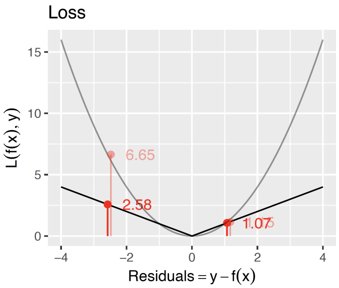
Einführung in das statistische Lernen

Loss Functions for Regression

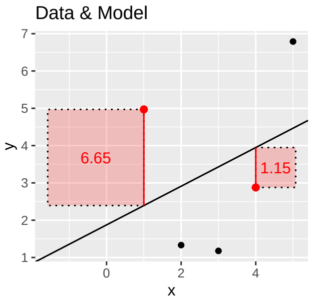


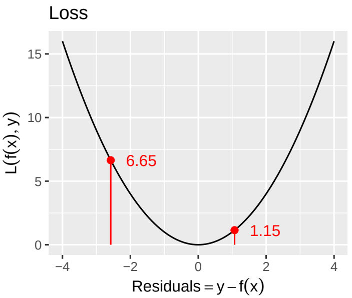
Learning goals

- Know definitions of L1 and L2 loss
- Understand difference between L1 and L2 loss
- Understand why optimization for L1 loss is harder than for L2 loss

REGRESSION LOSSES - L2 / SQUARED ERROR

- $L\left(y, f(\mathbf{x})\right) = (y - f(\mathbf{x}))^2$ or $L\left(y, f(\mathbf{x})\right) = 0.5(y - f(\mathbf{x}))^2$
- Convex
- Differentiable, gradient no problem in loss minimization
- For latter: $\frac{\partial 0.5(y - f(\mathbf{x}))^2}{\partial f(\mathbf{x})} = y - f(\mathbf{x}) = \epsilon$, derivative is residual
- Tries to reduce large residuals (if residual is twice as large, loss is 4 times as large), hence outliers in y can become problematic
- Connection to Gaussian distribution (see later)





REGRESSION LOSSES - L2 / SQUARED ERROR

What's the optimal constant prediction c (i.e. the same $\hat{y}$ for all $\mathbf{x}$)?

$$L\left(y, f(\mathbf{x})\right) = (y - f(\mathbf{x}))^2 = (y - c)^2$$

We search for the c that minimizes the empirical risk.

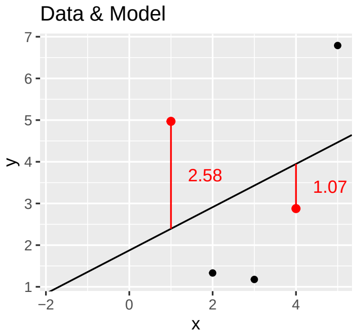
$$\hat{c} = \arg\min_{c \in \mathbb{R}} \mathcal{R}_{\text{emp}}(c) = \arg\min_{c \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^{n} (y^{(i)} - c)^2$$

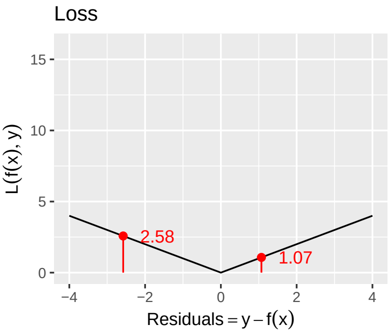
We set the derivative of the empirical risk to zero and solve for c:

$$\begin{aligned} -\frac{1}{n} \sum_{i=1}^{n} 2(y^{(i)} - c) &= 0 \\ \hat{c} &= \frac{1}{n} \sum_{i=1}^{n} y^{(i)} \end{aligned}$$

REGRESSION LOSSES - L1 / ABSOLUTE ERROR

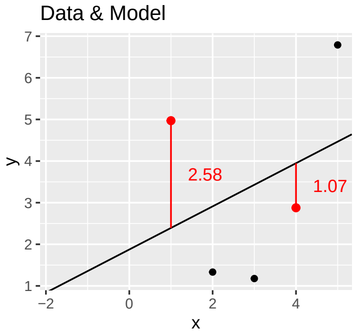
- $L(y, f(\mathbf{x})) = |y - f(\mathbf{x})|$
- Convex
- No derivatives for $= 0$, $y = f(\mathbf{x})$, optimization becomes harder
- $\hat{f}(\mathbf{x}) =$ median of $y|\mathbf{x}$

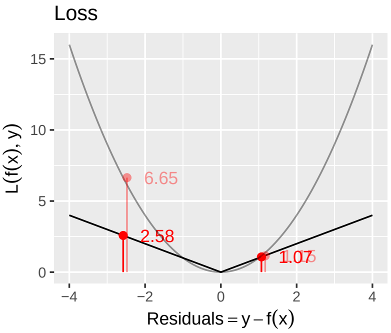




REGRESSION LOSSES - L1 / ABSOLUTE ERROR

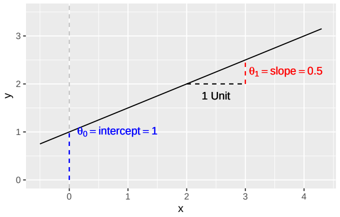
- $L(y, f(\mathbf{x})) = |y - f(\mathbf{x})|$
- Convex
- No derivatives for $\epsilon = 0$, $y = f(\mathbf{x})$, optimization becomes harder
- $\hat{f}(\mathbf{x}) =$ median of $y|\mathbf{x}$
- More robust, outliers in y are less influential than for L2





Einführung in das statistische Lernen

Linear Regression Models



Learning goals

- Know the hypothesis space of the linear model
- Understand the risk function that follows with L2 loss
- Understand how optimization works for the linear model
- Understand how outliers affect the estimated model differently when using L1 or L2 loss

LINEAR REGRESSION: HYPOTHESIS SPACE

We want to predict a numerical target variable by a *linear transformation* of the features $\mathbf{x} \in \mathbb{R}^p$.

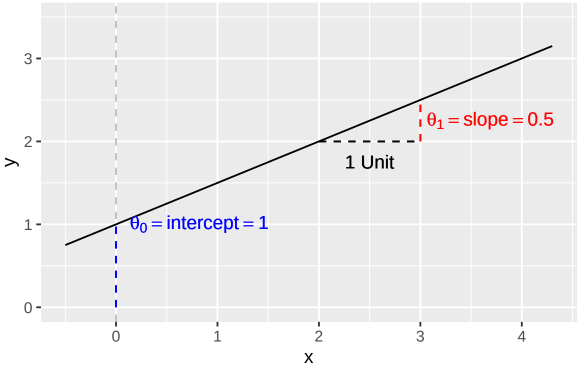
So with $\boldsymbol{\theta} \in \mathbb{R}^p$ this mapping can be written as:

$$\begin{aligned} y = f(\mathbf{x}) &= \theta_0 + \boldsymbol{\theta}^T \mathbf{x} \\ &= \theta_0 + \theta_1 x_1 + \cdots + \theta_p x_p \end{aligned}$$

This defines the hypothesis space $\mathcal{H}$ as the set of all linear functions in $\boldsymbol{\theta}$:

$$\mathcal{H} = \{\theta_0 + \boldsymbol{\theta}^T \mathbf{x} \mid (\theta_0, \boldsymbol{\theta}) \in \mathbb{R}^{p+1}\}$$

LINEAR REGRESSION: HYPOTHESIS SPACE



$$y = \theta_0 + \theta_1 \cdot x$$

LINEAR REGRESSION: HYPOTHESIS SPACE

Given observed labeled data $\mathcal{D}$, how to find $(\theta_0, \boldsymbol{\theta})$?

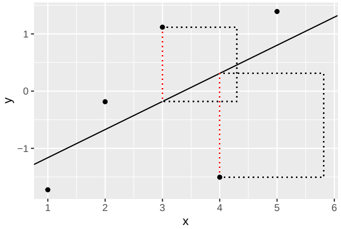
This is **learning** or parameter estimation, the learner does exactly this by **empirical risk minimization**.

NB: We assume from now on that θ_0 is included in $\boldsymbol{\theta}$.

LINEAR REGRESSION: RISK

We could measure training error as the sum of squared prediction errors (SSE). This is the risk that corresponds to **L2 loss**:

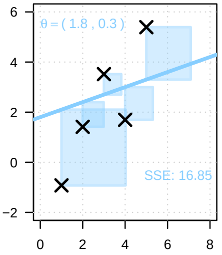
$$\mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) = \text{SSE}(\boldsymbol{\theta}) = \sum_{i=1}^{n} L\left(y^{(i)}, f\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)\right) = \sum_{i=1}^{n} \left(y^{(i)} - \boldsymbol{\theta}^T \mathbf{x}^{(i)}\right)^2$$



Minimizing the squared error is computationally much simpler than minimizing the absolute differences (**L1 loss**).

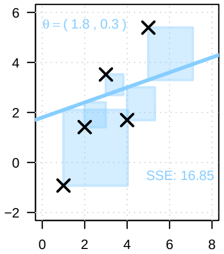
LINEAR MODEL: OPTIMIZATION

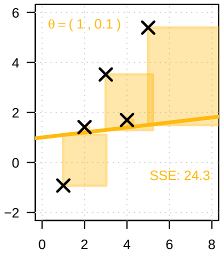
We want to find the parameters $\boldsymbol{\theta}$ of the linear model, i.e., an element of the hypothesis space $\mathcal{H}$ that fits the data optimally.
So we evaluate different candidates for $\boldsymbol{\theta}$.
A first (random) try yields a rather large SSE: (**Evaluation**).



LINEAR MODEL: OPTIMIZATION

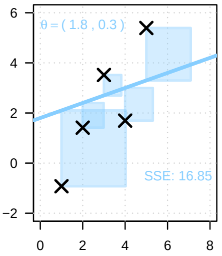
We want to find the parameters θ of the linear model, i.e., an element of the hypothesis space $\mathcal{H}$ that fits the data optimally.
So we evaluate different candidates for θ.
Another line yields an even bigger SSE (**Evaluation**). Therefore, this one is even worse in terms of empirical risk.

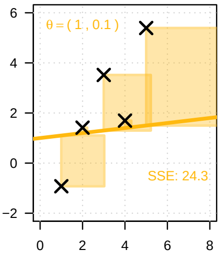


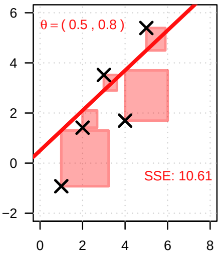


LINEAR MODEL: OPTIMIZATION

We want to find the parameters θ of the linear model, i.e., an element of the hypothesis space $\mathcal{H}$ that fits the data optimally.
So we evaluate different candidates for θ.
Another line yields an even bigger SSE (**Evaluation**). Therefore, this one is even worse in terms of empirical risk. Let's try again:

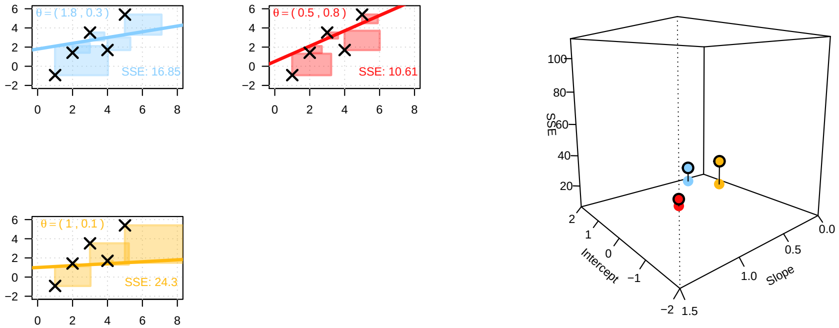






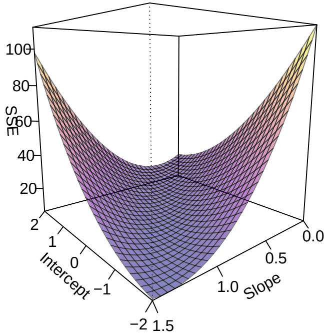
LINEAR MODEL: OPTIMIZATION

Since every $\boldsymbol{\theta}$ results in a specific value of $\mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$, and we try to find $\arg\min_{\boldsymbol{\theta}} \mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$, let's look at what we have so far:



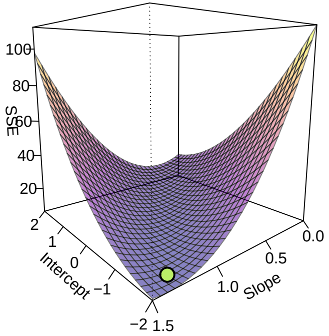
LINEAR MODEL: OPTIMIZATION

Instead of guessing, we use **optimization** to find the best $\boldsymbol{\theta}$:



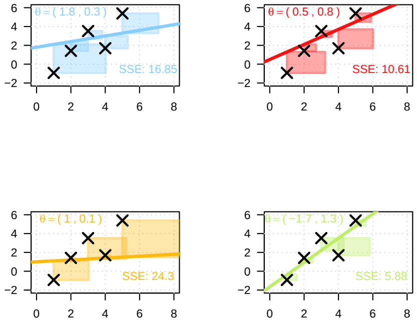
LINEAR MODEL: OPTIMIZATION

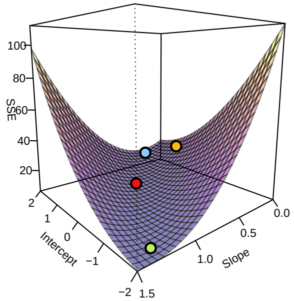
Instead of guessing, we use **optimization** to find the best $\boldsymbol{\theta}$:



LINEAR MODEL: OPTIMIZATION

Instead of guessing, we use **optimization** to find the best $\boldsymbol{\theta}$:





LINEAR MODEL: OPTIMIZATION

For L2 regression, we can find this optimal value analytically:

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \arg\min_{\boldsymbol{\theta}} \mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \left(y^{(i)} - \boldsymbol{\theta}^T \mathbf{x}^{(i)}\right)^2 \\ &= \arg\min_{\boldsymbol{\theta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2\end{aligned}$$

where $\mathbf{X} = \begin{pmatrix} 1 & x_1^{(1)} & \dots & x_p^{(1)} \\ 1 & x_1^{(2)} & \dots & x_p^{(2)} \\ \vdots & \vdots & & \vdots \\ 1 & x_1^{(n)} & \dots & x_p^{(n)} \end{pmatrix}$ is the $n \times (p+1)$-**design matrix**.

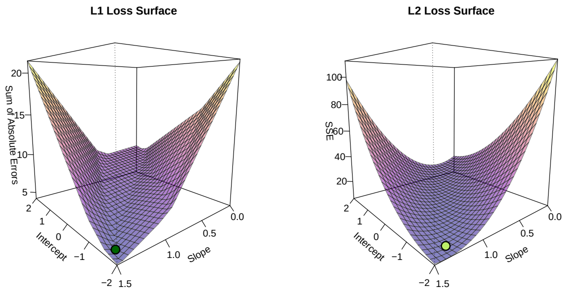
This yields the so-called normal equations for the LM:

$$\frac{\partial}{\partial \boldsymbol{\theta}} \mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) = 0 \quad \Longrightarrow \quad \hat{\boldsymbol{\theta}} = \left(\mathbf{X}^T \mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{y}$$

EXAMPLE: REGRESSION WITH L1 VS L2 LOSS

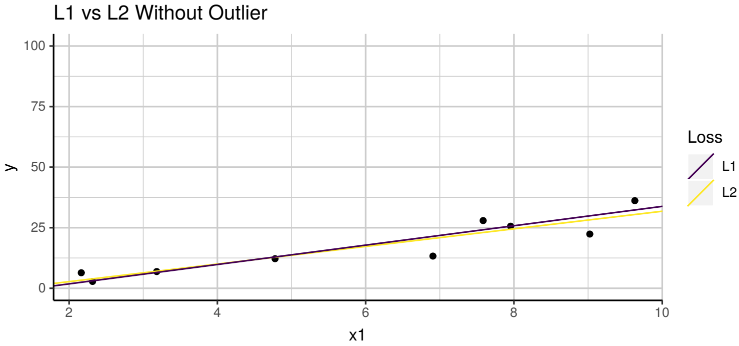
We could also minimize the L1 loss. This changes the risk and optimization steps:

$$\mathcal{R}_{\text{emp}}(\boldsymbol{\theta}) = \sum_{i=1}^{n} L\left(y^{(i)}, f\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)\right) = \sum_{i=1}^{n} \left|y^{(i)} - \boldsymbol{\theta}^T\mathbf{x}^{(i)}\right| \qquad \text{(Risk)}$$



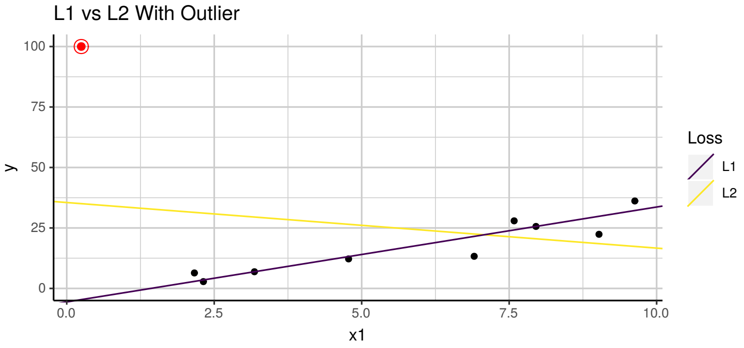
L1 loss is harder to optimize, but the model is less sensitive to outliers.

EXAMPLE: REGRESSION WITH L1 VS L2 LOSS



EXAMPLE: REGRESSION WITH L1 VS L2 LOSS

Adding an outlier (highlighted red) pulls the line fitted with L2 into the direction of the outlier:



LINEAR REGRESSION

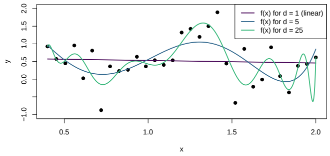
Hypothesis Space: Linear functions $\mathbf{x}^T\boldsymbol{\theta}$ of features $\in \mathcal{X}$.

Risk: Any regression loss function.

Optimization: Direct analytical solution for L2 loss, numerical optimization for L1 and others.

Einführung in das statistische Lernen

Polynomial Regression Models



Learning goals

- Understand how to add flexibility to the linear model by using polynomials
- Understand that this only affects the hypothesis space, not risk or optimization
- Understand that more flexibility is not equivalent to a better model

REGRESSION: POLYNOMIALS

We can make linear regression models much more flexible by using *polynomials* $\mathbf{x}_j^d$ – or any other *derived features* like $\sin(\mathbf{x}_j)$ or $(\mathbf{x}_j \cdot \mathbf{x}_k)$ – as additional features.

The optimization and risk of the learner remain the same.

Only the hypothesis space of the learner changes:
instead of linear functions

$$f\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right) = \theta_0 + \theta_1 \mathbf{x}_1^{(i)} + \theta_2 \mathbf{x}_2^{(i)} + \dots$$

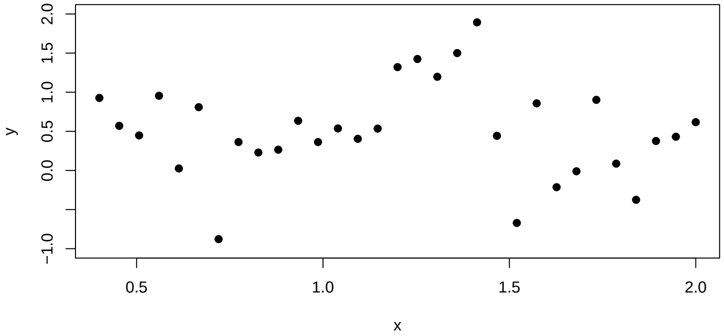
of only the original features,

it now includes linear functions of the derived features as well, e.g.

$$f\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right) = \theta_0 + \sum_{k=1}^{d} \theta_{1k} \left(\mathbf{x}_1^{(i)}\right)^k + \sum_{k=1}^{d} \theta_{2k} \left(\mathbf{x}_2^{(i)}\right)^k + \dots$$

REGRESSION: POLYNOMIALS

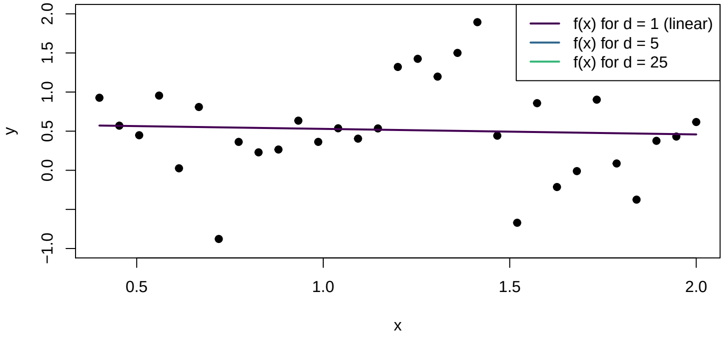
Polynomial regression example



REGRESSION: POLYNOMIALS

Polynomial regression example

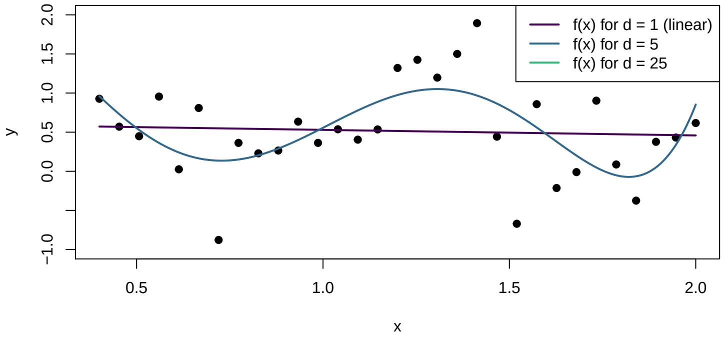
Models of different *complexity*, i.e., of different polynomial order d, are fitted to the data:



REGRESSION: POLYNOMIALS

Polynomial regression example

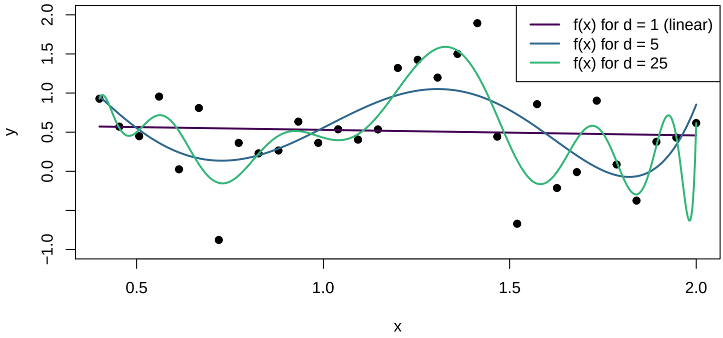
Models of different *complexity*, i.e., of different polynomial order *d*, are fitted to the data:



REGRESSION: POLYNOMIALS

Polynomial regression example

Models of different *complexity*, i.e., of different polynomial order d, are fitted to the data:



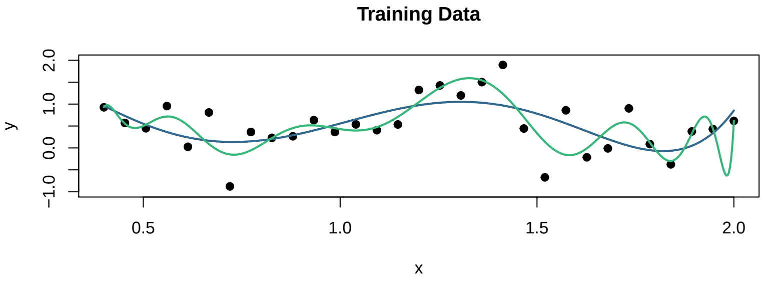
REGRESSION: POLYNOMIALS

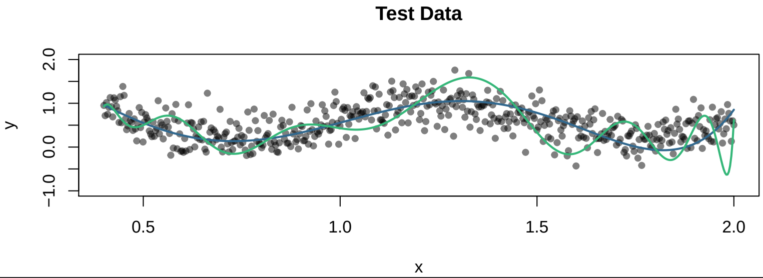
The higher d is, the more **capacity** the learner has to learn complicated functions of $\mathbf{x}$, but this also increases the danger of **overfitting**:

The model space $\mathcal{H}$ contains so many complex functions that we are able to find one that approximates the training data arbitrarily well.

However, predictions on new data are not as successful because our model has learnt spurious “wiggles” from the random noise in the training data (much, much more on this later).

REGRESSION: POLYNOMIALS


Training Data
y
x


Test Data
y
x