LEUPHANA
UNIVERSITÄT LÜNEBURG

INTRODUCTION TO MACHINE LEARNING

03. Non-linear transformations and regularization
Winter 2020/2021

Acknowledgement: Many slides are adopted from Yaser Abu-Mostafa

Problem set 1

Given a linearly separable training set S and learning rate $\eta \epsilon \mathbb{R}^+$

$\boldsymbol{w}_0 \leftarrow 0;\ b_0 \leftarrow 0;\ k \leftarrow 0$

$R \leftarrow \max_{1 \le i \le l} \|x_i\|$

repeat

for $i = 1$ to l

if $y_i(\langle w_k \cdot x_i \rangle + b_k) \le 0$ then

$w_{k+1} \leftarrow w_k + \eta y_i x_i$

$b_{k+1} \leftarrow b_k + \eta y_i R^2$

$k \leftarrow k + 1$

end if

end for

Until there are no mistakes within the *for* loop

Return the list (w_k, b_k)

Agenda

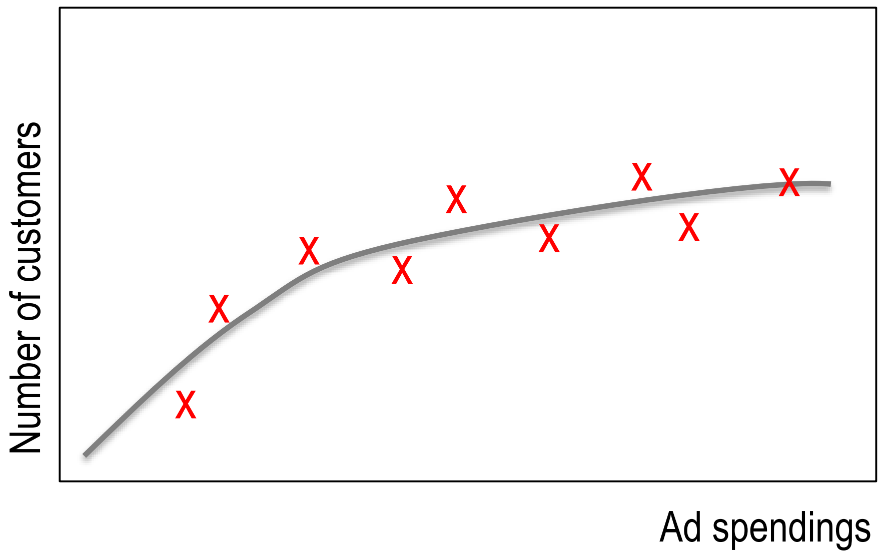
- Introduction
- Learning problem & linear classification
- Linear models: regression & logistic regression
- **Non-linear transformation, overfitting & regularization**
- Support Vector Machines and kernel learning
- Neural Networks: shallow [and deep]
- Theoretical foundation of supervised learning
- Unsupervised learning

Today's agenda

- **Non-linear transformations**
- Regularization: restricting solutions

Linear is limited

Often, the target does not linearly depend on the variables, but they are related



Again: linear in what?

Linear regression implements

$$\sum_{i=0}^{d} w_i x_i$$

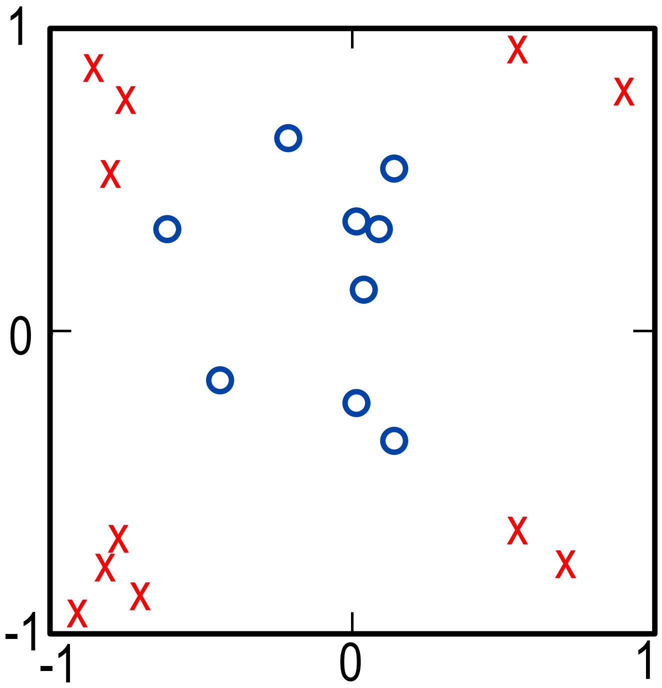
Linear classification implements

$$sign\left(\sum_{i=0}^{d} w_i x_i\right)$$

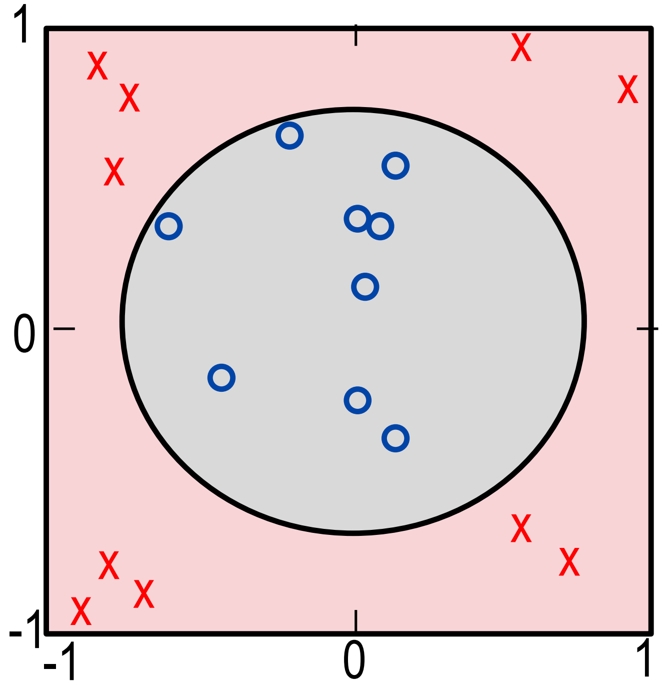
Algorithms work because of linearity in the weights

What we would like to have

Data:

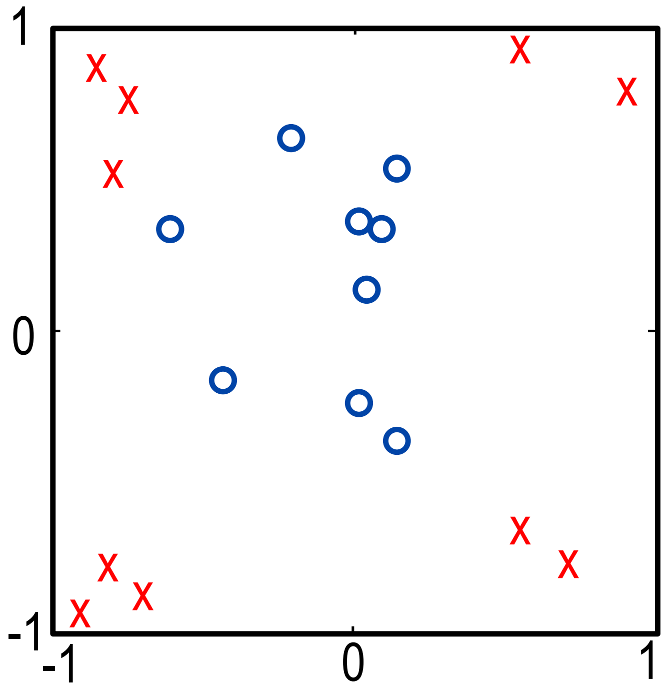


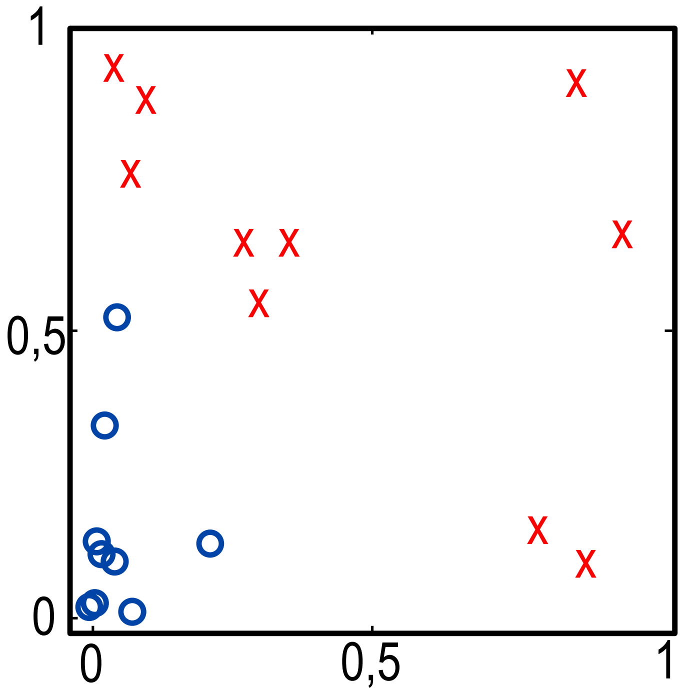
Hypothesis:



Transform the data nonlinearly

$$(x_1, x_2) \xrightarrow{\Phi} (x_1^2, x_2^2)$$





Nonlinear transforms

$$\mathbf{x} = (x_0, x_1, \dots, x_d) \quad \xrightarrow{\Phi} \quad \mathbf{z} = (z_0, z_1, \dots \dots \dots \dots \dots, z_{\tilde{d}})$$

Each $z_i = \phi_i(\mathbf{x})$ $\qquad \mathbf{z} = \Phi(\mathbf{x})$

Example: $\mathbf{z} = (1,\ x_1,\ x_2,\ x_1 x_2,\ x_1^2,\ x_2^2)$

Final hypothesis $g(\mathbf{x})$ operates on $\mathcal{X}$:

$$sign(\tilde{\mathbf{w}}^\intercal \Phi(\mathbf{x})) \quad \text{or} \quad \tilde{\mathbf{w}}^\intercal \Phi(\mathbf{x})$$

Quiz

— Let the hypothesis set be $\tilde{h} = sign(\tilde{\mathbf{w}}^\intercal \Phi(\mathbf{x}))$ with $\Phi(\mathbf{x}) = \mathbf{z} = (1, x_1^2,\ x_2^2)$

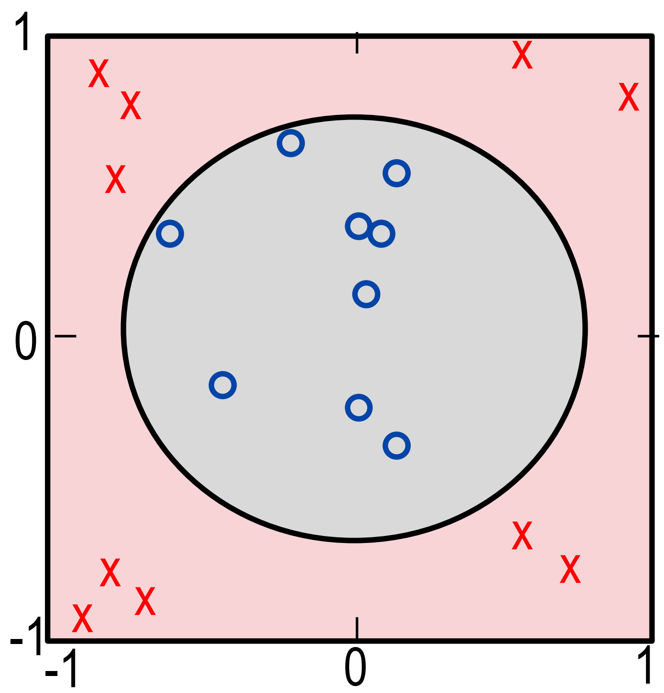
— Can you tell what geometric forms $\tilde{\mathbf{w}} = (\tilde{w}_0, \tilde{w}_1, \tilde{w}_2)$ corresponds to?

$\tilde{\mathbf{w}} = (1, -1, -1)$

$\tilde{\mathbf{w}} = (-1,1,1)$

$\tilde{\mathbf{w}} = (1, -1, -2)$

$\tilde{\mathbf{w}} = (1,1, -1)$



We will explore this in a problem set

A flexible way to transform raw data: radial basis functions

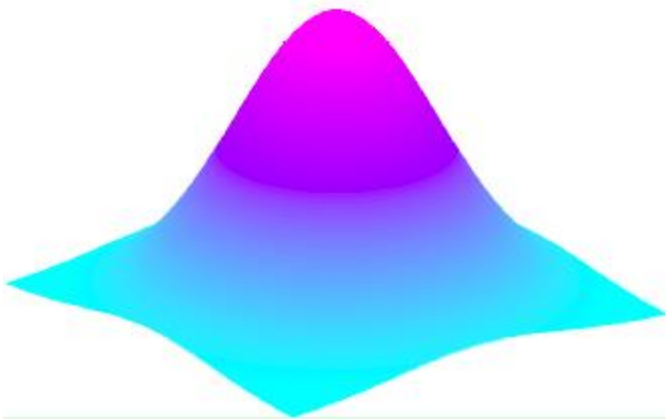
— The following transformation uses radial basis functions

$$z = \Phi(x) = \begin{pmatrix} 1 \\ \mathrm{B}(x, \mu_1, \gamma) \\ \vdots \\ \mathrm{B}(x, \mu_d, \gamma) \end{pmatrix} \text{ where } \mathrm{B}(x, \mu, \gamma) = \exp(-\gamma \|x - \mu\|^2)$$

Basic RBF model (using training data points)

Standard form (where $(x_n, y_n) \in D$ is our training data)

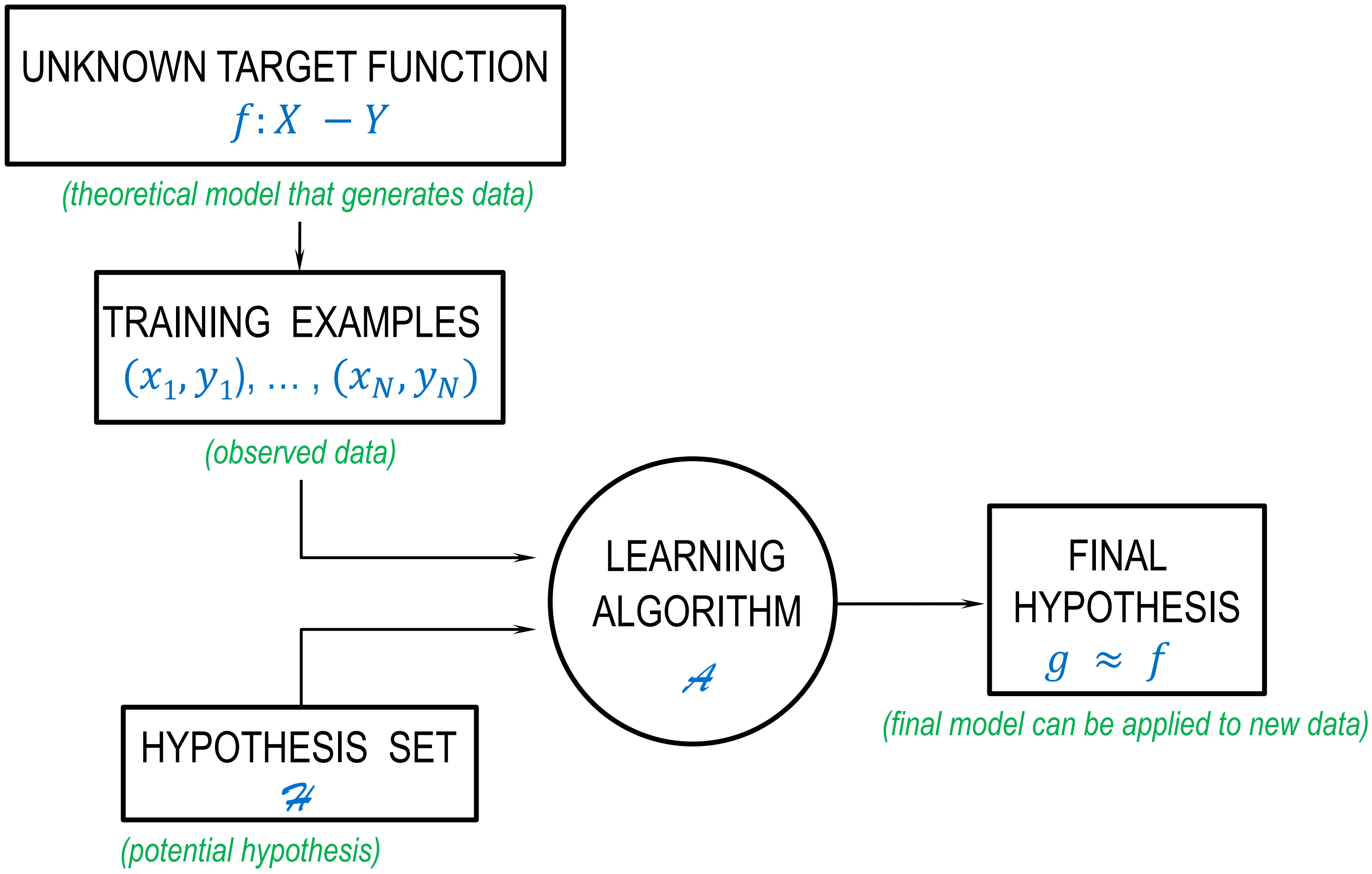
$$\mathrm{h}(\mathrm{x}) = \sum_{n=1}^{N} w_n \exp(-\gamma \|x - x_n\|^2)$$



Today’s agenda

— Non-linear transformations
— **Regularization: restricting solutions**

Review



UNKNOWN TARGET FUNCTION
$f: X \; - Y$
(theoretical model that generates data)
TRAINING EXAMPLES
$(x_1, y_1), \dots, (x_N, y_N)$
(observed data)
LEARNING ALGORITHM
$\mathcal{A}$
FINAL HYPOTHESIS
$g \approx f$
(final model can be applied to new data)
HYPOTHESIS SET
$\mathcal{H}$
(potential hypothesis)

Illustration of overfitting

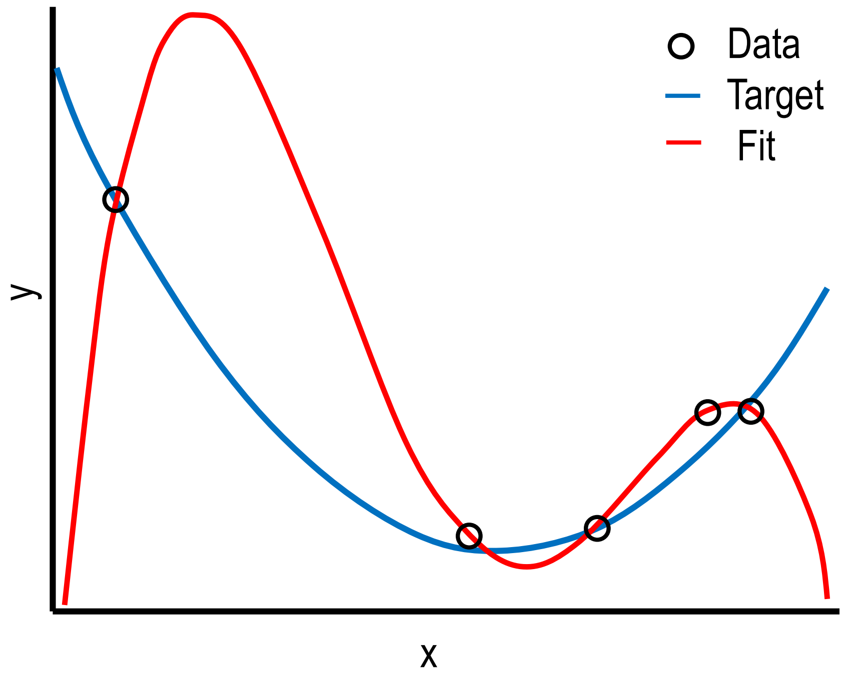
Simple target function

5 data points include a bit **noise**

4th-order polynomial fit

$E_{in} = 0$ (in-sample error)

E_{out} is huge (out-of-sample error)

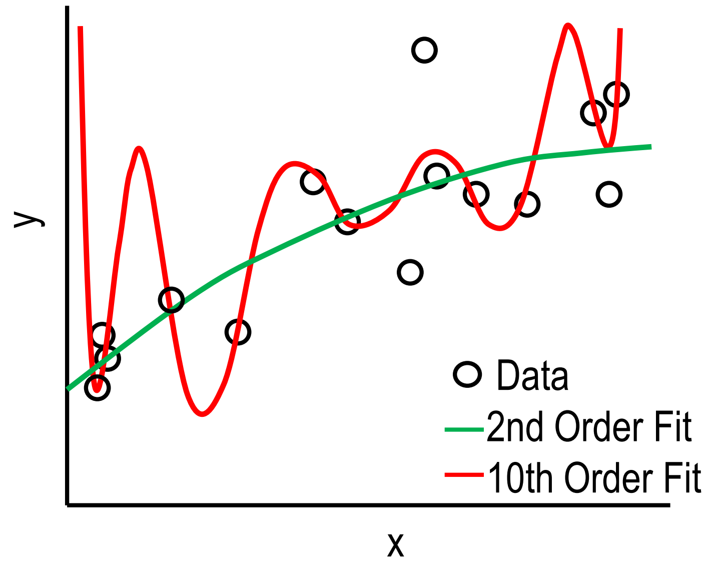


What is the optimal hypothesis set?

Two learners O (10^{th} order polynomial) and R (2^{nd} order polynomial)

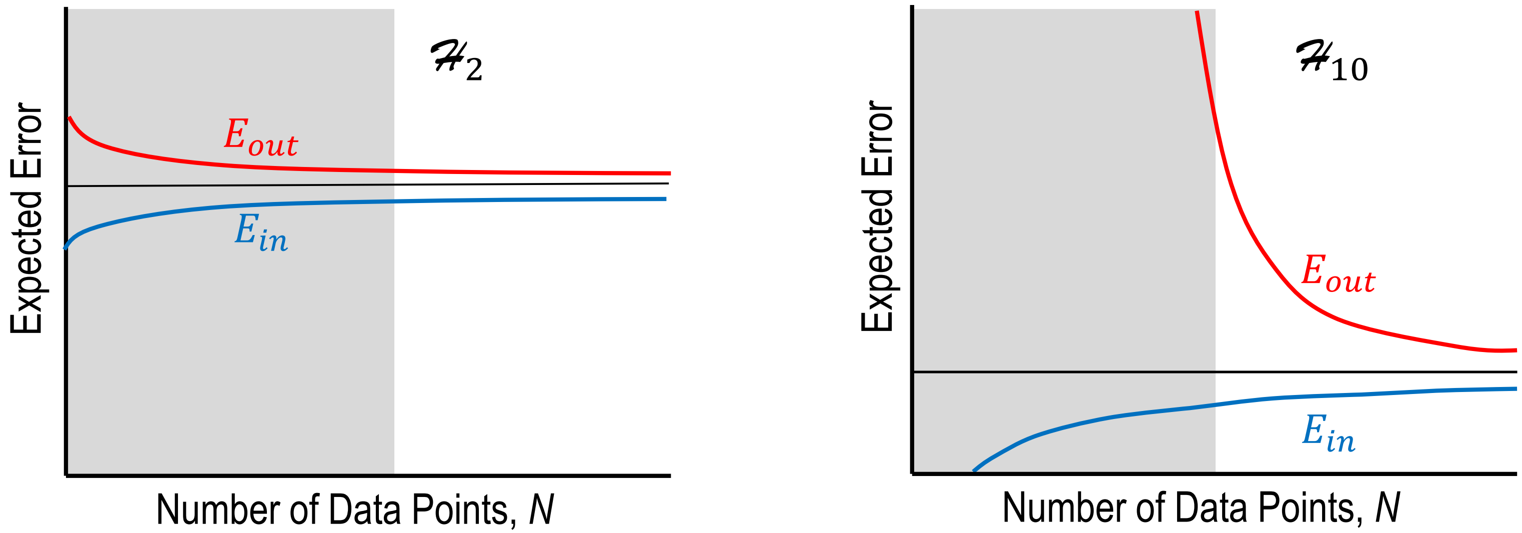
We know the target is 10^{th} order

O chooses $\mathcal{H}_{10}$ R chooses $\mathcal{H}_2$

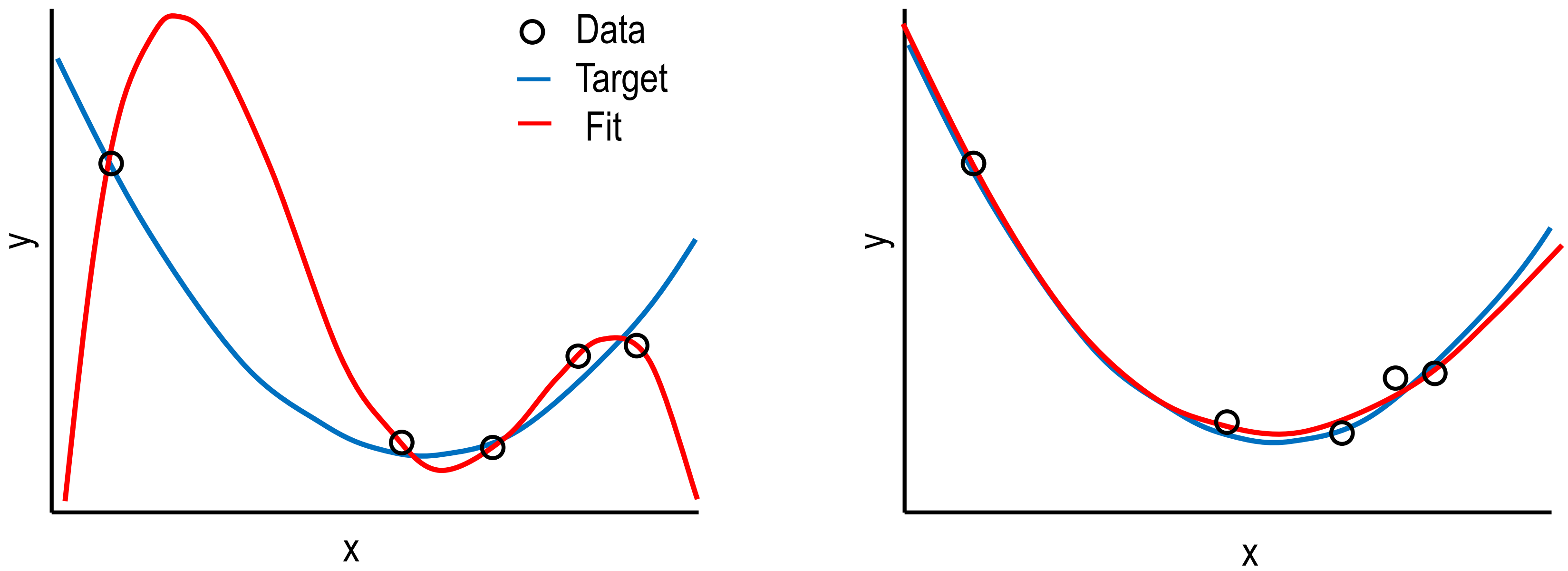


Learning a 10^{th} order target

Learning curves for both hypothesis sets



The goal of regularization



free fit

restrained fit

Unconstrained solution

Given $(x_{1,}y_1), \dots, (x_{N,}y_n) \longrightarrow (\mathbf{z}_{1,}y_1), \dots, (\mathbf{z}_{N,}y_n)$

Minimize $E_{in}(\mathrm{w}) = \frac{1}{N}\sum_{n=1}^{N}(\mathrm{w}^{\intercal}\,\mathbf{z}_n \ - y_n\ )^2$

Minimize $\frac{1}{N}(\mathrm{Z}\mathrm{w} - \mathbf{y})^{\intercal}(\mathrm{Z}\mathrm{w} - \mathbf{y})$

$\mathbf{w}_{\mathrm{lin}} = \ (Z^{\intercal}Z)^{\,-1}Z^{\intercal}\mathbf{y}$

Constraining the weights

Hard constraint: $\mathcal{H}_2$ is constrained version of $\mathcal{H}_{10}$ with $w_q = 0 \; for \; q > 2$

Softer version: $\sum_{q=0}^{Q} w_q^2 \leq C$ "**soft-order**" constraint

Minimize $\frac{1}{N}(\mathrm{Z}\mathbf{w} - \mathbf{y})^\top(\mathrm{Z}\mathbf{w} - \mathbf{y})$

subject to: $\mathbf{w}^\top\mathbf{w} \leq C$

Solution: $\mathbf{w}_{\mathrm{reg}}$ instead of $\mathbf{w}_{\mathrm{lin}}$

Solving for $\mathbf{w}_{\text{reg}}$

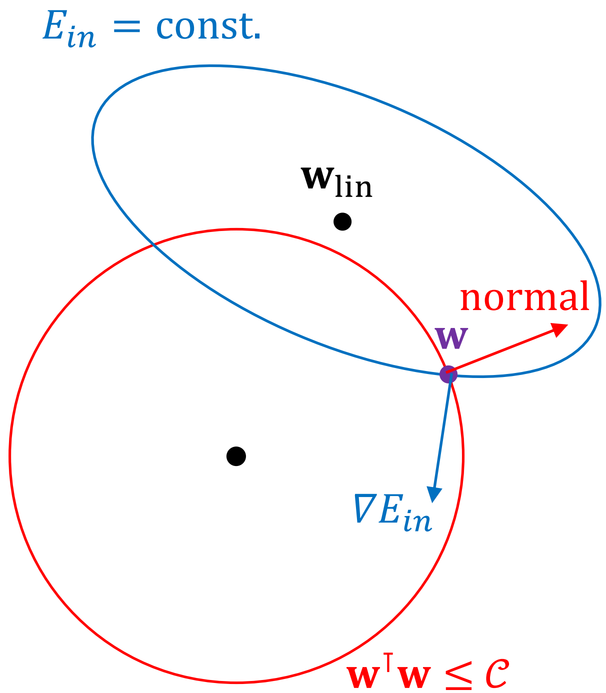
Minimize $E_{in}(\mathbf{w}) = \frac{1}{N}(\mathrm{Z}\mathrm{w} - \mathbf{y})^{\intercal}(\mathrm{Z}\mathrm{w} - \mathbf{y})$

subject to $\mathbf{w}^{\intercal}\mathbf{w} \leq C$

$$\nabla E_{in}(\mathbf{w}_{\text{reg}}) \propto -\mathbf{w}_{\text{reg}} =: -2\frac{\lambda}{N}\mathbf{w}_{\text{reg}}$$

$$\nabla E_{in}(\mathbf{w}_{\text{reg}}) + 2\frac{\lambda}{N}\mathbf{w}_{\text{reg}} = \mathbf{0}$$

Minimize $E_{in}(\mathbf{w}) + \frac{\lambda}{N}\mathbf{w}^{\intercal}\mathbf{w}$



Augmented error

Minimizing $E_{\text{aug}}(\mathbf{w}) = E_{\text{in}}(\mathbf{w}) + \frac{\lambda}{N}\mathrm{w}^{\intercal}\mathrm{w}$

$= \frac{1}{N}(\mathrm{Zw} - \mathbf{y})^{\intercal}(\mathrm{Zw} - \mathbf{y}) + \frac{\lambda}{N}\mathrm{w}^{\intercal}\mathrm{w}$ unconditionally

– solves –

Minimizing $E_{\text{in}}(\mathbf{w}) = \frac{1}{N}(\mathrm{Zw} - \mathbf{y})^{\intercal}(\mathrm{Zw} - \mathbf{y})$

subject to: $\mathrm{w}^{\intercal}\mathrm{w} \leq C$

The solution

Minimize $$E_{\text{aug}}(\mathbf{w}) = E_{\text{in}}(\mathbf{w}) + \frac{\lambda}{N}\mathrm{w}^{\intercal}\mathrm{w}$$

$$= \frac{1}{N}\left((\mathrm{Z}\mathrm{w} - \mathbf{y})^{\intercal}(\mathrm{Z}\mathrm{w} - \mathbf{y}) + \lambda \mathrm{w}^{\intercal}\mathrm{w}\right)$$

$\nabla E_{\text{aug}}(\mathbf{w}) = 0 \quad \Longrightarrow \quad \mathrm{Z}^{\intercal}(\mathrm{Z}\mathrm{w} - \mathbf{y}) + 2\lambda \mathrm{w} = 0$ (typically we drop the 2)

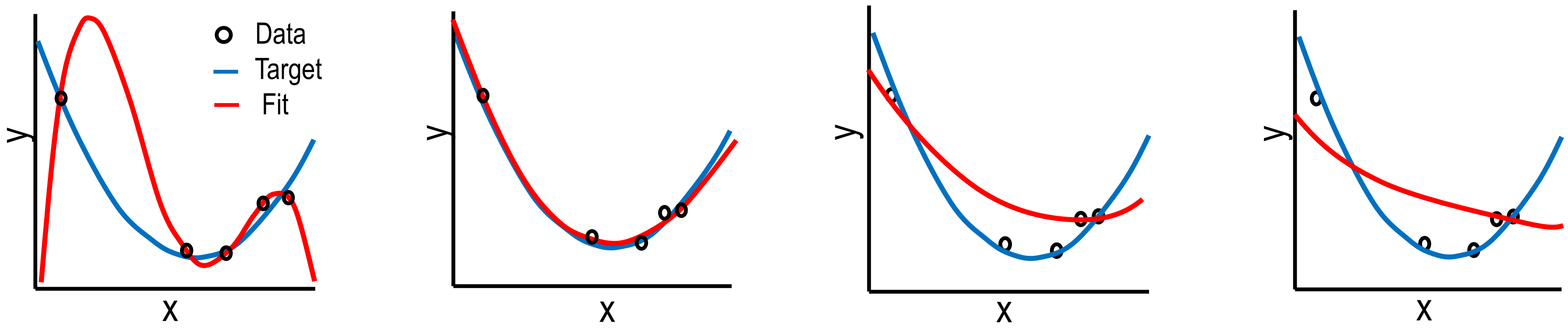
$$\boxed{\mathbf{w}_{\text{reg}} = (\mathrm{Z}^{\intercal}\mathrm{Z} + \lambda \mathrm{I})^{-1}\mathrm{Z}^{\intercal}\mathbf{y}}$$ (with regularization)

as opposed to $$\mathbf{w}_{\text{lin}} = (\mathrm{Z}^{\intercal}\mathrm{Z})^{-1}\mathrm{Z}^{\intercal}\mathbf{y}$$ (without regularization)

The result

Minimizing $E_{\text{aug}}(\mathbf{w}) = E_{\text{in}}(\mathbf{w}) + \frac{\lambda}{N}\mathrm{w}^{\mathsf{T}}\mathrm{w}$ for different λ's:

$\lambda = 0$ $\lambda = 0.0001$ $\lambda = 0.01$ $\lambda = 1$



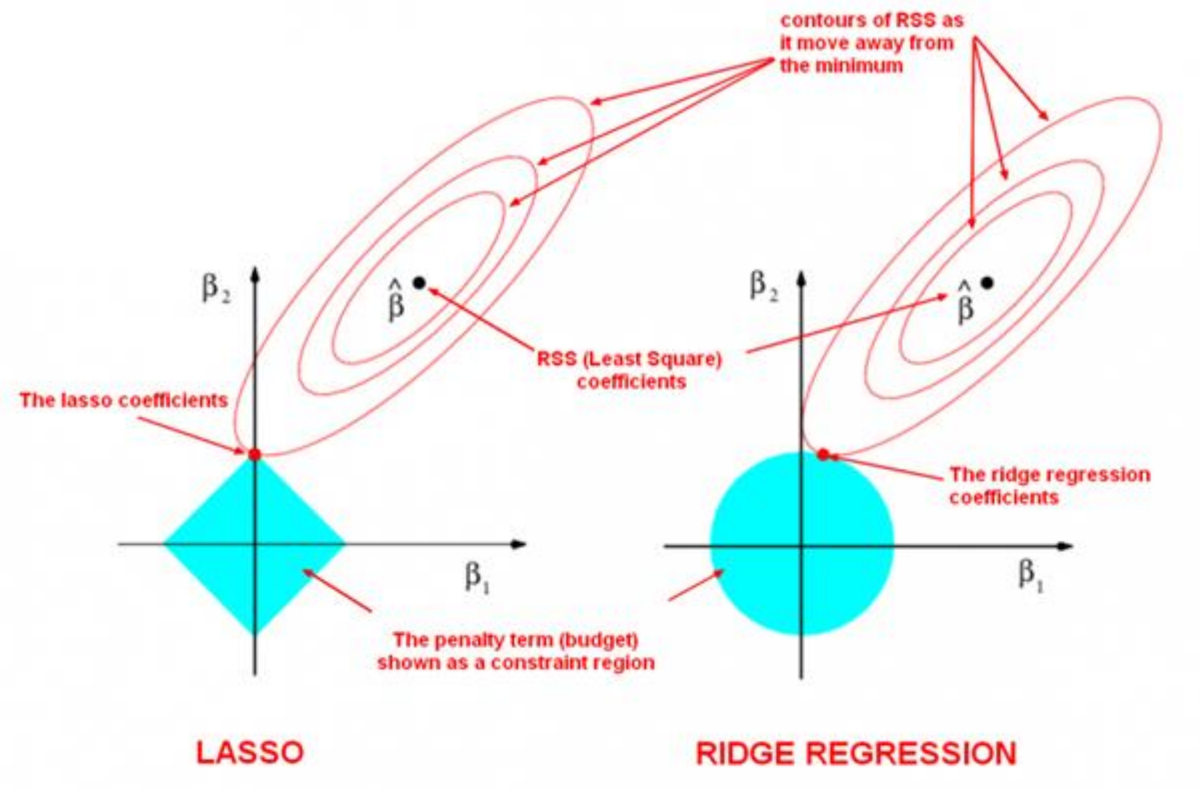
overfitting $\longrightarrow$ $\longrightarrow$ $\longrightarrow$ $\longrightarrow$ underfitting

Another exciting regularizer

— In LASSO regression we use

$$E_{\text{in}}(\mathbf{w}) + \frac{\lambda}{N}|\mathrm{w}|$$

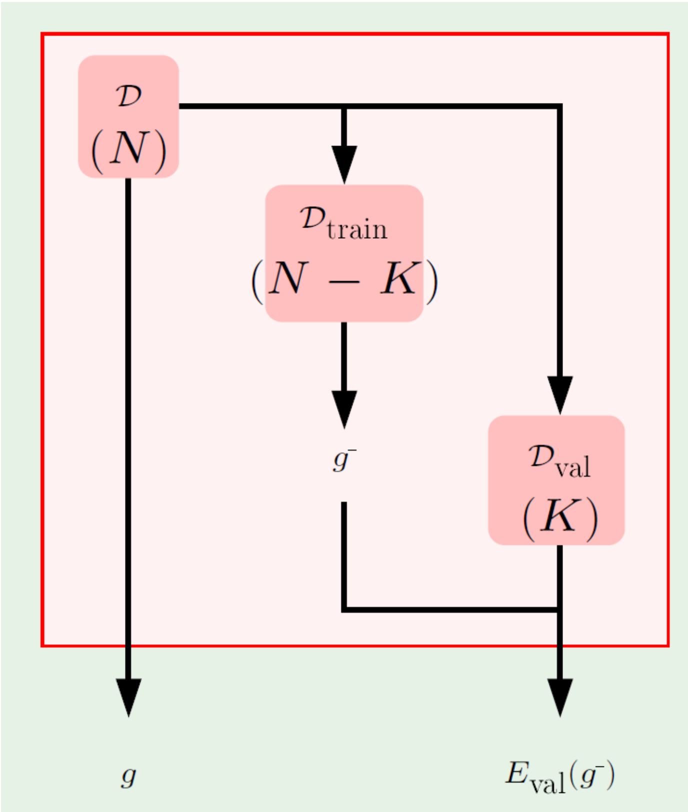
— LASSO regression can be used for variable selection



Source: https://www.quora.com/How-would-you-describe-the-difference-between-linear-regression-lasso-regression-and-ridge-regression

Finding the best λ

- We want to estimate/minimize E_{out} - we can do so by using a separate dataset that has not been used for training
- This approach is called validation
- Now we can minimize E_{val} by finding the best λ



Additional material

The learning algorithm

— Finding $w_1, \ldots, w_n$ for

$$\mathrm{h(x)} = \sum_{n=1}^{N} w_n \exp(-\gamma \|x - x_n\|^2)$$

— Based on $\mathrm{D} = (x_1, y_1), \ldots, (x_n, y_n)$

— Can we choose $w_1, \ldots, w_n$ such that $E_{in} = 0$ or $h(x_n) = y_n$?

— We need to solve $y_n = \sum_{m=1}^{N} w_m \exp(-\gamma \|x_n - x_m\|^2)$

Solution

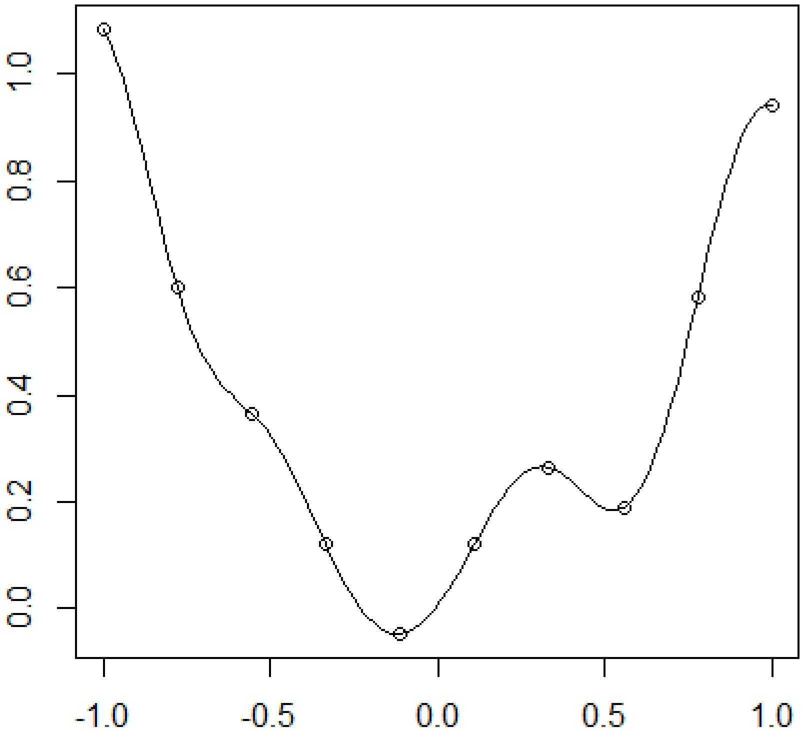
— $y_n = \sum_{m=1}^{N} w_m \exp(-\gamma \|x_n - x_m\|^2)$ gives us N equations with N unknowns

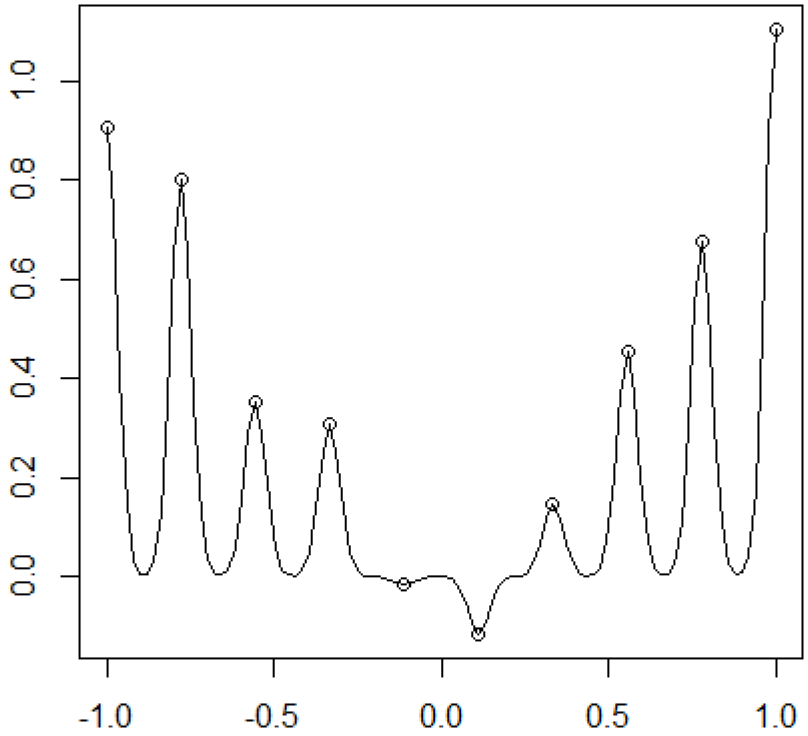
$$\underbrace{\begin{bmatrix} \exp(-\gamma \|x_1 - x_1\|^2) & \cdots & \exp(-\gamma \|x_1 - x_N\|^2) \\ \exp(-\gamma \|x_2 - x_1\|^2) & \ldots & \exp(-\gamma \|x_2 - x_N\|^2 \\ \vdots & & \vdots \\ \exp(-\gamma \|x_N - x_1\|^2) & & \exp(-\gamma \|x_N - x_N\|^2 \end{bmatrix}}_{\Phi} \begin{bmatrix} w_1 \\ w_2 \\ \\ w_N \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \\ y_N \end{bmatrix}$$

— If Φ is invertible, then $\boldsymbol{w} = \boldsymbol{\Phi}^{-1}\boldsymbol{y}$

The impact of γ

— Large or small γ?





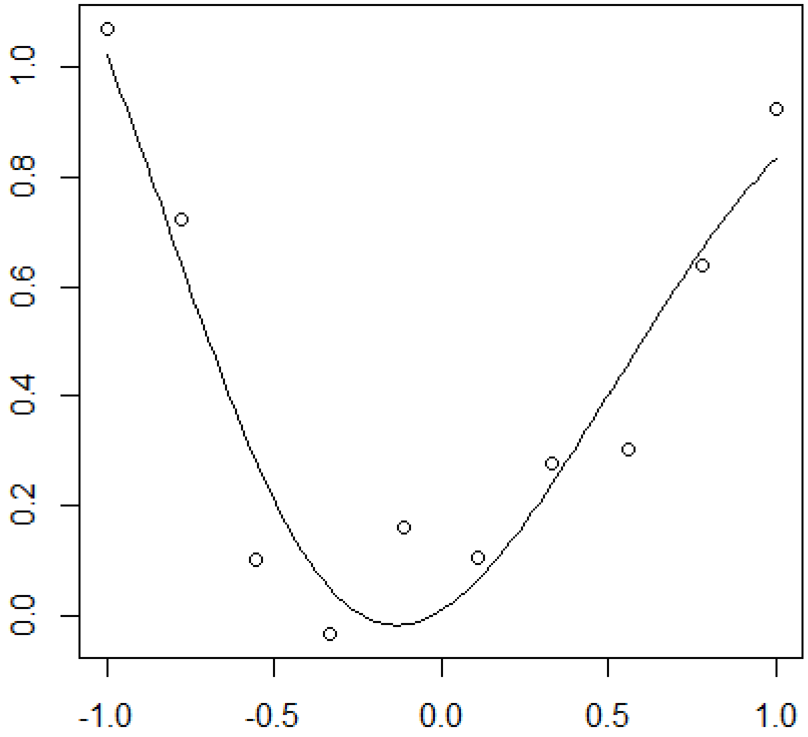
What about generalization?

- So far, we have fitted the training dataset perfectly → hypothesis is not likely to generalize

- Solution: Regularization – add penalty term:

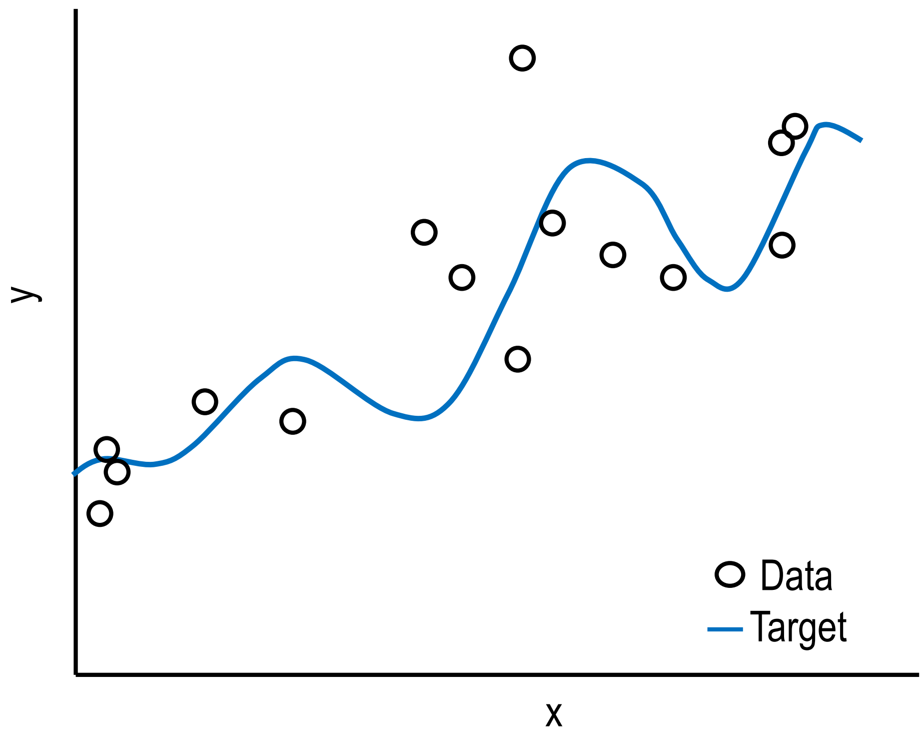
$$\boldsymbol{w}_{ridge} = (\Phi^T\Phi + \boldsymbol{\delta^2 I})^{-1}\Phi^T\boldsymbol{y}$$

- This will be the topic of the lecture on overfitting and regularization



A detailed experiment to understand overfitting

Impact of noise level and target complexity



$$y = f(x) + \underbrace{\epsilon(x)}_{\sigma^2} = \sum_{q=0}^{Q_f} a_q\, x^q + \epsilon(x)$$

noise level: σ^2

target complexity: Q_f

Data set size: N

Legendre polynomials

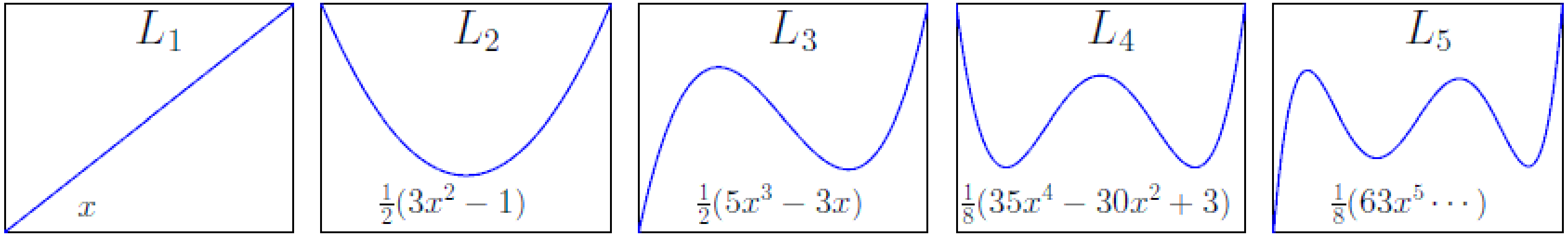
$\mathcal{H}_Q$: polynomials of Order Q

$$z = \begin{bmatrix} 1 \\ L_1(\mathrm{x}) \\ \vdots \\ L_Q(\mathrm{x}) \end{bmatrix}$$

linear regression in $\mathcal{Z}$ space

$$\mathcal{H}_Q = \left\{ \sum_{q=0}^{Q} w_q L_q(\mathrm{x}) \right\}$$

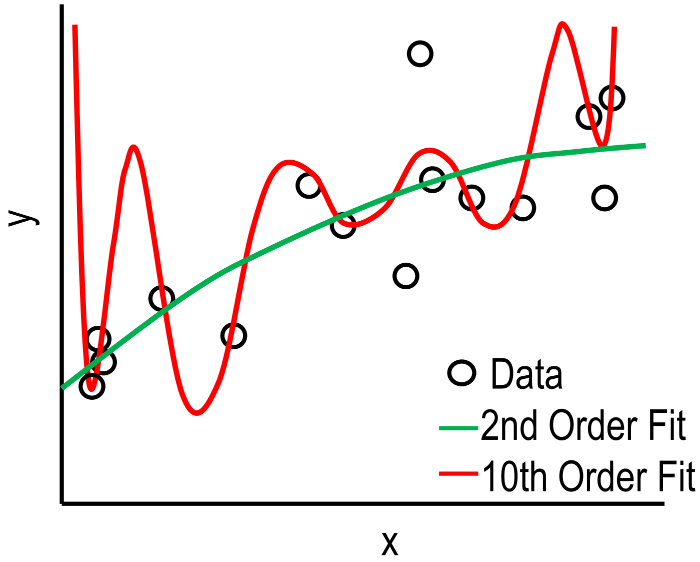
Legendre polynomials:



The overfit measure

We fit the data set $(x_1, y_1), \ldots, (x_N, y_N)$ using our two models:

$\mathcal{H}_2$: 2nd-order polynomials



$\mathcal{H}_{10}$: 10th –order polynomials

Compare out-of-sample errors of

$g_2 \in \mathcal{H}_2$ and $g_{10} \in \mathcal{H}_{10}$

overfit measure: $E_{out}(g_{10}) - E_{out}(g_2)$

Weight ‘decay‘

Minimizing $E_{\text{in}}(\mathbf{w}) + \frac{\lambda}{N}\mathrm{w}^{\mathsf{T}}\mathrm{w}$ is called weight *decay*. Why?

Gradient descent:

$$\mathbf{w}(t+1) = \mathbf{w}(t) - \eta\,\nabla E_{\text{in}}\big(\mathbf{w}(t)\big) - 2\eta\frac{\lambda}{N}\mathbf{w}(t)$$

$$= \mathbf{w}(t)\left(1 - \frac{2\eta\lambda}{N}\right) - \eta\,\nabla E_{\text{in}}\big(\mathbf{w}(t)\big)$$