

Zusammenfassung

*Maschinelles Lernen*

*WS 19/20*

November 26, 2019

# Grundlagen

## 1.1 Lineare Algebra

### 1.1.1 Skalarprodukt

- Vektoren  $x, y \in \mathbb{R}^n$ :  $x \circ y = \sum_{i=1}^n x_i \cdot y_i = x^T y$
- $\begin{bmatrix} 1 \\ 2 \end{bmatrix} \circ \begin{bmatrix} 3 \\ 4 \end{bmatrix} = 1 \cdot 3 + 2 \cdot 4 = 11$

### 1.1.2 Vektornorm

$f: \mathbb{R}^n \rightarrow \mathbb{R}$  mit

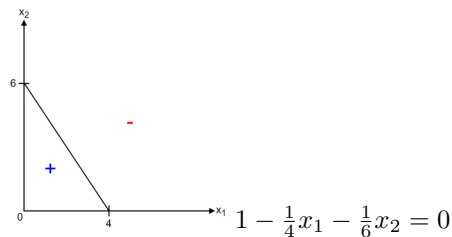
- $f(x) = 0 \Rightarrow x = 0$
  - $f(x + y) \leq f(x) + f(y)$  (Dreiecksungleichung)
  - $f(\alpha x) = |\alpha| f(x)$
- $L_1$ -Norm:  $\|x\|_1 = \sum_i |x_i|$
  - $L_2$ -Norm:  $\|x\|_2 = \sqrt{\sum_i x_i^2}$  (euklidische Norm)

### 1.1.3 Matrizen

- $m$  Zeilen und  $n$  Spalten  $A = \begin{bmatrix} A_{11} & \dots & A_{1n} \\ A_{m1} & \dots & A_{mn} \end{bmatrix}$ ,  $\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}^T = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}$
- $\begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \cdot \begin{bmatrix} g & h \\ i & j \\ k & l \end{bmatrix} = \begin{bmatrix} ag + bi + ck & ah + bj + cl \\ dg + ei + fk & dh + ej + fl \end{bmatrix}$ ,  $I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- $A^{-1}A = I$  (Matrizen mit linear abhängigen Zeilen oder Spalten (niedriger Rang) sind nicht invertierbar)

### 1.1.4 Hyperebene

- $x \in \mathbb{R}^d$  erfüllen Gleichung  $w_0 + w_1 x_1 + w_2 x_2 + \dots + w_d x_d = 0$  ( $w_0 + w^T x = 0$ )
- $d = 1$ : Skalar ( $w_0 + w_1 x_1$ ),  $d = 2$ : Gerade ( $w_0 + w_1 x_1 + w_2 x_2$ ),  $d = 3$ : Ebene
- Für einen Punkt  $x$  entscheidet das Vorzeichen  $\text{sgn}(w_0 + w^T x) \in \{-1, 0, 1\}$  auf welcher Seite der Hyperebene er liegt (bzw. ob er auf ihr liegt)



## 1.2 Analysis

### 1.2.1 Kettenregel

- Wenn  $z$  von  $y$  und  $y$  von  $x$  abhängt, dann gilt:  $\frac{dz}{dx} = \frac{dz}{dy} \frac{dy}{dx}$
- $f(x) = g(h(x)) = \frac{1}{2} \cdot (x_1 - x_2)^2 \rightarrow g(x) = \frac{1}{2}x^2$  und  $h(x) = x_1 - x_2$
- $\frac{df}{dx_2} = \frac{dg}{dh} \frac{dh}{dx_2} = h(x)(-1) = -(x_1 - x_2) = x_2 - x_1$

### 1.2.2 Partielle Ableitung

$$f(x) = 2x_1^3 - 5x_2^2 + 3, \quad \frac{df}{dx_1} = 6x_1^2, \quad \frac{df}{dx_2} = -10x_2$$

### 1.2.3 Gradient

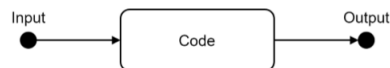
$$\nabla f = \begin{bmatrix} \frac{df}{dx_1} \\ \vdots \\ \frac{df}{dx_n} \end{bmatrix}, \quad f(x) = 2x_1^3 - 5x_2^2 + 3, \quad \nabla f = \begin{bmatrix} 6x_1^2 \\ -10x_2 \end{bmatrix}$$

## 1.3 Was ist maschinelles Lernen

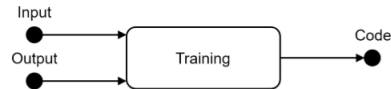
### 1.3.1 Paradigmenwechsel

Es ist schwierig, den entsprechenden Programmcode manuell zu schreiben, daher wird ein anderes Paradigma verwendet:

Traditionelle Programmierung:



Maschinelles Lernen:



Drei verschiedene Lernmethoden

- Überwachtes Lernen (*Supervised Learning*)
- Unüberwachtes Lernen (*Unsupervised Learning*)
- Bestärkendes Lernen (*Reinforcement Learning*)

## 1.4 Überwachtes Lernen

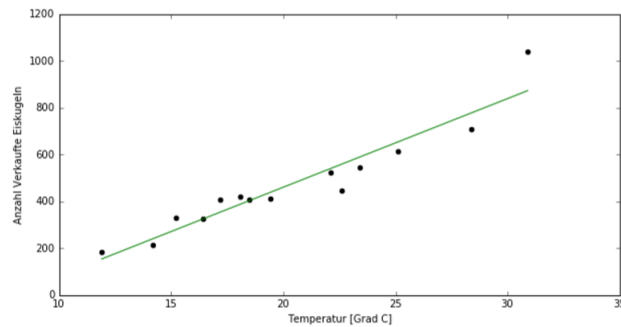
- Ziel: finden einer Funktion  $f : X \rightarrow Y$  wobei  $X$  auch *Features* / *Prädiktoren* und  $Y$  auch *Responses* genannt werden
- $X = \mathbb{R}^d$  ( $d$ -dimensionaler Vektorraum) mit  $d \in \mathbb{N}$
- Eine perfekte Abbildung ist nicht möglich, es treten *reduzierbare* Fehler (z.B. durch eine bessere Funktion  $f$ ) und *nicht reduzierbare* Fehler (z.B. Messfehler in Eingabedaten) auf
  - *Vorhersage*:  $y = f(x)$  optimieren wobei  $f$  auch *Blackbox* sein kann
  - *Inferenz*: Interpretierbarkeit von  $f$  steht im Vordergrund (Welche Prädiktoren sind für welche Response verantwortlich)
  - *Parametrische* Methoden: Annahme einer parametrisierten Struktur von  $f$  dessen Parameter mit Hilfe von Daten bestimmt werden
  - *Nicht-parametrische* Methoden: Keine Annahme einer Struktur von  $f$  sondern möglichst direkte Definition mit Hilfe von Daten
- Menge  $X$  und  $Y$  bekannt, genaue Abbildung  $f$  kann aber nur anhand von Beispielen  $D = \{(x^i, y^i) | x^i \in X, y^i \in Y, 1 \leq i \leq n\}$  (*Trainingsdatensatz* bzw. *gelabelte* Daten) erahnt werden

### 1.4.1 Beispiel Klassifikation

- Wenn  $Y$  diskrete Menge  $\{C_1, \dots, C_k\}$  für  $k \in \mathbb{N}$  dann handelt es sich um ein *Klassifikationsproblem*,  $C_1, \dots, C_k$  sind dann *Klassen* / *Kategorien*
- $|Y| = 2$  (*Binäre* Klassifikation) mit  $f : \mathbb{R} \rightarrow \{\text{angenehm, unangenehm}\}$  (Temperaturklassifikation)
- $|Y| = 5$  (*Mehrklassen*-Klassifikation) mit  $f : \mathbb{R} \rightarrow \{\text{frostig, kalt, angenehm, warm, heiß}\}$

### 1.4.2 Beispiel Regression

- Wenn  $Y$  kontinuierliche Menge, d.h.  $Y \subseteq \mathbb{R}$ , dann handelt es sich um ein *Regressionsproblem*
- Interesse an *quantitativen* Aussagen



- Ausgabemenge  $Y$  kann auch mehrdimensional sein (z.B.  $\{\text{gut, schlecht}\} \times \{\text{günstig, normal, teuer}\}$ )

## 1.5 Unüberwachtes Lernen

- Mehrwert erhalten ohne Zuhilfenahme von gelabelten Daten
- Man geht von Menge an Daten  $D = \{x^i | x^i \in X, 1 \leq i \leq n\}$  aus und versucht mehr über Beschaffenheit von  $X$  herauszufinden
- z.B. *Verteilung* von  $X$  bei Sprachmodellen, *Dimensionsreduktion* zur Verbesserung von überwachtem Lernverfahren

## 1.6 Datenvisualisierung

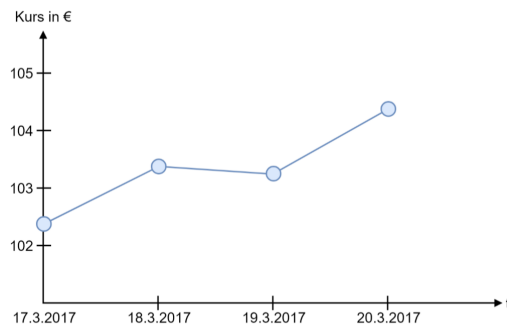


Abbildung 6: Beispiel eines Liniendiagramms.

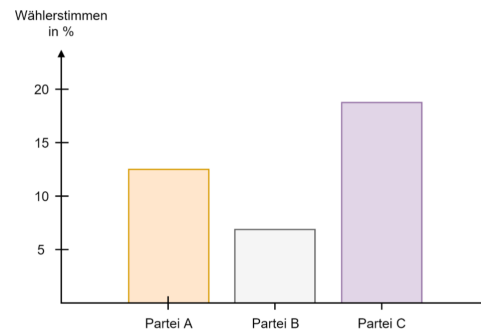


Abbildung 7: Beispiel eines Balkendiagramms.

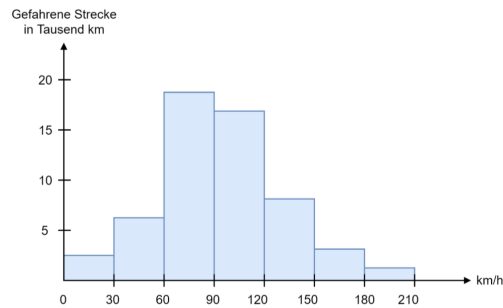


Abbildung 8: Beispiel eines Histogramms – eines speziellen Balkendiagramms.

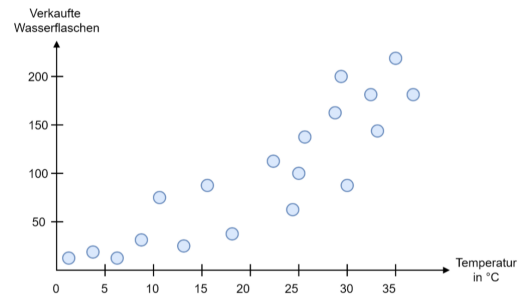


Abbildung 9: Beispiel eines Streudiagramms.

## 1.7 Datenvorverarbeitung

Bevor ein Modell erstellt und trainiert werden kann, müssen Daten durch

- *Auswahl*: Nur für den Anwendungsfall relevante Daten verwenden
- *Aufbereitung*
  - Dateiformat (Tabellen, BigData)
  - Bereinigung von unvollständigen oder ungültigen Daten
  - Repräsentative Auswahl bei langer Laufzeit / großem Speicheraufwand
- *Transformation*
  - Features in geeigneten Wertebereich bringen ( $[0, 1]$ )
  - Zerlegen in sinnvolle Features
  - Aggregation mehrerer Features