Life Sciences and
Facility Management

Institute of
Computational Life Sciences

Linear regression

Machine Learning

Norman Juchler

Example

- A basketball coach wants to analyze the effect of yoga sessions on the performance of their players.
- They thus collect data on the points scored during training (y) and number of yoga sessions attended (x), for each of their four players:

x	y
1	6
2	5
3	7
4	10

- To model the effect, they choose the following model:

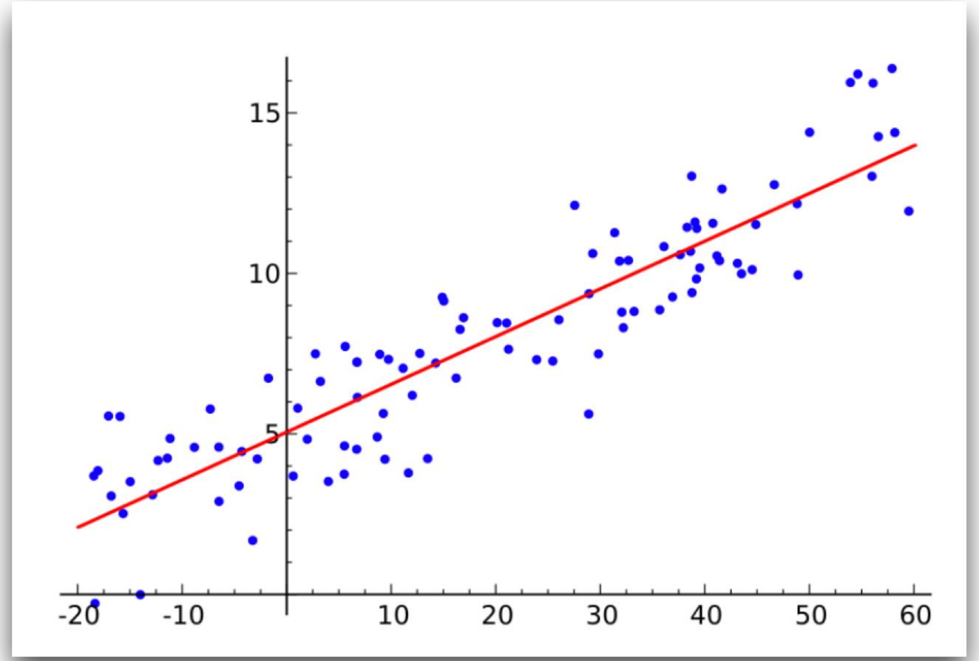
$$\textbf{points scored} = \beta_0 + \beta_1 \cdot \textbf{(yoga sessions)}$$

Linear model

- Assume we have a set of p features: $\boldsymbol{x}_i := (x_{i1}, x_{i2}, \ldots, x_{ip})$
- We want to use them to predict a target variable y
- The simple assumption we can make is (model ansatz):

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}$$

$$f_i(\boldsymbol{x}, \boldsymbol{\beta}) = \boldsymbol{\beta}\boldsymbol{x}$$



- Where the β_i are unknown parameters that we want to determine from the data

Ordinary least squares (OLS) regression

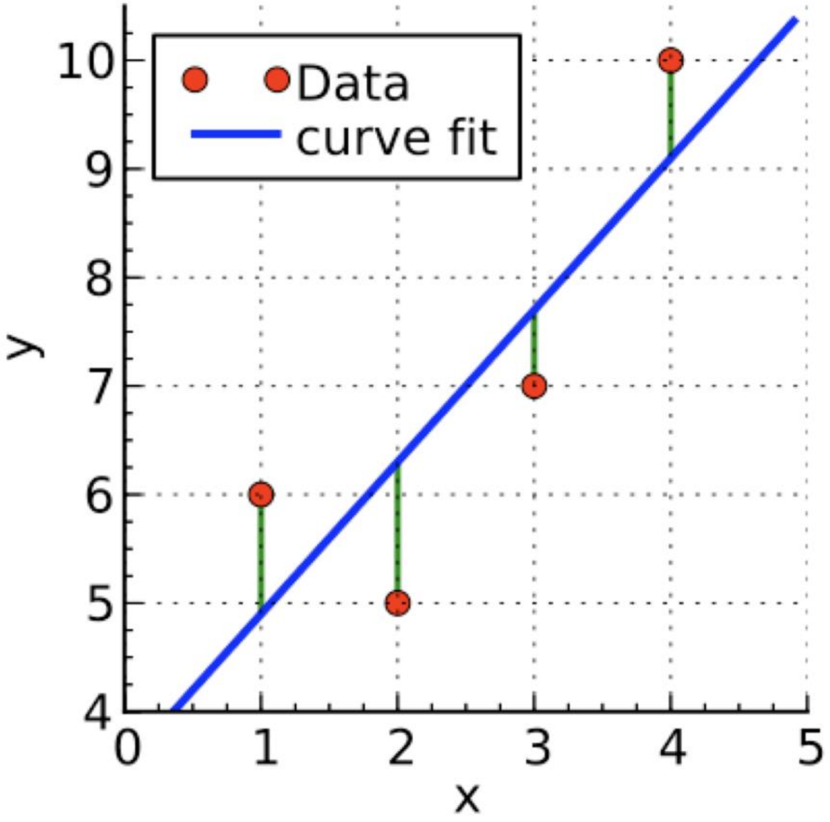
- Optimization objective for linear regression:

$$\boldsymbol{\beta}^* = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} (y_i - f(x_i, \boldsymbol{\beta}))^2 = \arg\min_{\boldsymbol{\beta}} \mathcal{L}(\boldsymbol{\beta})$$

- **In words**: Find parameters **β** such that the sum of squared residuals (green lines) is minimized.
- Using **calculus**, we can derive an analytical solution. For the above example (see also figure):

$$\begin{aligned}\mathcal{L}(\boldsymbol{\beta}) = S(\beta_0, \beta_1) &= r_1^2 + r_2^2 + r_3^2 + r_4^2 \\ &= [6 - (\beta_0 + 1\beta_1)]^2 + [5 - (\beta_0 + 2\beta_1)]^2 + \\ &\qquad [7 - (\beta_0 + 3\beta_1)]^2 + [10 - (\beta_0 + 4\beta_1)]^2 \\ &= 4\beta_0^2 + 30\beta_1^2 + 20\beta_0\beta_1 - 56\beta_0 - 154\beta_1 + 210\end{aligned}$$

$$\Rightarrow \quad \begin{cases} \frac{\partial \mathcal{L}}{\partial \beta_0} = 0 = 8\beta_0 + 20\beta_1 - 56 \\ \frac{\partial \mathcal{L}}{\partial \beta_1} = 0 = 20\beta_0 + 60\beta_1 - 154 \end{cases} \quad \Rightarrow \quad \begin{matrix} \beta_0 = 3.5 \\ \beta_1 = 1.4 \end{matrix}$$

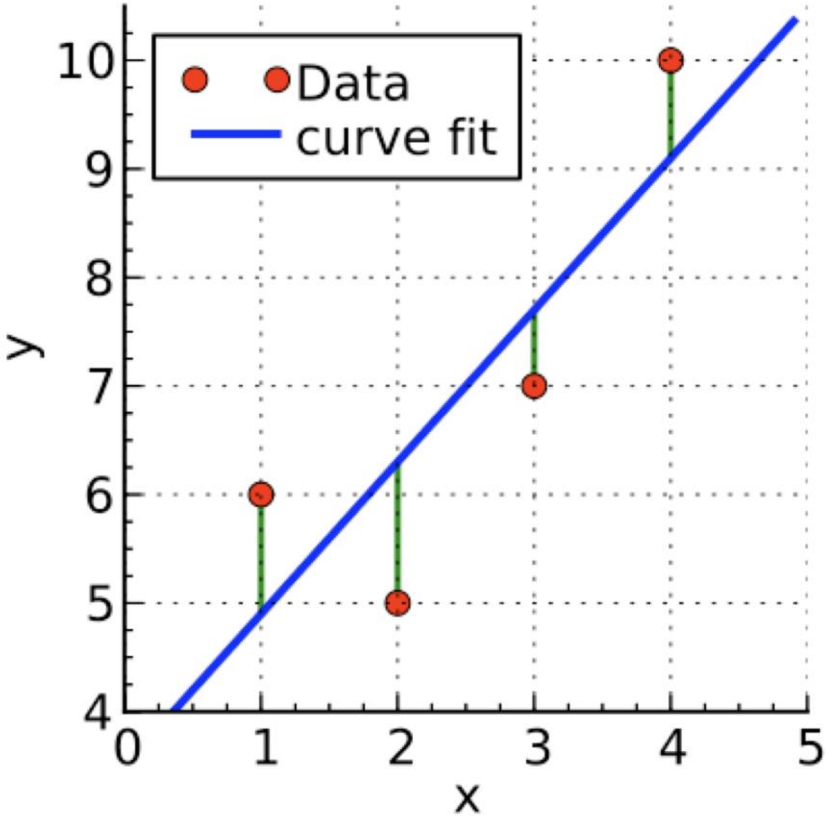


Ordinary least squares (OLS) regression

- Optimization objective for linear regression:

$$\boldsymbol{\beta}^* = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} (y_i - f(x_i, \boldsymbol{\beta}))^2 = \arg\min_{\boldsymbol{\beta}} \mathcal{L}(\boldsymbol{\beta})$$

- **In words**: Find parameters **β** such that the sum of squared residuals (green lines) is minimized.
- Using **calculus**, we can derive an analytical solution.
- For a general solution of the OLS regression problem, as well as proof, see here:
  - Definition: Simple linear regression
  - Proof: Ordinary least squares / optimal parameters β



Loss functions

- A loss function (a.k.a. cost function) in the context of machine learning usually measures how well the predicted values match the true target values.

$$\mathcal{L}(y - \hat{y})$$

- For regression, we used the residual sum of squares (RSS) as loss function.

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

- Sometimes, it is more meaningful to compute the **mean squared error** (MSE)

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \frac{1}{n} RSS$$

- (Note: both yield the same optimal solution!)

Model fitting ↔ training

- Goal of model fitting: Find parameter values $\boldsymbol{\beta}$ that minimize the loss function
- This requires solving an optimization problem.

- Possible approaches:
  - Try random parameter values and choose the best of these
  - Use calculus to derive exact solution
  - Use heuristic iterative method (Newton's method)
  - Gradient descent

Numerical methods for optimization: Newton's method

- There is no analytical solution for many (if not most) optimization problems
- We usually must rely on numerical/iterative methods to find (approximate) solutions

- **Newton's method** for finding the roots x^* (Nullstellen) of a function $f(x)$:
  - Start with an initial guess: x_0
  - Update the current guess x_n according to

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

  - Under certain conditions, the Newton method is guaranteed to converge to the true value of x^*.

- **Newton's method for optimization**:
  - Use Newton's method to solve $f'(x) = 0$

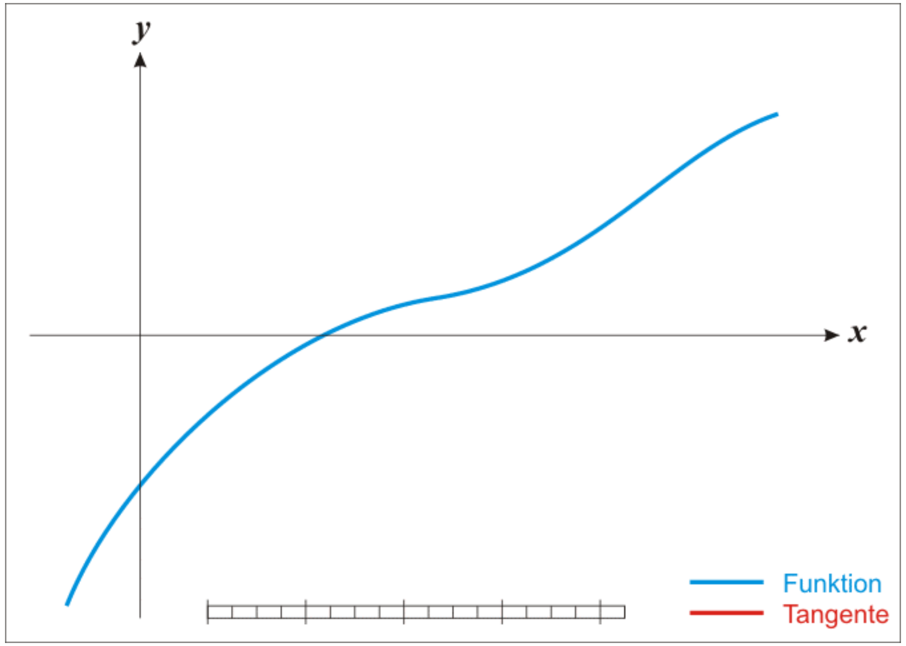


Image source: Link

Numerical methods for optimization: Gradient descent

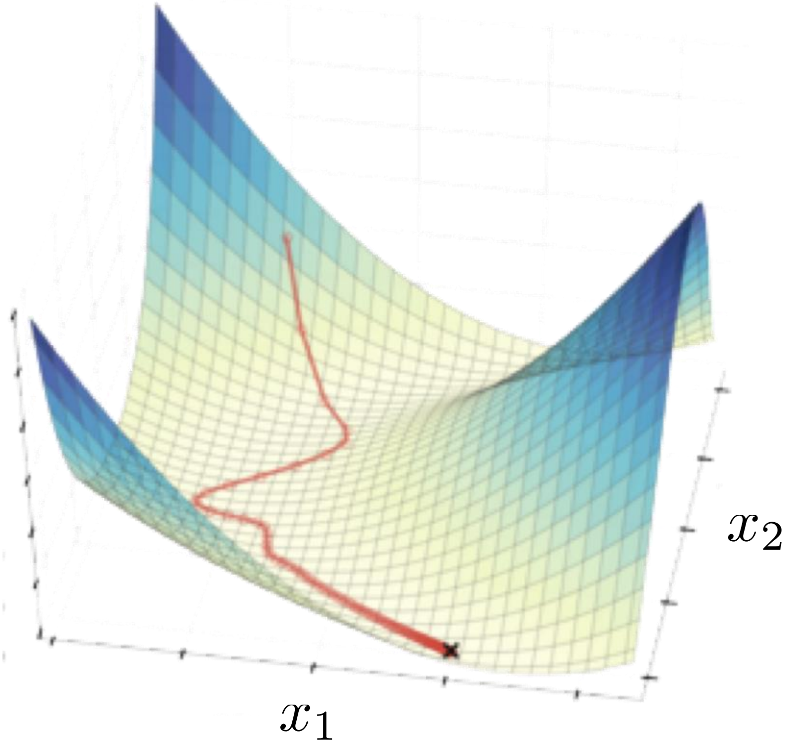
- Gradient descent is another iterative method to find solutions to $f'(\boldsymbol{x}) = 0$
- The method advances a small step in the direction of the gradient each time
- Gradient descent is applicable under more relaxed conditions

Gradient = direction of steepest assent

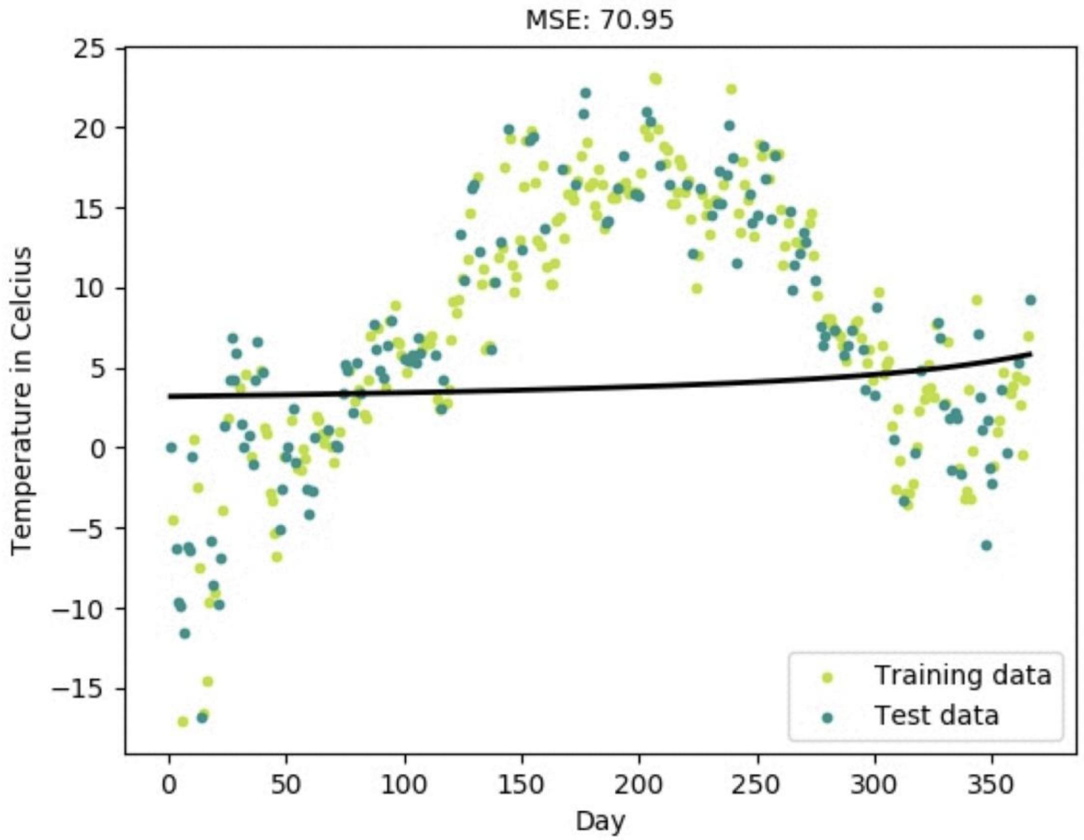
$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \gamma_n \nabla f(\boldsymbol{x}_n)$$

Learning rate (adjustable)

$$\mathcal{L}(\boldsymbol{x}) = f(\boldsymbol{x})$$



Example of a model fit



https://github.com/eriklindernoren/ML-From-Scratch

Multivariate regression

- Multivariate regression: Predict **multiple target variables** $y_1, \ldots, y_q$ simultaneously

- Example: The basketball coach extends the analysis to predict also the healthiness of the players:

  y_1 = points scored, y_2 = healthiness

- We then have two equations of the form (with j=1,2):

$$y_{ij} = \beta_{0j} + \beta_{1j} x_{i1} + \beta_{2j} x_{i2} + \ldots + \beta_{pj} x_{ip}$$

- ...where the same x features are used