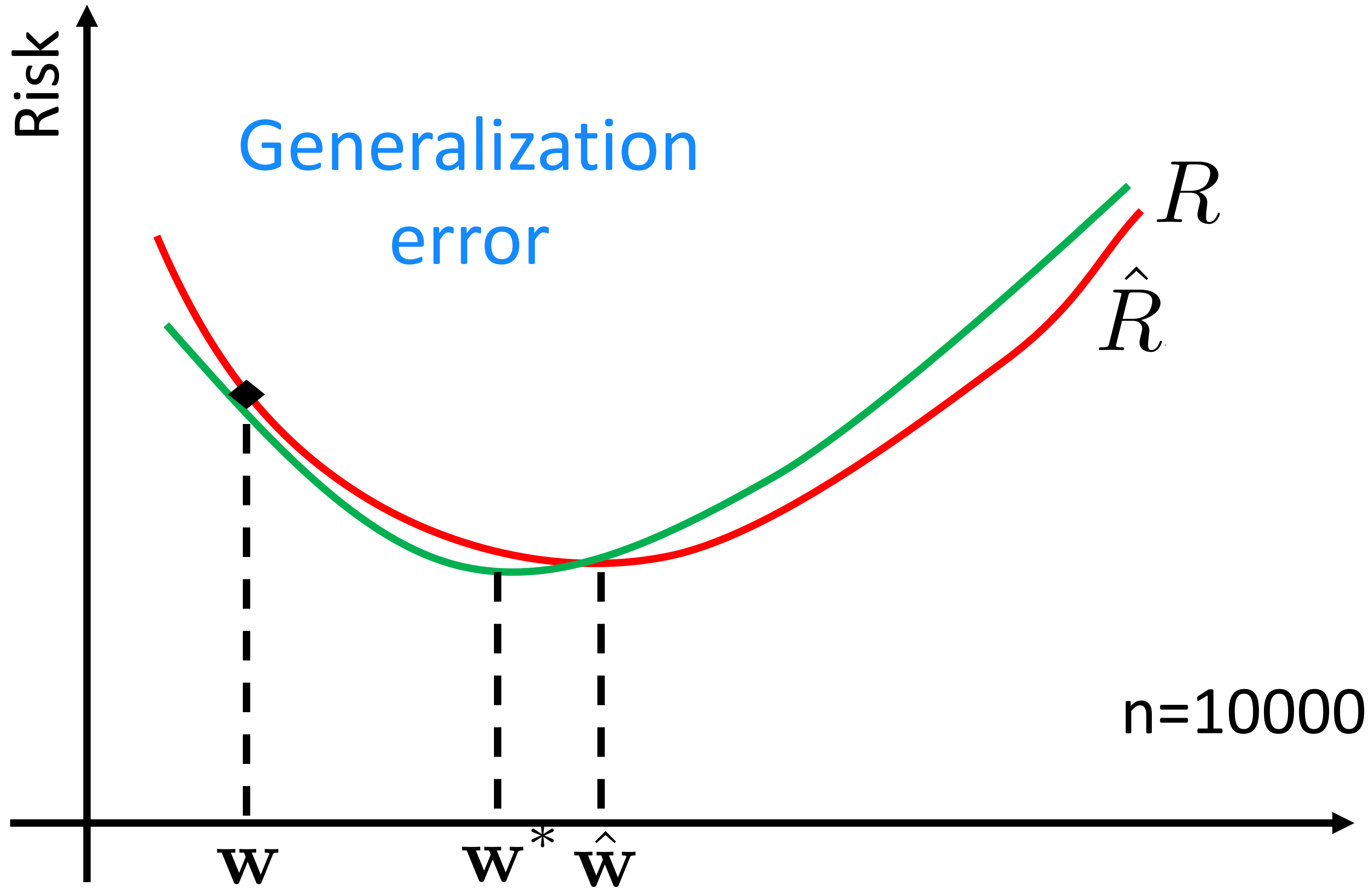
Empirical vs true risk



Empirical vs true risk



Empirical vs true risk



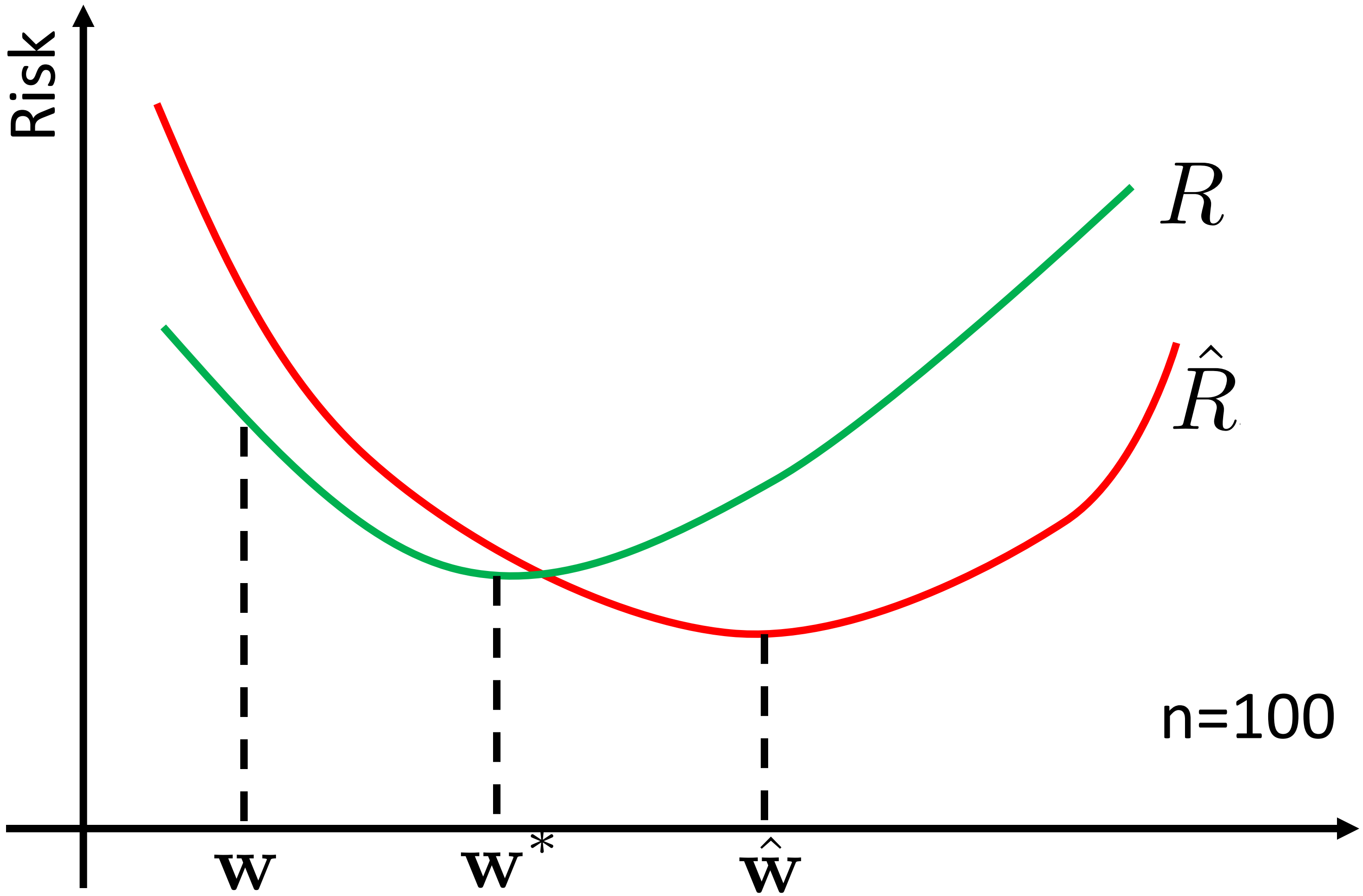
Outlook: Requirements for learning

- For learning via empirical risk minimization to be successful, need uniform convergence:

$$\sup_{\mathbf{w}} |R(\mathbf{w}) - \hat{R}_D(\mathbf{w})| \to 0 \text{ as } |D| \to \infty$$

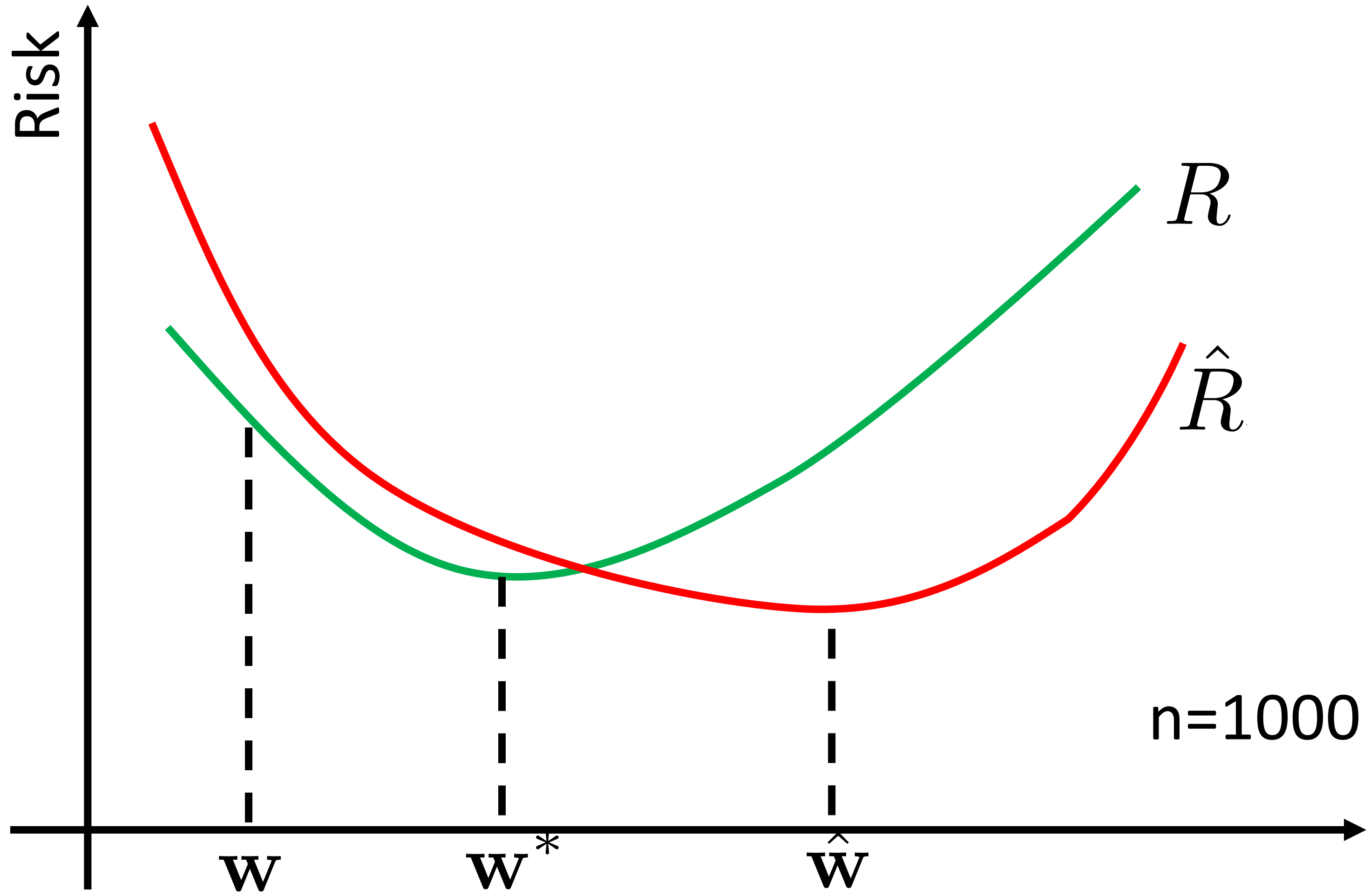
- This is not implied by law of large numbers alone, but depends on model class (holds, e.g., for squared loss on data distributions with bounded support)
  ➔ Statistical learning theory

What can go wrong in ERM

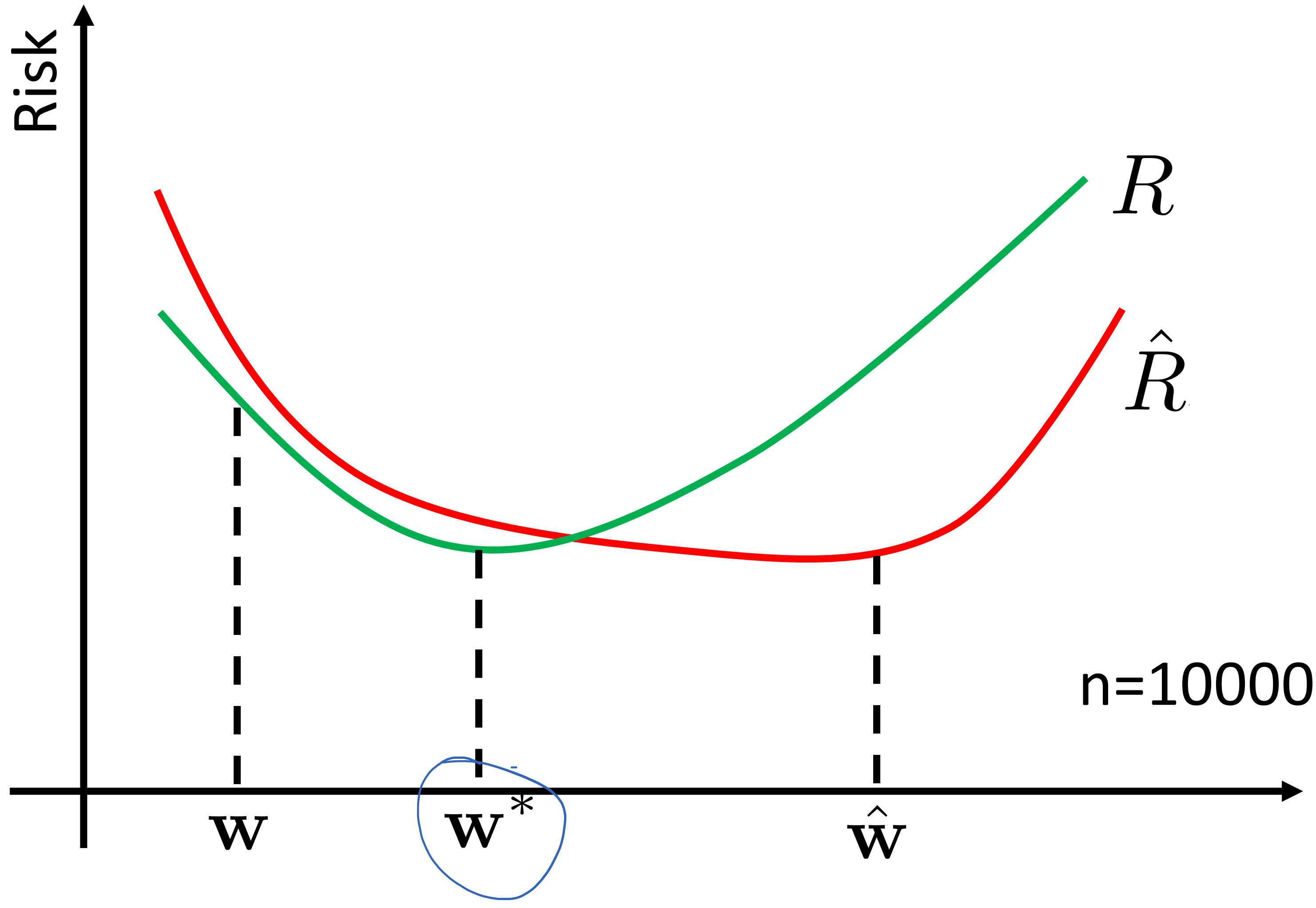


$n = |D|$

What can go wrong in ERM



What can go wrong in ERM



Learning from finite data

- Law of large numbers / uniform convergence are asymptotic statements (with $n \rightarrow \infty$)
- In practice one has finite amount of data.

- What can go wrong?

Simple example

$$\hat{\mathbf{w}}_D = \underset{\mathbf{w}}{\operatorname{argmin}}\, \hat{R}_D(\mathbf{w}) \qquad \mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}}\, R(\mathbf{w})$$

$f(x) = wx$

$P = \text{Uniform}(\{1,1\}, \{1,-1\}, \{-1,1\}, \{-1,-1\})$

$D = \{(1,1)\} \quad (x_1, y_1) = (1,1), \quad (x_1, y_1) \sim P$

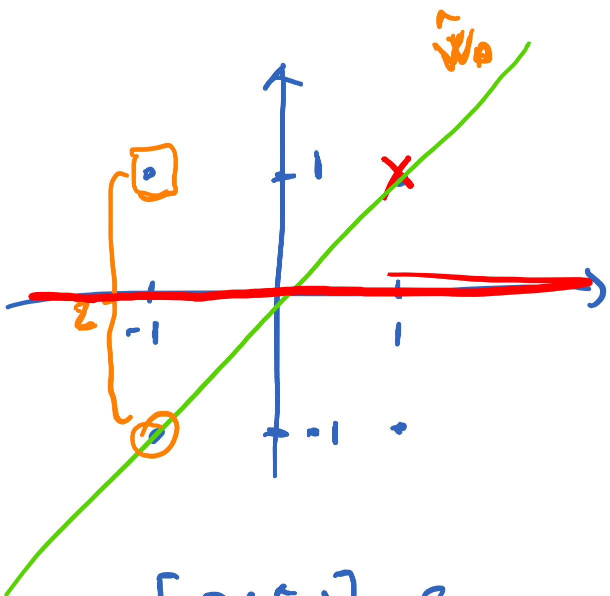
$\hat{w}_D = \underset{w}{\operatorname{argmin}}\, (y_1 - w x_1)^2 = 1$

$\hat{R}_D(\hat{w}_D) = (1 - 1 \cdot 1)^2 = 0$

$R(\hat{w}_D) = \mathbb{E}_{(x,y) \sim P}\left[(y - x)^2\right] = \frac{1}{2} \cdot 0 + \frac{1}{2} \cdot 2^2$
$= 2$

$w^* = 0$

$R(w^*) = 1 \qquad \hat{R}_D(w^*) = 1$



$\mathbb{E}_{D, |D|=1}\left[R(\hat{w}_D)\right] = 2$

$\mathbb{E}_D\left[\hat{R}(\hat{w}_D)\right] = 0$

What if we evaluate performance on training data?

$$\hat{\mathbf{w}}_D = \operatorname*{argmin}_{\mathbf{w}} \hat{R}_D(\mathbf{w}) \qquad \mathbf{w}^* = \operatorname*{argmin}_{\mathbf{w}} R(\mathbf{w})$$

0 2

- In general, it holds that $\mathbb{E}_D\left[\hat{R}_D(\hat{\mathbf{w}}_D)\right] \leq \mathbb{E}_D\left[R(\hat{\mathbf{w}}_D)\right]$

$$\mathbb{E}_D\left[\hat{R}_D(\hat{w}_D)\right] = \mathbb{E}_D\left[\min_w \hat{R}_D(w)\right] \quad \text{(ERM)}$$

$$\leq \min_w \mathbb{E}_D\left[\hat{R}_D(w)\right] \quad \text{(Jensen's ineq.)}$$

$$= \min_w \mathbb{E}_D\left[\frac{1}{|D|}\sum_{i=1}^{|D|}(y_i - wx_i)^2\right] \quad \text{(Def. } \hat{R}_D(\cdot)\text{)}$$

$$= \min_w \frac{1}{|D|}\sum_{i=1}^{|D|} \underbrace{\mathbb{E}_{(x_i,y_i)\sim P}(y_i - wx_i)^2}_{R(w)} \quad \text{(lin. exp.)}$$

$$= \min_w R(w) \leq \mathbb{E}_D\left[R(\hat{w}_D)\right]$$

- **Thus, we obtain an overly optimistic estimate!**

More realistic evaluation?

- Want to avoid underestimating the prediction error
- **Idea**: Use separate test set from the same distribution P
- Obtain independent training and test data D_{train} and D_{test}
- Optimize ***w*** on training set

$$\hat{\mathbf{w}}_{D_{train}} = \underset{\mathbf{w}}{\operatorname{argmin}} \hat{R}_{train}(\mathbf{w})$$

- Evaluate on test set

$$\hat{R}_{test}(\hat{\mathbf{w}}_{D_{train}}) = \frac{1}{|D_{test}|} \sum_{(\mathbf{x},y)\in D_{test}} (y - \hat{\mathbf{w}}^T \mathbf{x})^2$$

- Then:

$$\mathbb{E}_{D_{train},D_{test}}\left[\hat{R}_{D_{test}}(\hat{\mathbf{w}}_{D_{train}})\right] = \mathbb{E}_{D_{train}}\left[R(\hat{\mathbf{w}}_{D_{train}})\right]$$