Einführung in Web- und Data-Science

Prof. Dr. Ralf Möller

Universität zu Lübeck

Institut für Informationssysteme

Tanya Braun (Übungen)

Teilnehmerkreis und Voraussetzungen

Studiengänge

- Bachelor **Informatik**
  - Pflicht in der kanonischen Vertiefung Web and Data Science

Voraussetzungen

- Keine

Organisatorisches: Übungen

- **Vorlesung**: Donnerstags, ab dem 18. Oktober 2018
- **Übungen**: Mittwochs 13-14 Uhr, ab 7.11.
  Anmeldung über Moodle nach dieser Veranstaltung
- **Übungsaufgaben** stehen jeweils nach der Vorlesung ca. ab 18 Uhr über Moodle bereit (erstes Übungsblatt erscheint am 25.10.2017)
- Aufgaben sollen in einer **2-er Gruppe** bearbeitet werden (also bitte Name(n) und Übungsgruppennummer vermerken)
- **Abgabe der Lösungen** erfolgt bis Montag 12 Uhr in der IFIS-Teeküche (jeweils in der zweiten Woche nach der Ausgabe)
- In den Übungen am Mittwoch wird der Übungszettel besprochen, dessen Lösungen bis zum jeweils vorigen Montag abgegeben werden, und auch Fragen zum jeweils neuen Übungszettel geklärt

Organisatorisches: Prüfung

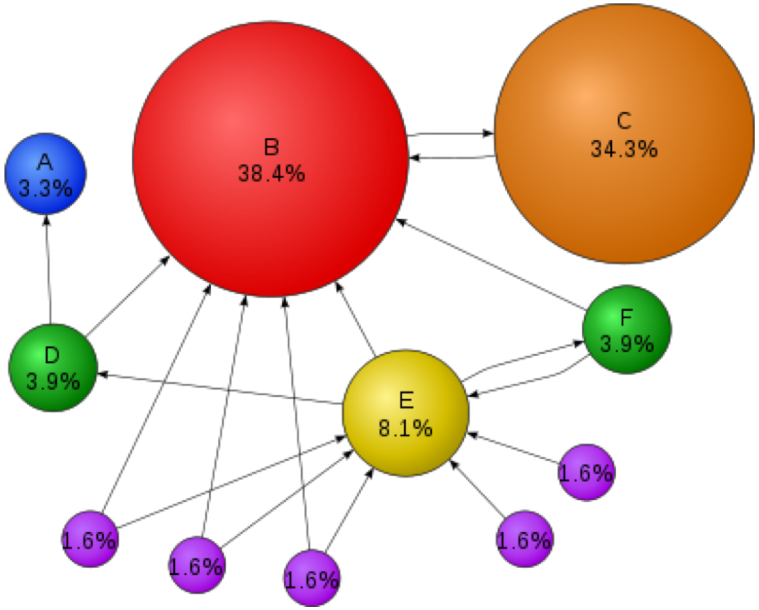
- Die **Eintragung in den Kurs** und in eine Übungsgruppe ist **Voraussetzung**, um an dem Modul Einführung in Web und Data Science teilnehmen zu können
- Am Ende des Semesters findet eine **Klausur** statt
  - Es geht nur um bestehen
- **Voraussetzung** zur Teilnahme an der Klausur sind mindestens **50% der gesamtmöglichen Punkte aller Übungszettel (240 Punkte insgesamt)**

Einführung in Web und Data Science

Begriffsbestimmung

Web und Data Science

- Web Science
  - Analyse von Strukturen im Web (Mensch und Computer)
  - Formalisierung durch große Graphstrukturen und entsprechende Entscheidungsprobleme über Graphen
  - Beispiel: Pagerank (Bewertung von Webseiten)



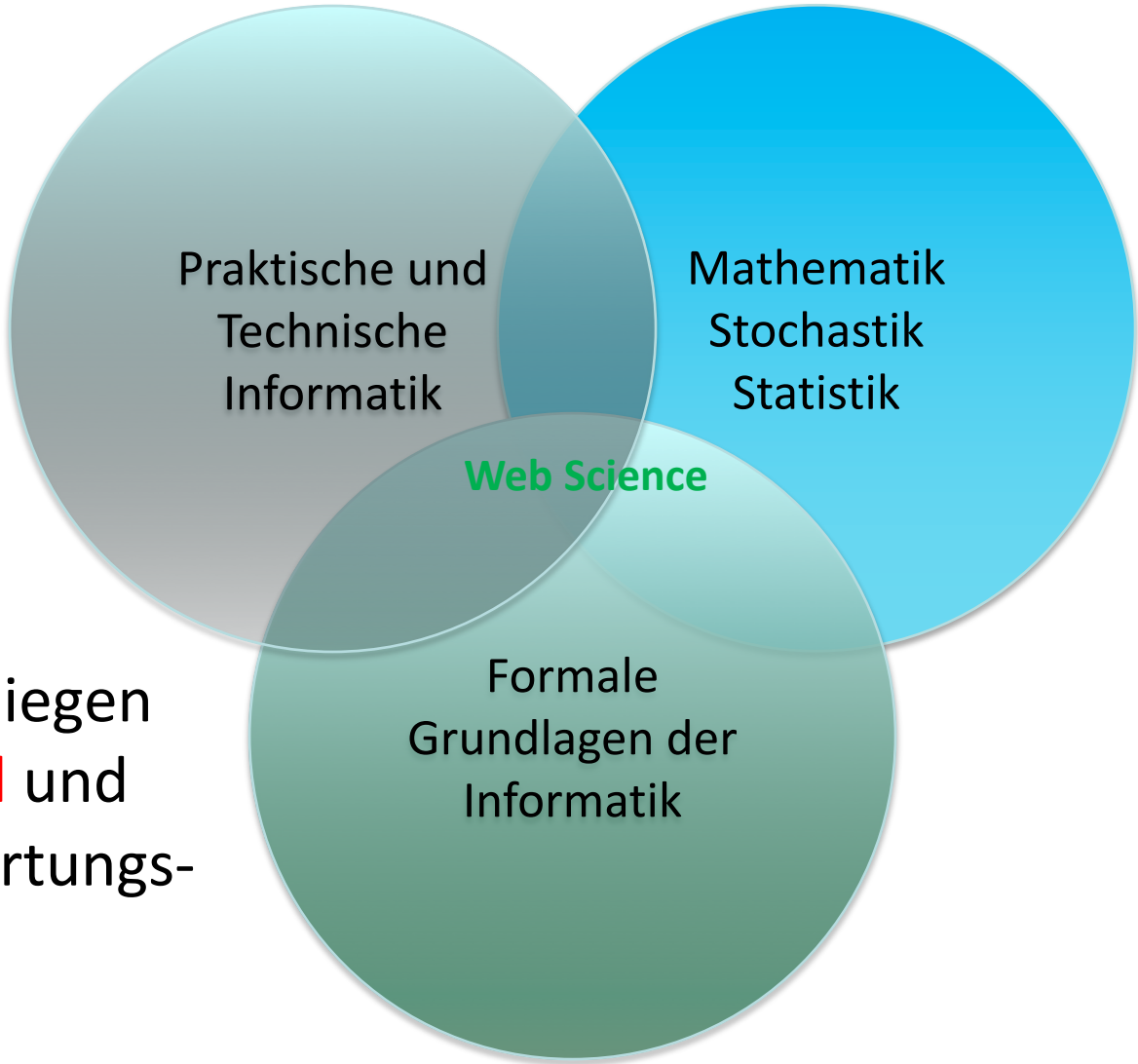
Zufallssurfer-Modell:

Größe der Kreise in etwa proportional der relativen Häufigkeit, mit der sich ein Surfer auf einer Seite befindet

[Wikipedia]

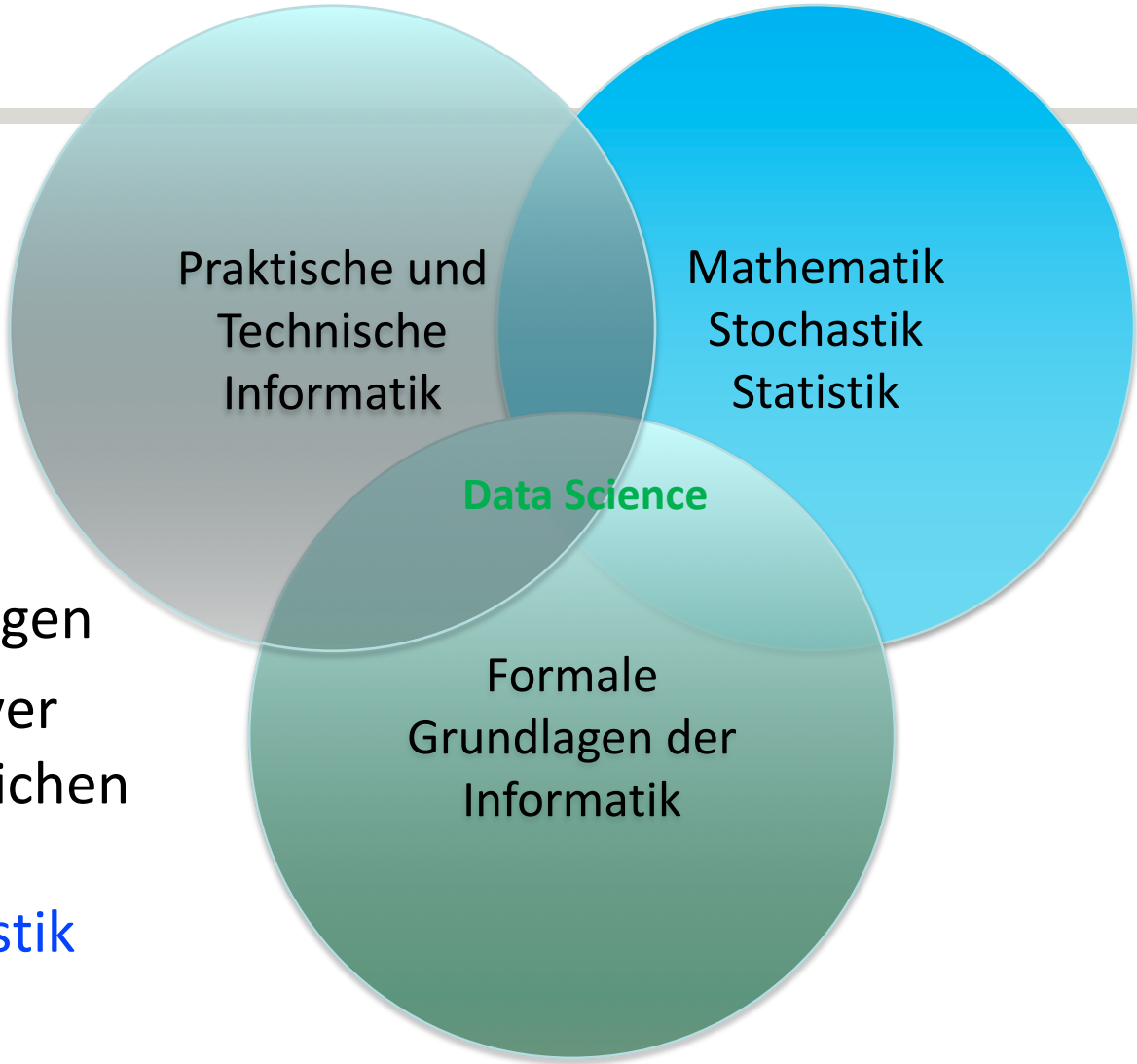
Herausforderungen für die Informatik

- Graphstrukturen extrem groß
- Verfahren zur Lösung von Entscheidungs-problemen extrem aufwendig
- Graphdaten unterliegen ständigem Wandel und so auch die Auswertungs-ergebnisse



Data Science

- Extraktion von Wissen aus Daten (u.a. Graphdaten)
- Begriff schon vor 50 Jahren für Informatik vorgeschlagen
- Entwicklung innovativer Konzepte in den Bereichen Logik, Datenbanken und Stochastik / Statistik (Datenanalyse und Wissensentdeckung)
- Verwendung von LADS und Analysis



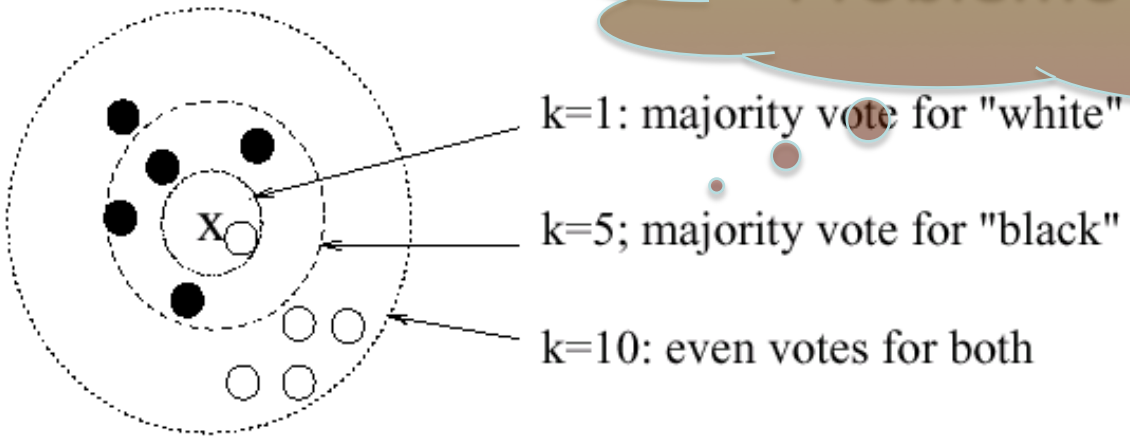
Herausforderungen für die Informatik

- Große Datenbestände
  - Speicher und Zugriffstechnologie
- Starker Zuwachs an Daten, hohe Dynamik
  - Hohe Datenraten und Echtzeitanforderungen
- Heterogene Datenbestände
  - Verteiltes Datenmanagement
  - Datenintegration
- Robuste Modelle der menschlichen Interpretationsvorgänge (Kognition) notwendig für Auswertung und Präsentation von Daten
  - Verarbeitung von Text-, Sprach- und Videodaten
- …

Speicheranforderungen für Anwendungen

- Annahme: Gegeben viele Datenpunkte
  - Beispielmerkmale: (x, y, Farbe), Farbe ∈ { white, black }
- Anfrage: Datenpunkt ohne Wert für bestimmtes Merkmal
  - Beispiel: Merkmal Farbe ohne Wert
- Anfragebeantwortung (Klassifikation des Anfragepunkts): Mehrheitsvotum der k-nächsten Nachbarn (kNN-Verfahren)

Probleme erkennbar?



Fix, E., Hodges, J.L. Discriminatory analysis, nonparametric discrimination: Consistency properties. Technical Report 4, USAF School of Aviation Medicine, Randolph Field, Texas, 1951

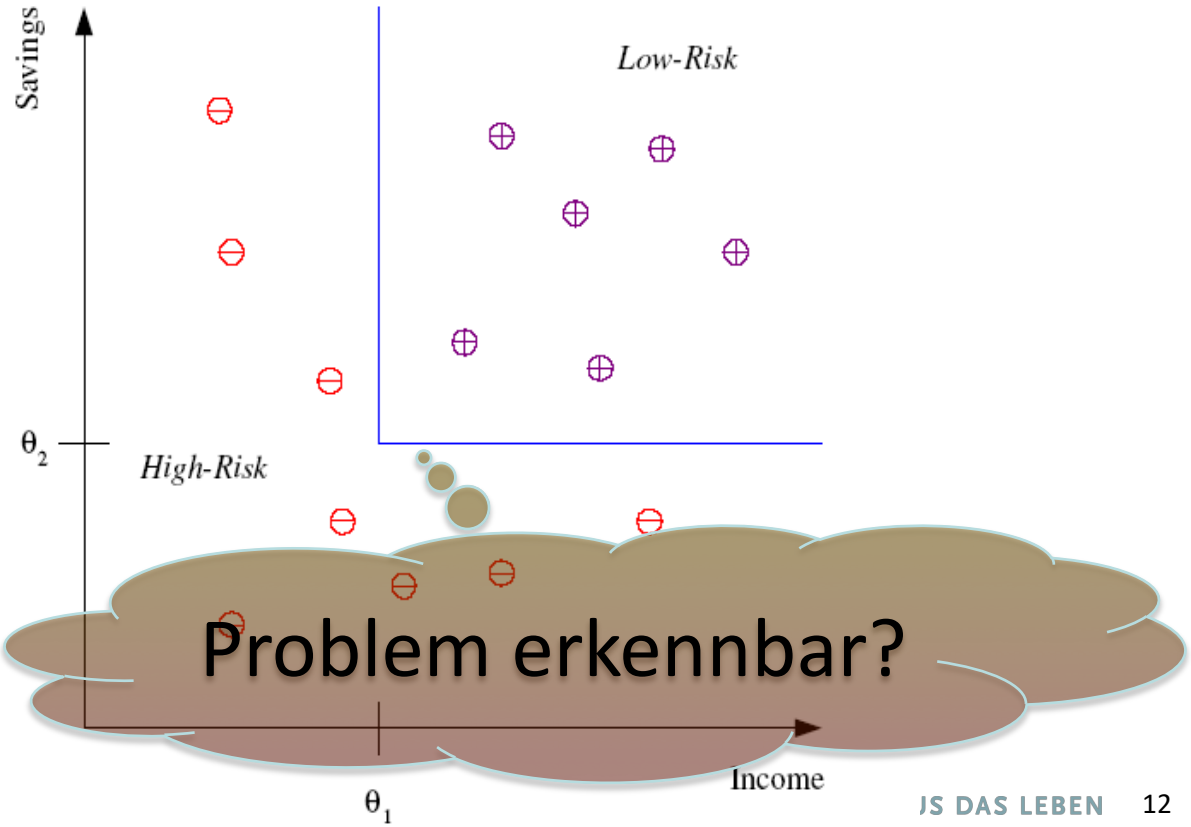
Probleme mit kNN

- Klassifikationsergebnis stark von k abhängig
- Hoher Speicherbedarf
- Effizienter Zugriff auf "Nachbarn" erfordert weitere Maßnahmen (noch mehr Speicherbedarf)

- Klassifikation basierend auf den Daten

Generalisierung von Daten möglich?

- Repräsentation der Daten durch Parameter
  - Wenn (Einkommen > θ_1 ∧Ersparnisse > θ_2), dann kreditwürdig (⊕), sonst nicht (⊖)
- Nur 2 Parameter nötig: (θ_1, θ_2)
- Modell fordert geringen Speicher



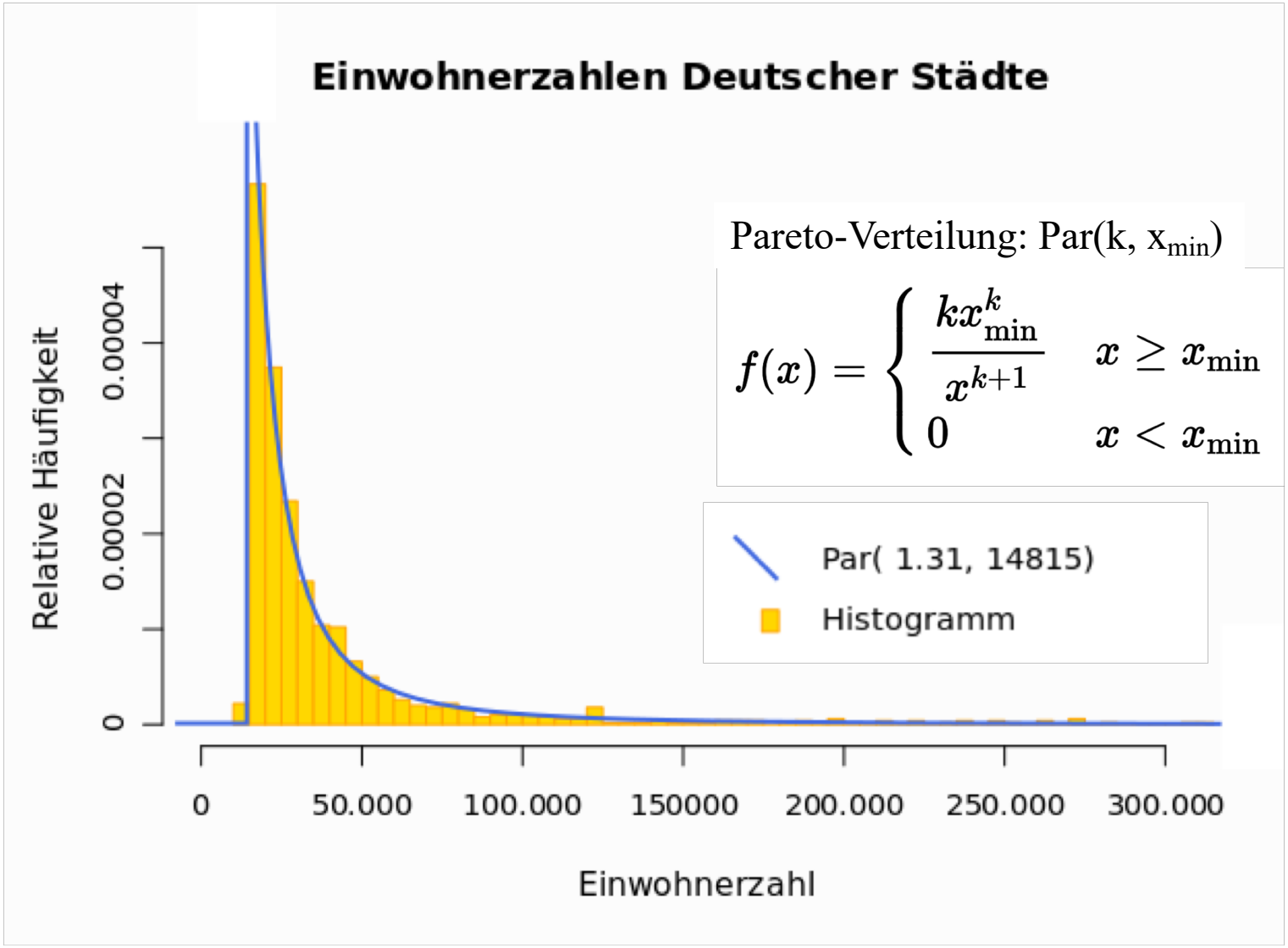
Beispiel

- Anzahl von Städten mit bestimmten Einwohnerzahlen schätzen

- Daten: Liste von Einwohnerzahlen (einige kommen mehrfach vor)

- Explizites Modell: Zählerfeld aufbauen
  - Zählerfeld ggf. sehr groß
- Implizites Modell: Potenzgesetz $y=ax^b$
  - a und b bestimmen
  - Aufwendiges Optimierungsproblem lösen

Begriff der "Verteilung"



Pareto-Verteilung: Par(k, x_{min})

$$f(x) = \begin{cases} \frac{k x_{\min}^k}{x^{k+1}} & x \geq x_{\min} \\ 0 & x < x_{\min} \end{cases}$$

[Wikipedia]

Über den Kurs

- Repräsentationssprachen
  - Modellierungsansatz
  - Entscheidungs- und Berechnungsprobleme
  - Verfahren zu deren Lösung
- Gewinnung von Modellen aus Daten als Optimierungsproblem (Algorithm. Datenanalyse, Data Mining, Maschinelles Lernen)
  - Überwachtes Lernen,
  - Unüberwachtes Lernen
    - Deep Learning, Transduktives Lernen, Reinforcement-Lernen
  - Transfer-Lernen
- Verwendung von Modellen in Anwendungen
  - Prädikation, Forensik, Verstehen der Realität (Science-Aspekt!)
  - Autonome Aktivitäten/ Agenten

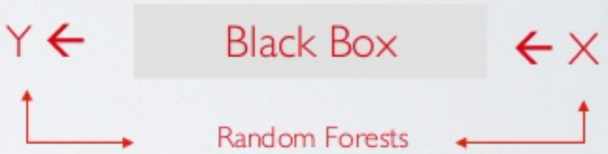
Data Models vs. Algorithmic Models

Data Modeling

vs.

Algorithmic Modeling

$Y \leftarrow F(X, \text{random noise, parameters})$



We understand the world

How well 'my data model' works
Statisticians, Data Analysts, Data Miners
Linear Regression
Logistic Regression
Known Distributions
Confidence Intervals
Predictor Variables & Goodness of Fit

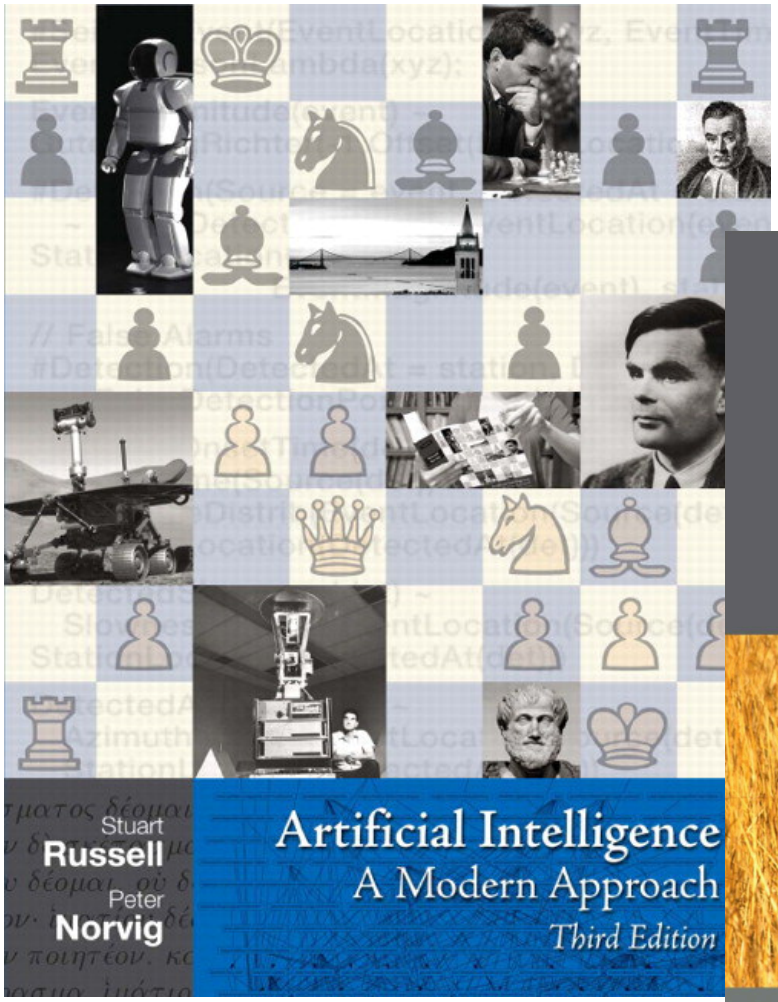
We don't understand the world

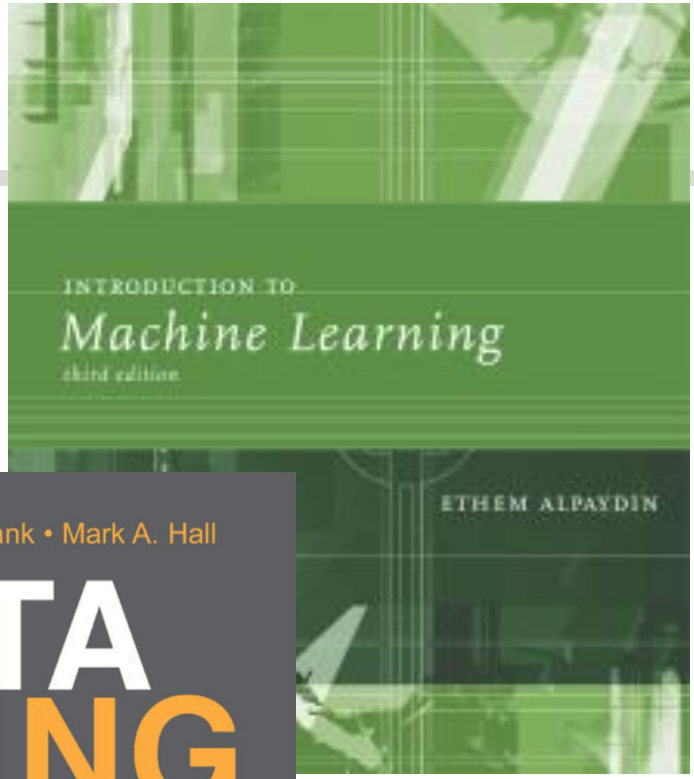
The world produces data in a black-box
Data Scientists
Machine Learning, AI & Neural Nets
Random Forests, SVM, GBT
Unknown Multivariate Distributions
Iterative
Predictive Accuracy

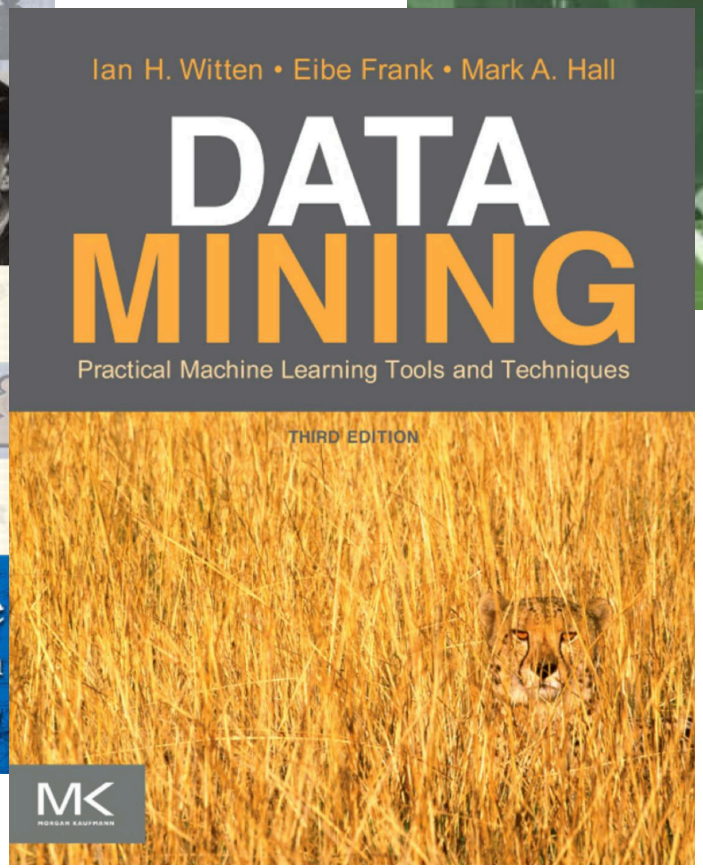
"Statistical Modeling: The Two Cultures" Leo Breiman, 2001

Data Science London

Literatur

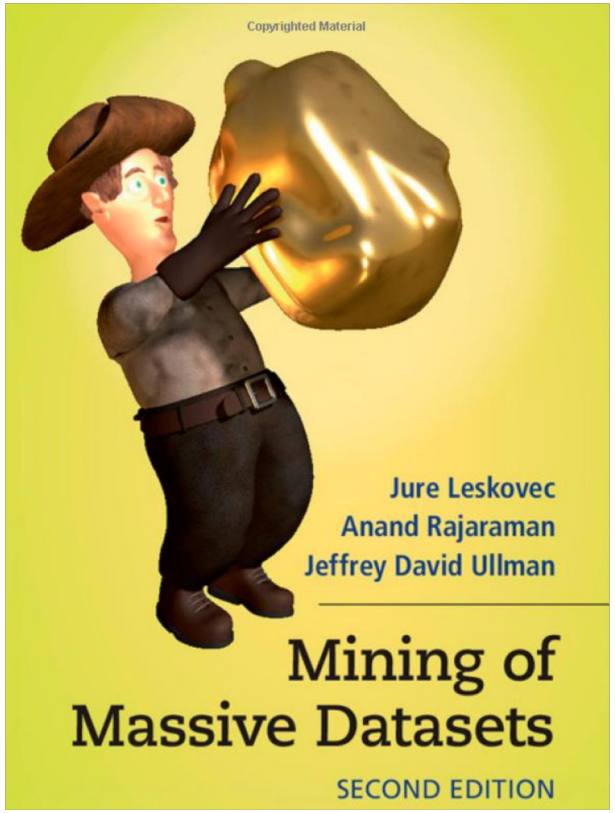

Stuart
Russell
Peter
Norvig
Artificial Intelligence
A Modern Approach
Third Edition


INTRODUCTION TO
Machine Learning
third edition
ETHEM ALPAYDIN


Ian H. Witten • Eibe Frank • Mark A. Hall
DATA
MINING
Practical Machine Learning Tools and Techniques
THIRD EDITION
MK

Literatur

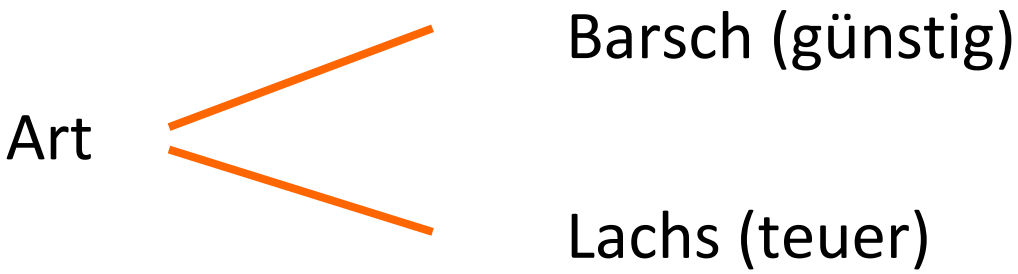
- Stuart Russell, Peter Norvig, **Artificial Intelligence – A Modern Approach**, Pearson, 2009 (oder 2003er Ed.)
- Ian H. Witten, Eibe Frank, Mark A. Hall, Data Mining: **Practical Machine Learning Tools and Techniques**, Morgan Kaufmann, 2011
- Ethem Alpaydin, **Introduction to Machine Learning**, 3rd Ed., MIT Press, 2014
- Jure Leskovec, Anand Rajaraman, Jeffrey D. Ullman, **Mining of Massive Datasets**, 2nd Ed., Cambridge University Press, 2014



- Viele zusätzliche Bücher, Präsentationen, und Videos im Web

Ein erweitertes Beispiel

“Sortierung von Fischen auf einem Förderband nach Arten durch Bildverarbeitung”



Problemanalyse

Verwende Kamera und nehme Bilder auf,
um Merkmale zu bestimmen:

- Länge
- Helligkeit
- Breite
- Anzahl und Form der Flossen
- Position des Mundes usw.

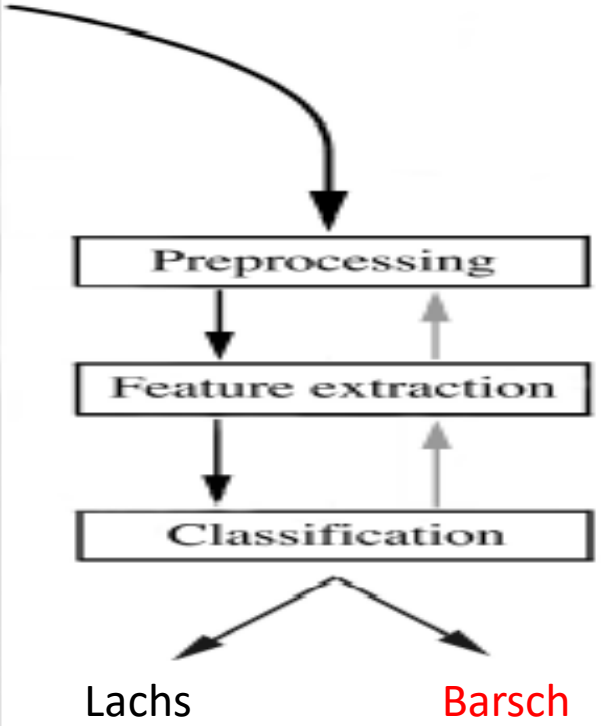
Menge aller möglichen Merkmale

Ziel: Wähle die relevanten aus

Vorverarbeitung

- Verwende Segmentierungsoperator, um Fisch und Hintergrund zu unterscheiden
- Fischregion wird verwendet, um Merkmalswerte zu extrahieren (geeignete Merkmale vorher festgelegt)
- Merkmalswerte weiterverarbeitet durch Klassifikator (Barsch oder Lachs)

Klassifikation



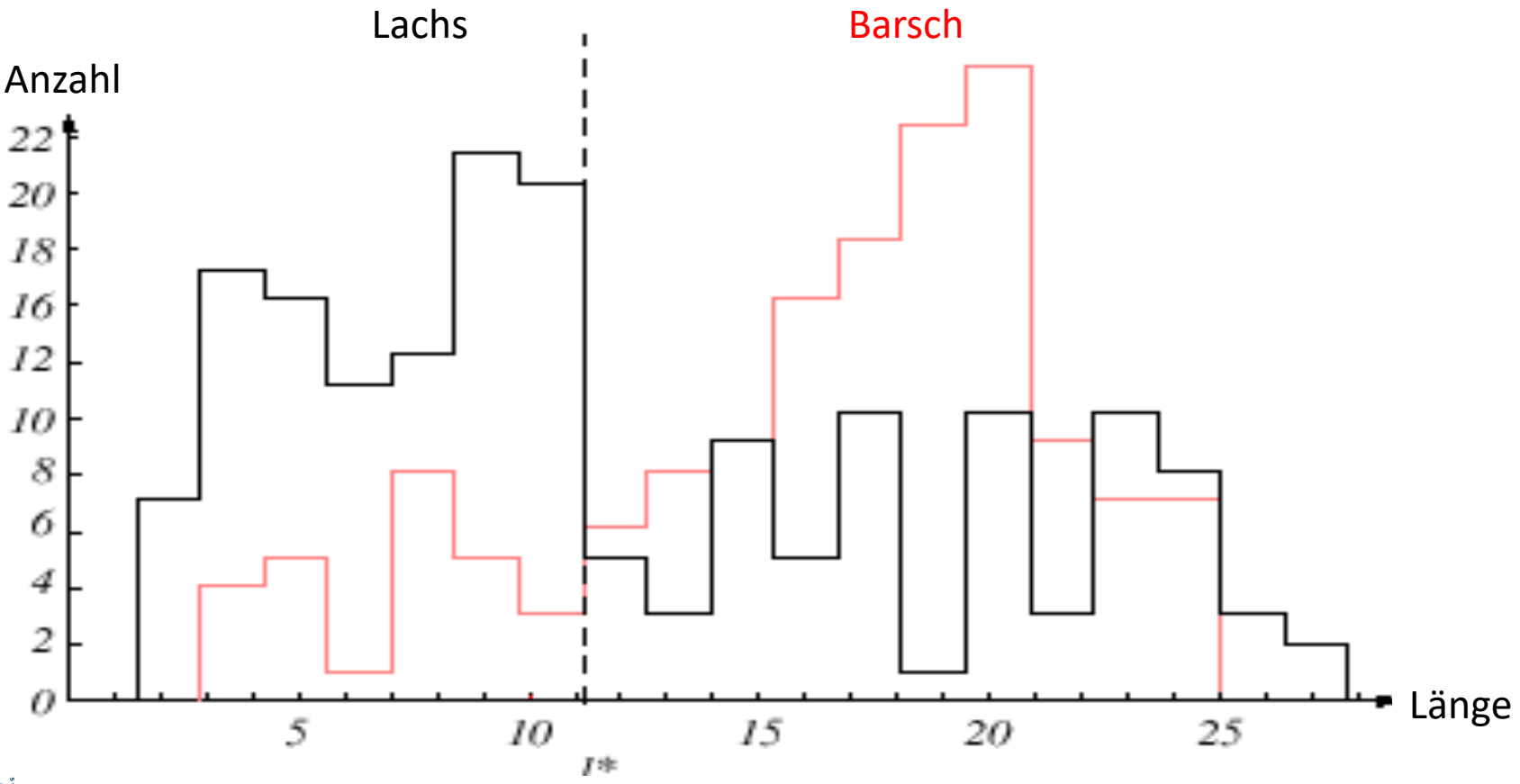
Bestimmung geeigneter Merkmale

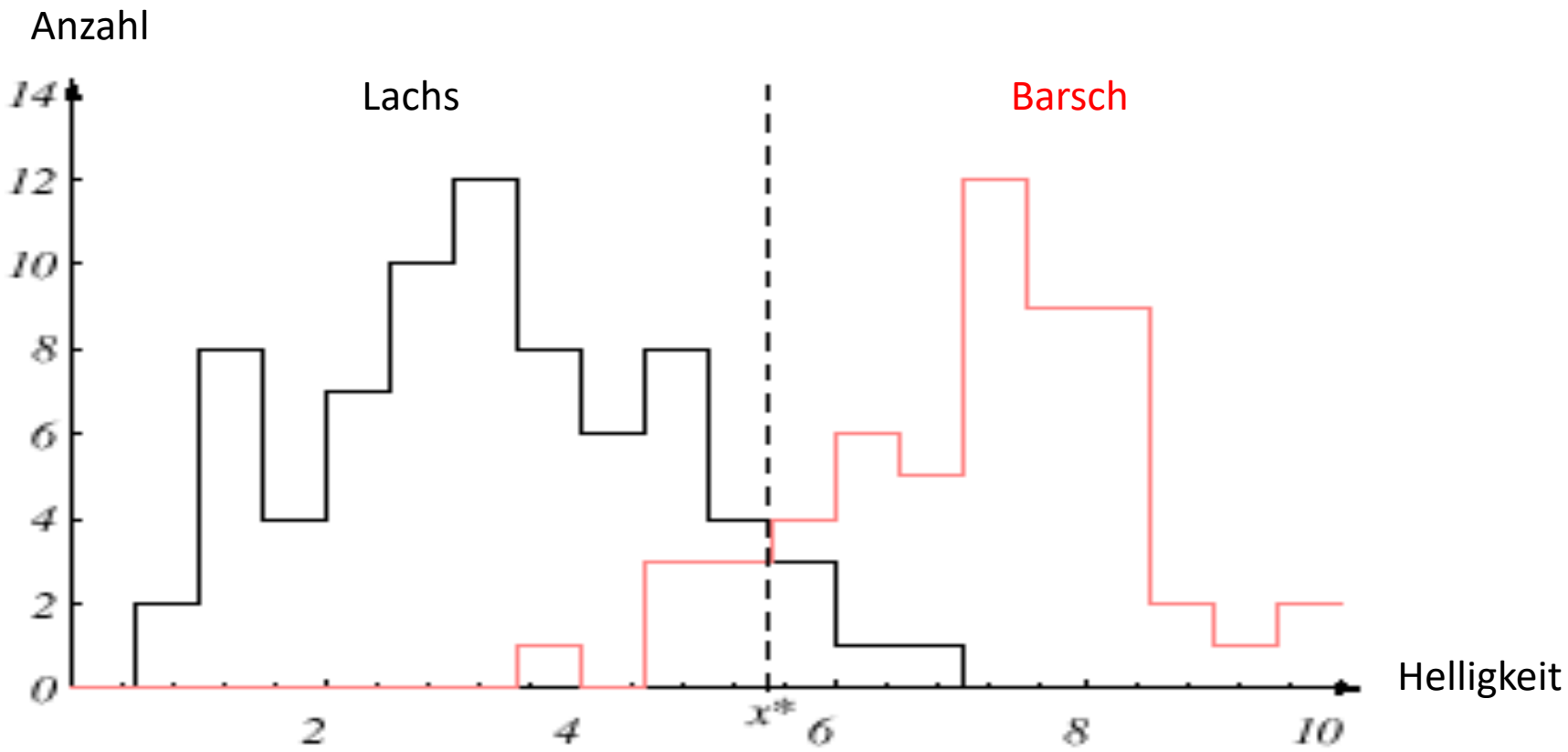
- Wir benötigen einen Experten, um die Merkmale festzulegen, mit denen man Barsche und Lachse richtig klassifizieren kann
- Wie wäre es mit Länge als Merkmal zur Unterscheidung?

Länge allein ist kein gutes Merkmal!

→ Hohe Kosten bei Fehlentscheidung

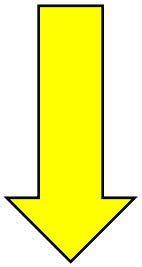
Wie wäre es mit Helligkeit?




Anzahl
Lachs
Barsch
14
12
10
8
6
4
2
0
2
4
x*
6
8
10
Helligkeit

Schwellwert-Entscheidungsgrenze und induzierte Kosten

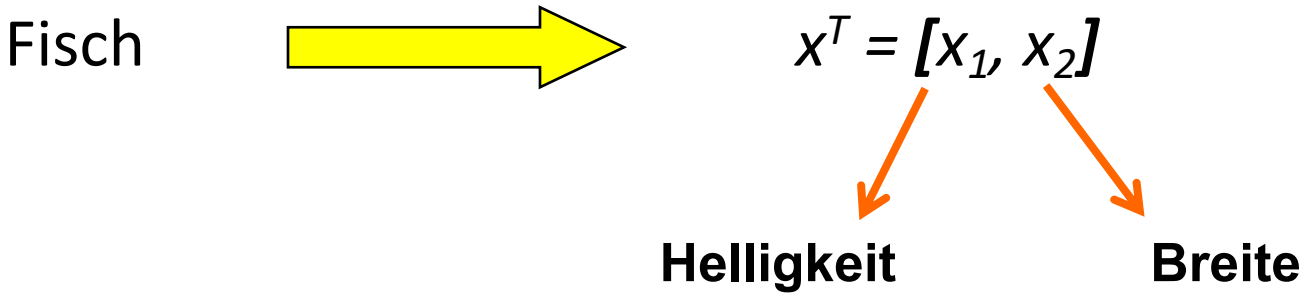
– Schwellwert-Entscheidungsgrenze in Richtung mittlerer Helligkeitswerte minimiert die Kosten (der Fehlklassifikation)

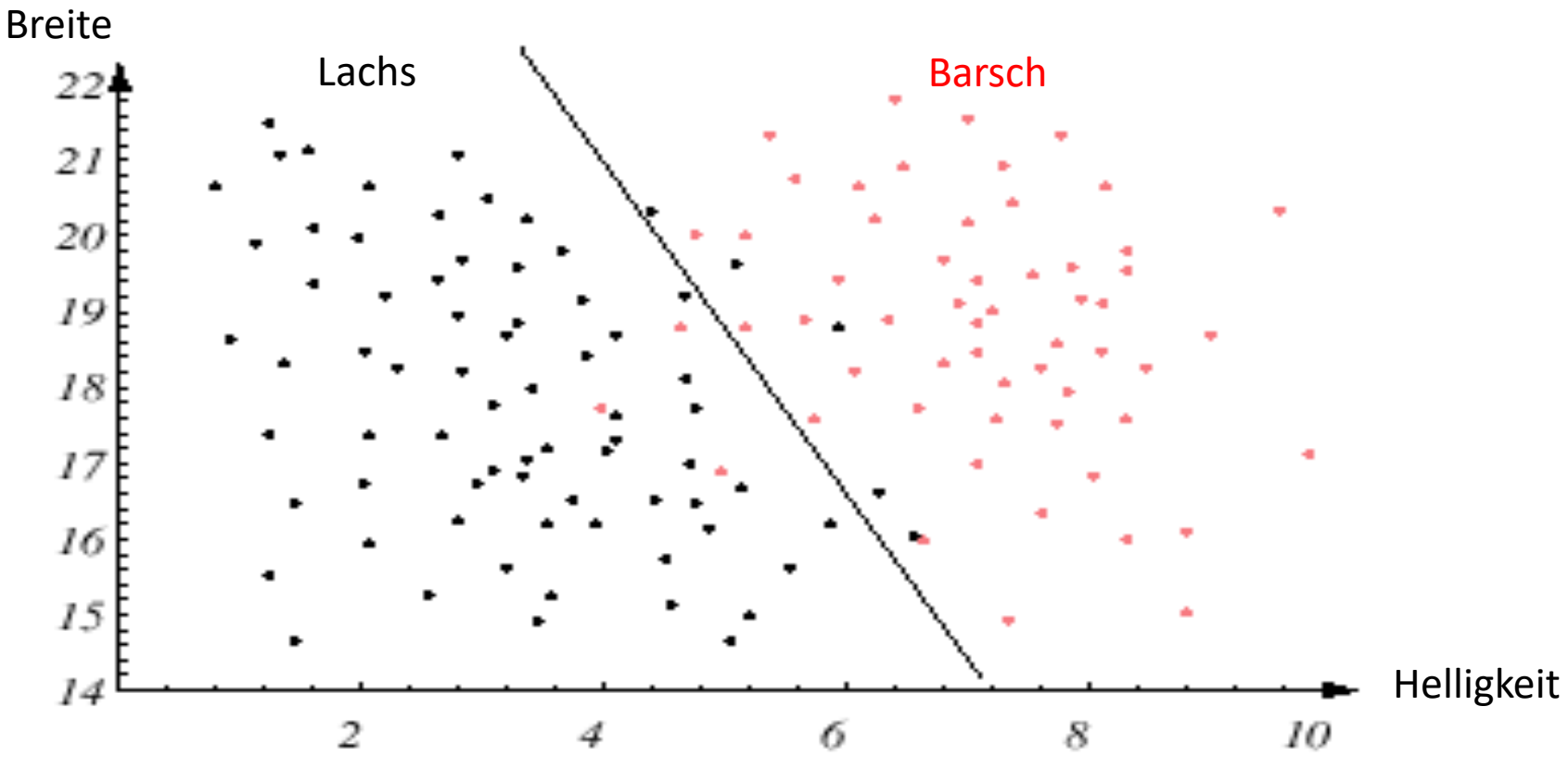


Untersuchung in der sog. Entscheidungstheorie

Ziel ist die automatische Bestimmung von Berechnungsfunktionen für geeignete Merkmale

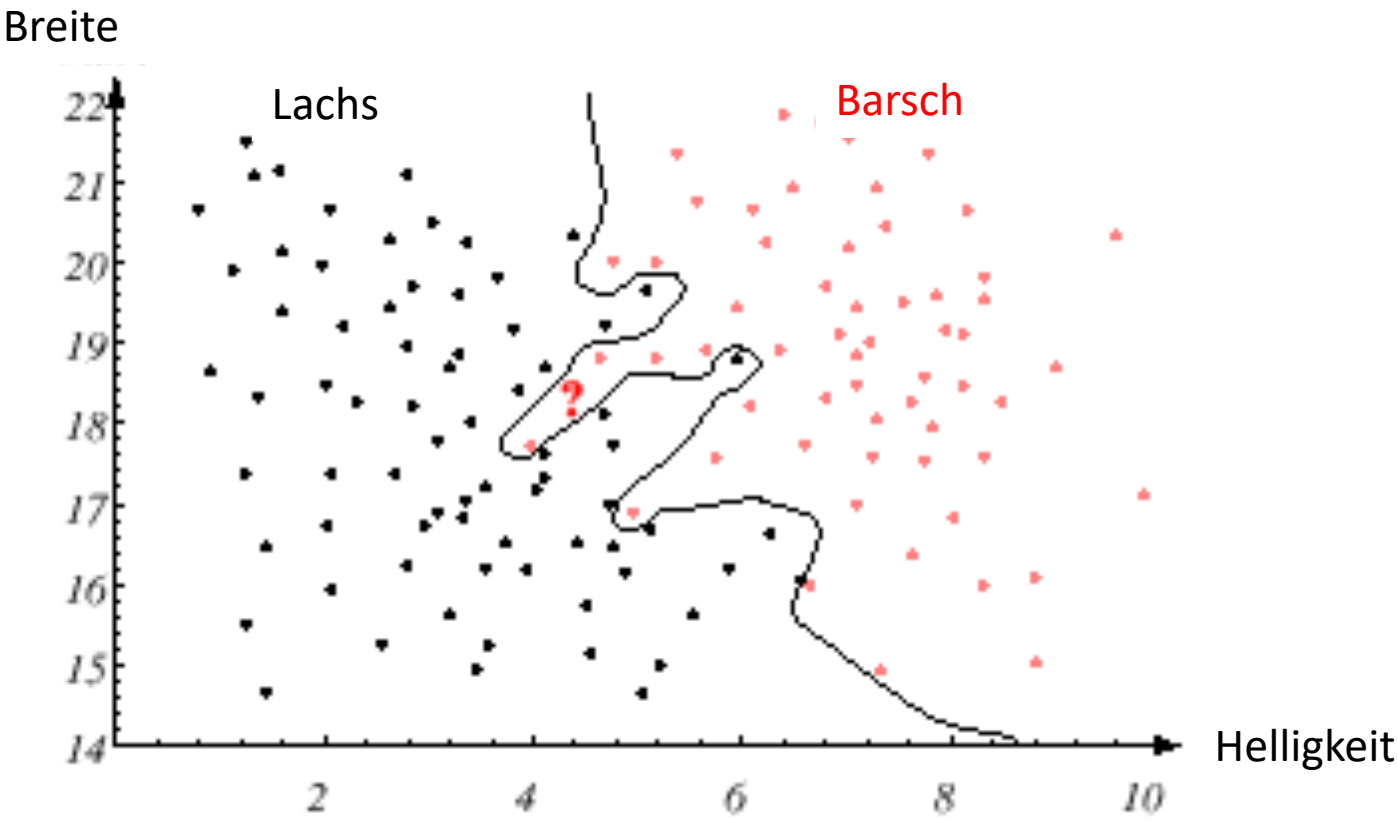
Passe Helligkeit an
und verwende zusätzlich die Breite des Fisches



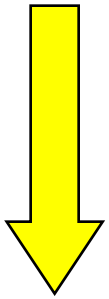

Breite
Lachs
Barsch
22
21
20
19
18
17
16
15
14
2
4
6
8
10
Helligkeit

- Weitere Merkmale, die nicht direkt zu Helligkeit und Breite in Beziehung stehen, könnten hinzukommen
  - Vorsicht aber vor Reduktion durch "verrauschte Merkmale"

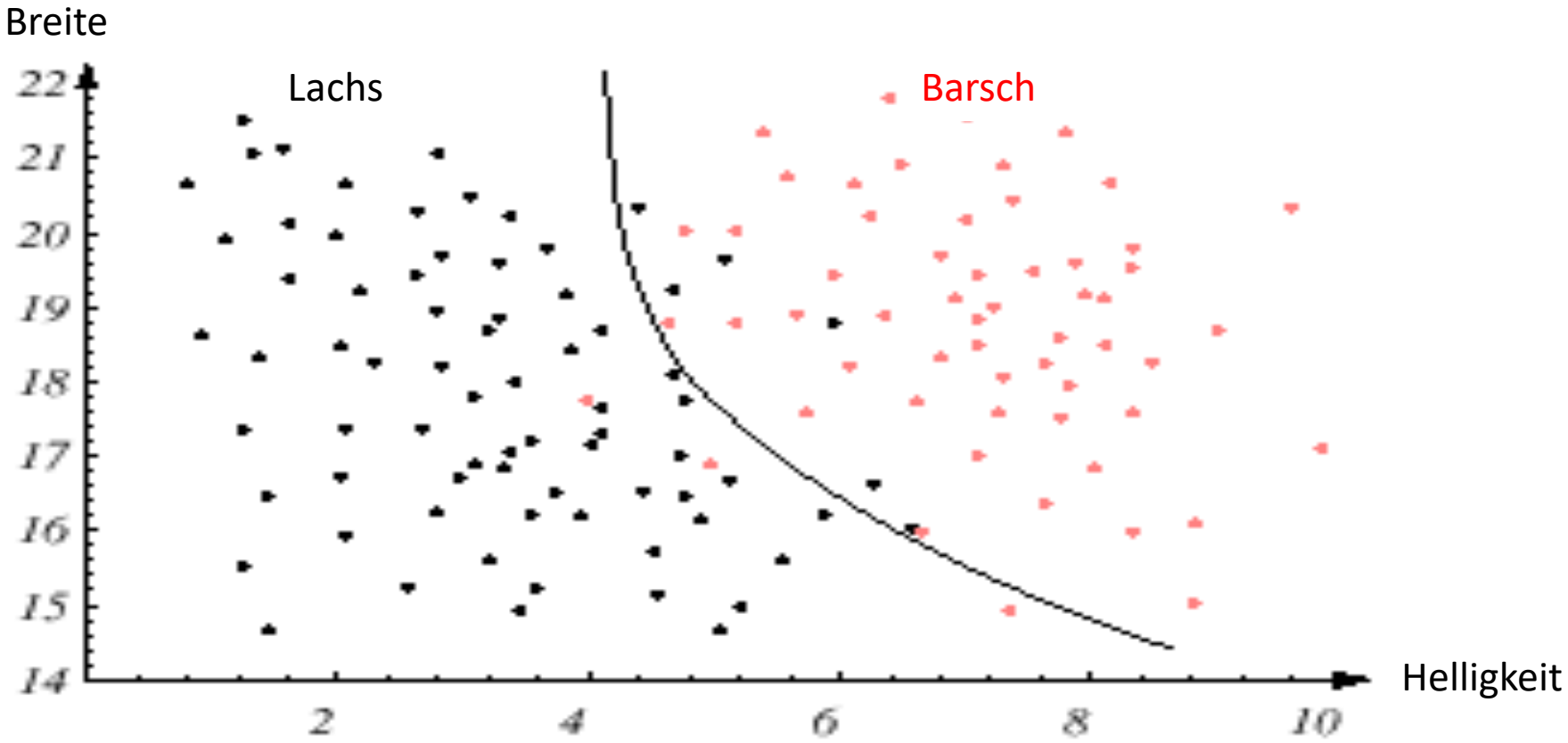
- Wünschenswerterweise ergibt
  die beste Entscheidungsgrenze eine optimale Performanz
  (im Sinne einer Verlustminimierung durch Falschklassifikation)


Breite
Lachs
Barsch
?
22
21
20
19
18
17
16
15
14
2
4
6
8
10
Helligkeit

Vorfreude über Klassifikationsleistung auf Testdaten kann verfrüht sein. Wichtig ist die Leistung auf neuen Daten!



Generalisierungsfähigkeit zählt!


Breite
Lachs
Barsch
22
21
20
19
18
17
16
15
14
2
4
6
8
10
Helligkeit

Klassifikatorentwicklung

- Relevante Merkmale automatisch bestimmbar?
  - Deep Learning, Ensembles
- Dynamische Angepassung des Klassifikators möglich?
  - Ohne spezielle Groundtruth Daten: Transduktives Lernen
  - Mit Belohnungssystem: Reinforcement-Lernen
- Übertragung eines Klassifikators auf neue Anwendung?
  - Transfer-Lernen

Einführung in Web- und Data-Science

Prof. Dr. Ralf Möller

Universität zu Lübeck

Institut für Informationssysteme

Tanya Braun (Übungen)

Bewertung eines Klassifikators

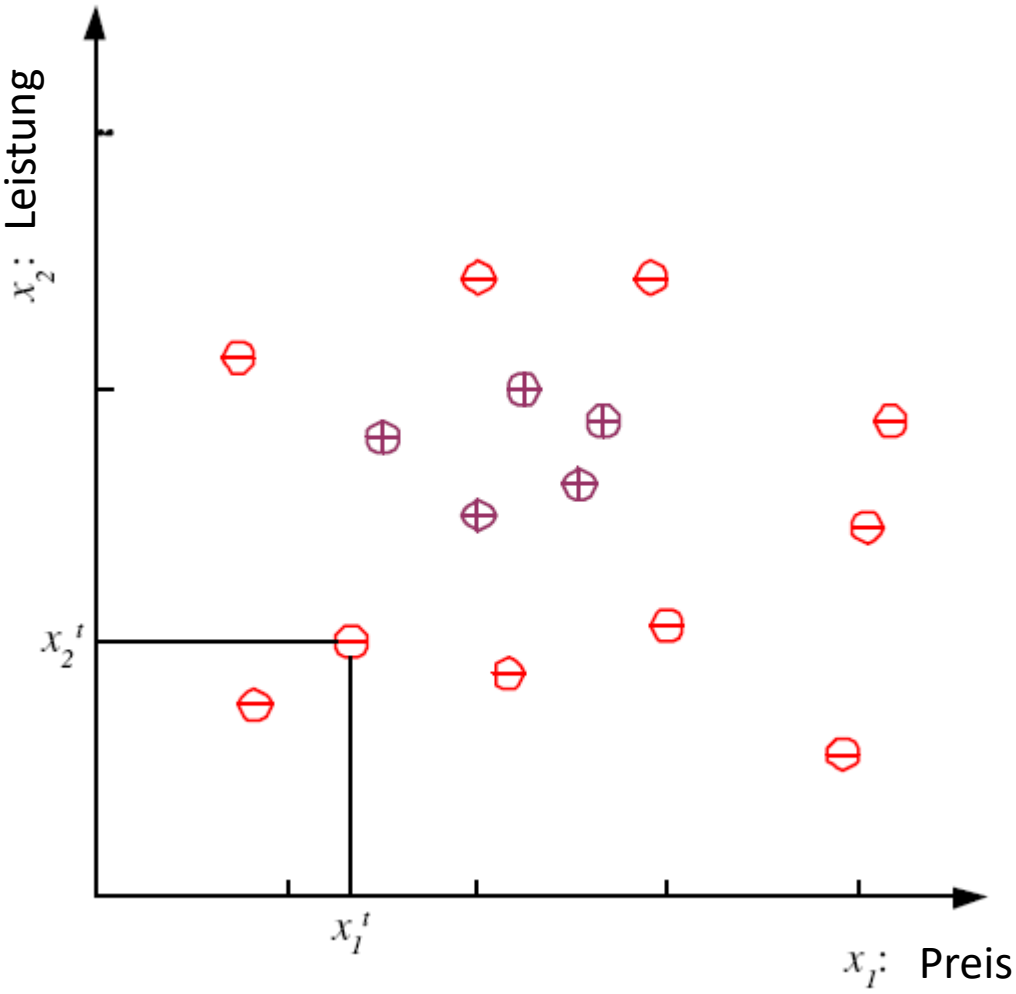
- Klasse C eines “Familienautos”
  - Vorhersage: Ist Auto x ein Familienauto?
  - Wissensextraktion:
    Was erwarten Menschen von einem Familienauto?
- Ausgabe:

  Positive (+) und negative (–) Beispiele
- Repräsentation der Eingabe:

  x_1: Preis, x_2 : Leistung

Frage: Wie gut funktioniert ein bestimmter Klassifikator?

Trainingsmenge $\mathcal{X}$



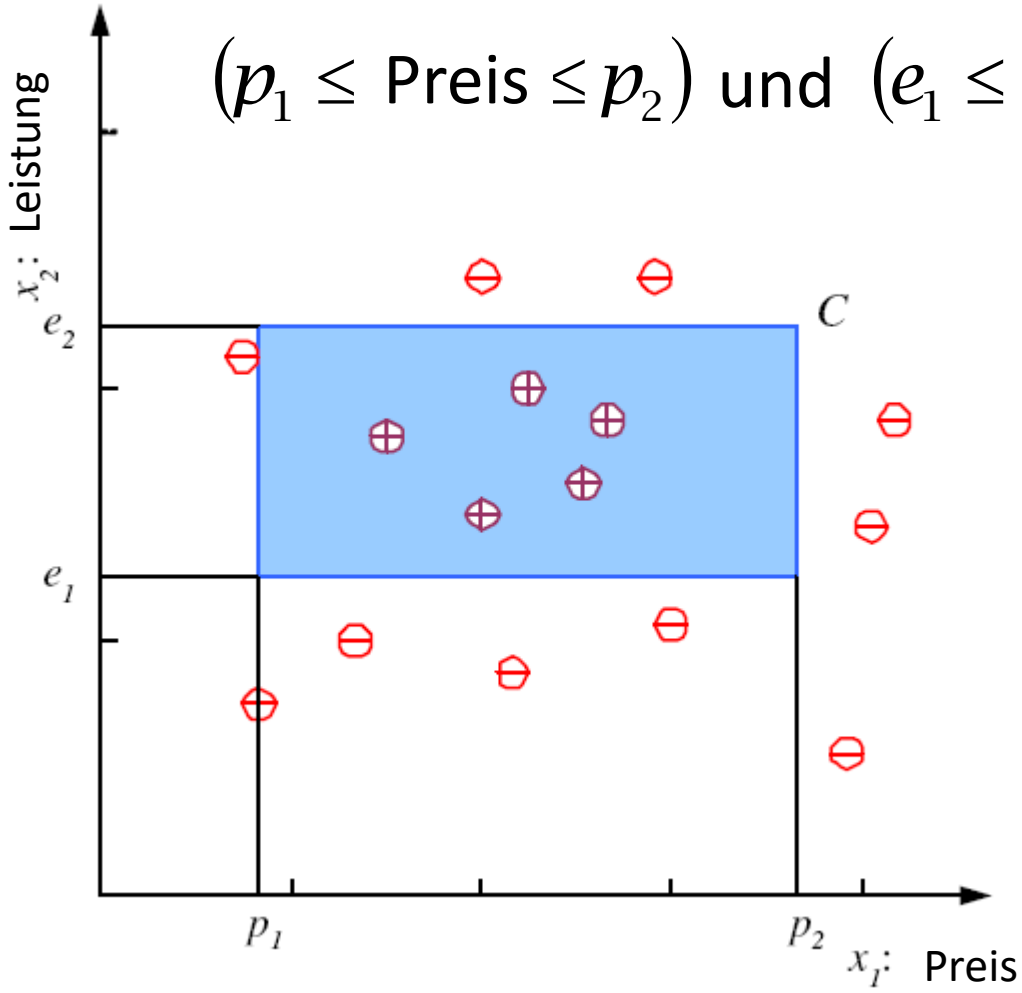
$$\mathcal{X} = \{\boldsymbol{x}^t, r^t\}_{t=1}^N$$

$$r = \begin{cases} 1 \text{ if } \boldsymbol{x} & \text{ist positiv} \\ 0 \text{ if } \boldsymbol{x} & \text{ist negativ} \end{cases}$$

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

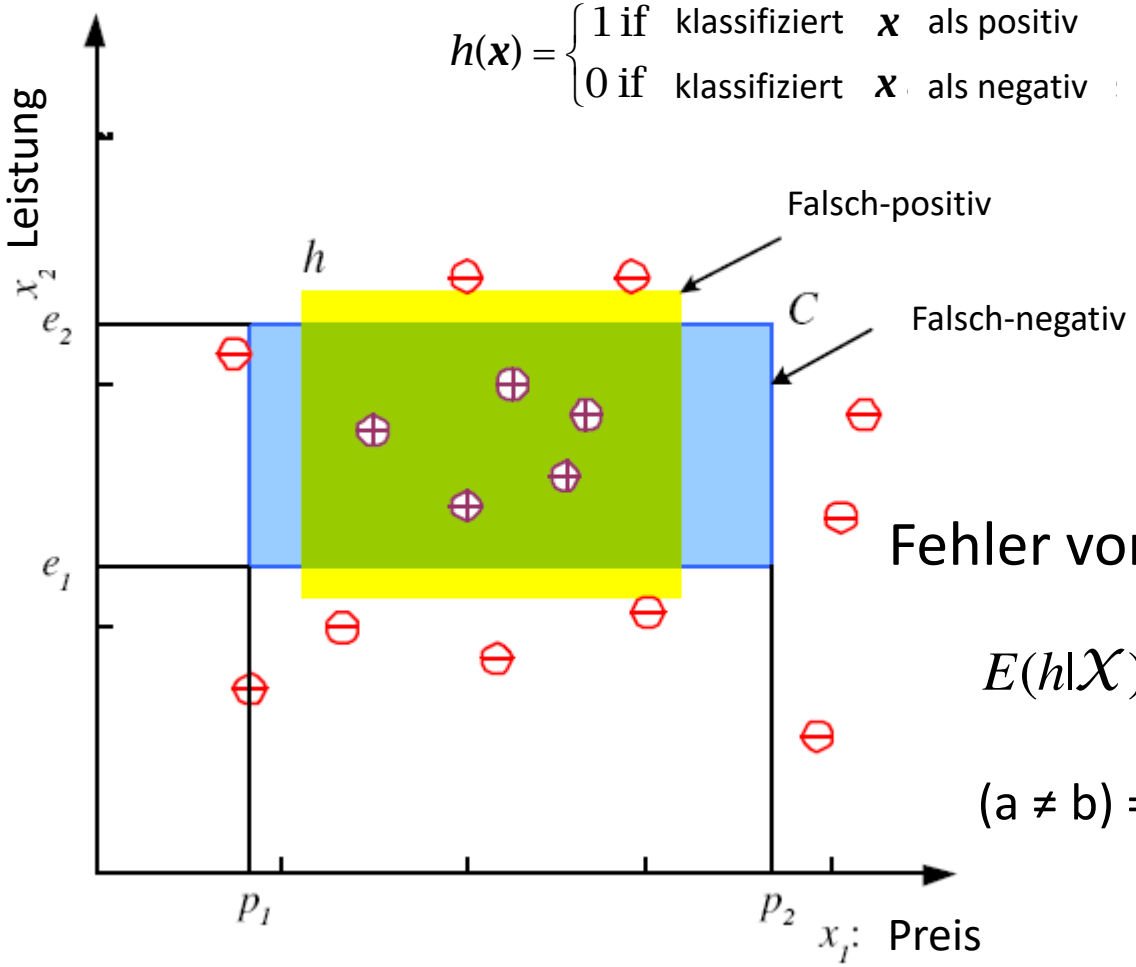
Richtige Klasse C

$(p_1 \leq \text{Preis} \leq p_2)$ und $(e_1 \leq \quad \text{Leistung} \quad \leq e_2)$



Hypothesenklasse $\mathcal{H}$ (z.B. $h \in \mathcal{H}$ in gelb)

$$h(\boldsymbol{x}) = \begin{cases} 1 \text{ if } & \text{klassifiziert } \boldsymbol{x} \text{ als positiv} \\ 0 \text{ if } & \text{klassifiziert } \boldsymbol{x} \text{ als negativ} \end{cases}$$

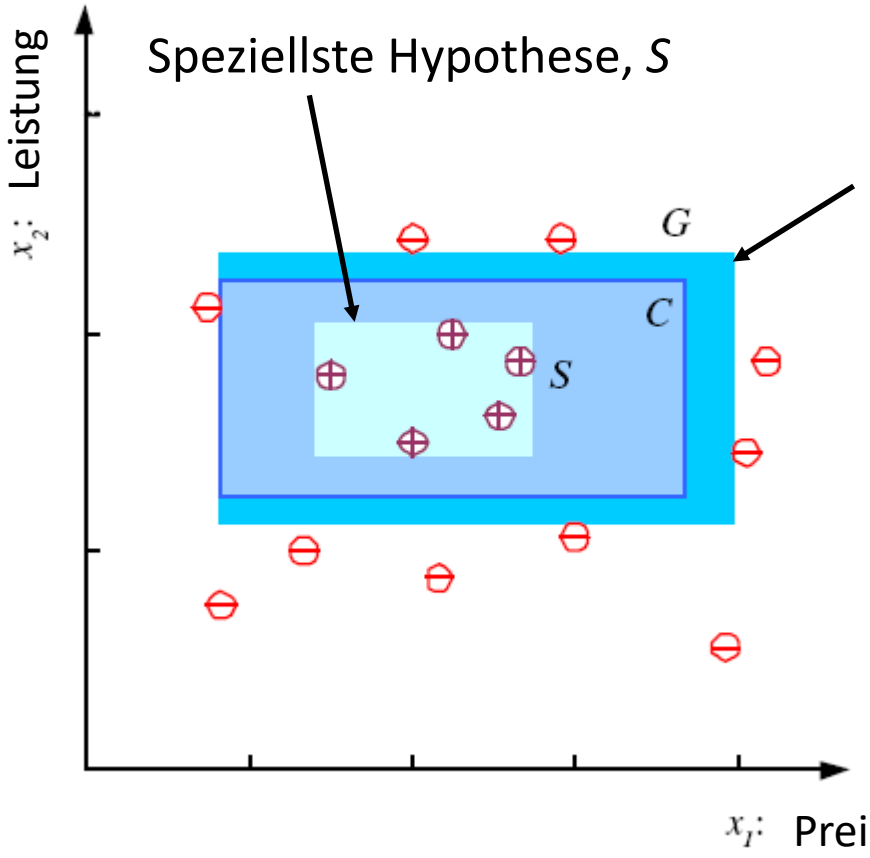


Fehler von h bzgl. $\mathcal{H}$

$$E(h|\mathcal{X}) = (1/N)\sum_{t=1}^{N}\left(h\left(\mathbf{x}^t\right) \neq r^t\right)$$

(a ≠ b) = 1 if ≠, 0 sonst

S, G, and der Versionsraum (Version Space)



$h \in H$, zwischen S and G ist konsistent und definiert den Versionsraum

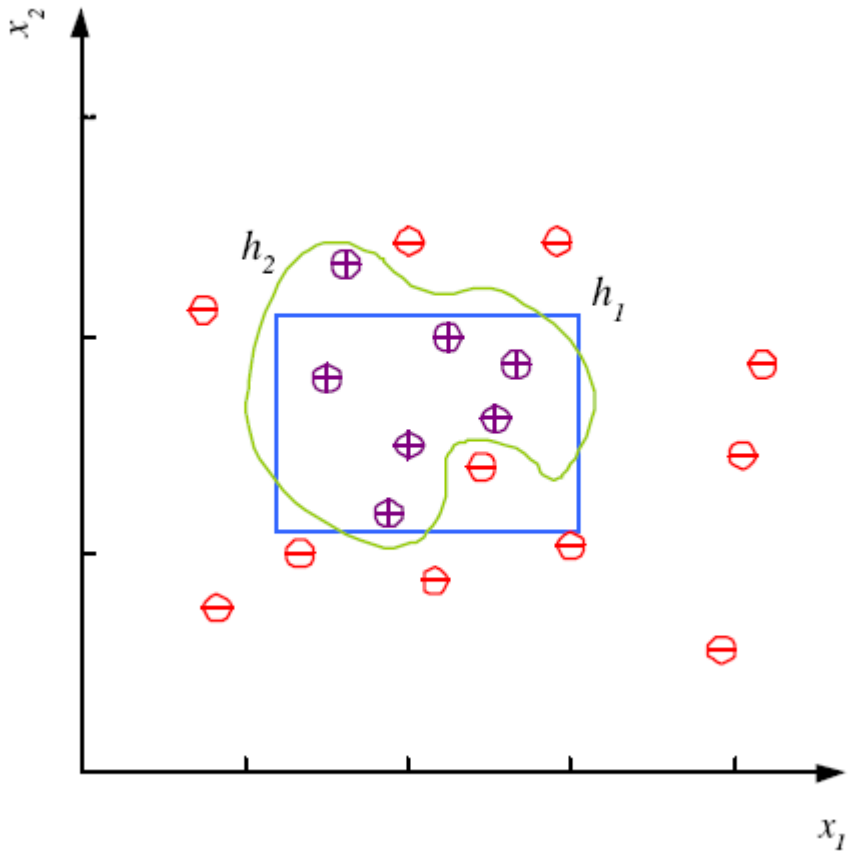
Mitchell, Tom M., Generalization as search. *Artificial Intelligence*. **18** (2): 203–226, **1982**

Rauschen und Modellkomplexität

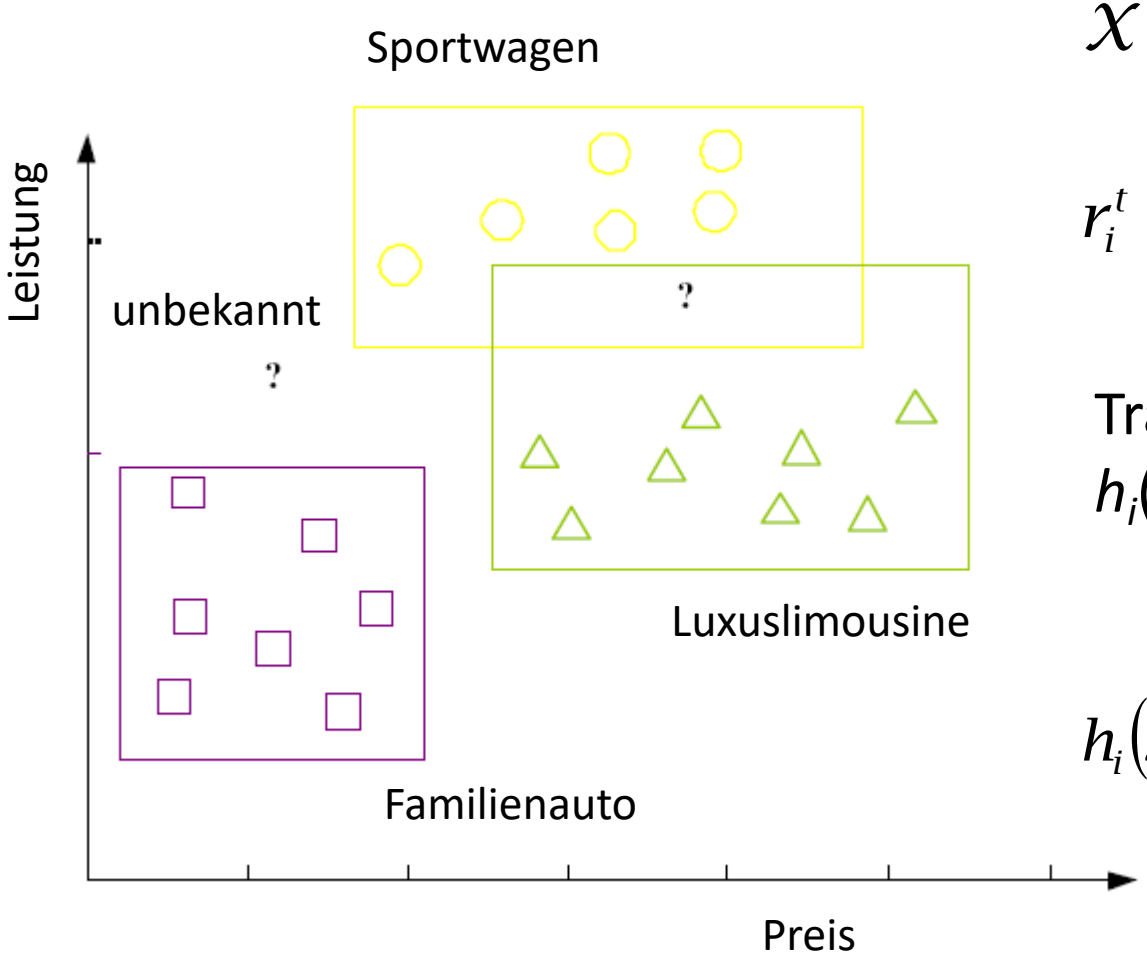
Verwende einfaches Modell:

- Einfacher zu verwenden
  (weniger Berechnungsschnitte)
- Leichter zu trainieren
  (weniger Daten zu speichern)
- Leichter zu erklären
  (besser interpretierbar)
- Bessere Generalisierung
  (Occam's Razor)

Modellkomplexität:
"Größe" der Beschreibung



Clustering: Verschiedene Klassen, C_i i=1,...,K



$$\mathcal{X} = \{\boldsymbol{x}^t, r^t\}_{t=1}^N$$

$$r_i^t = \begin{cases} 1 \text{ if } \boldsymbol{x}^t \in C_i \\ 0 \text{ if } \boldsymbol{x}^t \in C_j, j \neq i \end{cases}$$

Trainiere Hypothesen $h_i(\boldsymbol{x})$, i =1,...,K:

$$h_i\left(\boldsymbol{x}^t\right) = \begin{cases} 1 \text{ if } \boldsymbol{x}^t \in C_i \\ 0 \text{ if } \boldsymbol{x}^t \in C_j, j \neq i \end{cases}$$

Ausgleichsprobleme: Verschiedene Modellklassen

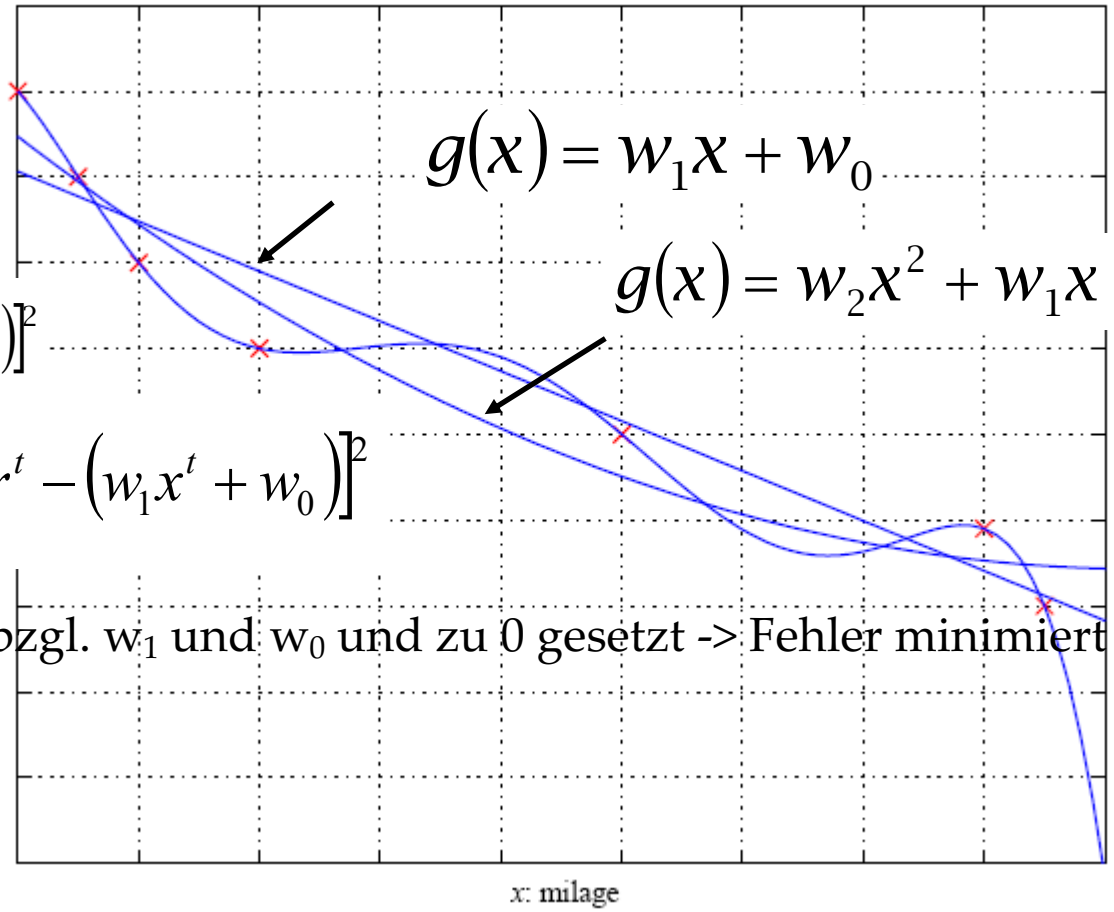
$$\mathcal{X} = \left\{ x^t, r^t \right\}_{t=1}^{N}$$

$$r^t \in \Re$$

$$r^t = f\left(x^t\right)$$

$$E(g \mid \mathcal{X}) = \frac{1}{N} \sum_{t=1}^{N} \left[r^t - g\left(x^t\right) \right]^2$$

$$E\left(w_1, w_0 \mid \mathcal{X}\right) = \frac{1}{N} \sum_{t=1}^{N} \left[r^t - \left(w_1 x^t + w_0 \right) \right]^2$$



Partielle Ableitungen von E bzgl. w_1 und w_0 und zu 0 gesetzt -> Fehler minimiert

$$w_1 = \frac{\sum_t x^t r^t - \overline{xr} N}{\sum_t (x^t)^2 - N \overline{x}^2}$$

$$w_0 = \overline{r} - w_1 \overline{x}$$

Überwachtes Lernen

- Beispiel: Ausgleichsrechnung (Regression)
  - Gegeben Datenpunkte, bestimme Parameter, so dass für gegebene x-Werte, die y-Werte geschätzt werden können
  - Optimierungsproblem Minimierung eines Normmaßes

$$\|x\|_p := \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}$$

- Beispiel: Klassifikation
  - Gegebene Datenpunkte jeweils mit Klassifikationswert, bestimme Klassifikationswert für Datenpunkte ohne diesen (binärer oder mehrwertiger Klassifikator)
  - Kann als Spezialfall eines Ausgleichs gesehen werden

Lineare Ausgleichsprobleme

Beispiel

Zu den Wertepaaren

x_i	1	2	3	4
y_i	6	6.8	10	10.5

soll die Ausgleichsgerade bestimmt werden.

Lösung: Gesucht ist die Ausgleichsgerade in der Form $y = ax + b$, also $\mathcal{F} := \{a_1 f_1 + a_2 f_2 | a_1, a_2 \in \mathbb{R}\}$ mit den Ansatzfunktionen $f_1(x) = x$ und $f_2(x) = 1$.

Das Fehlerfunktional hat dann die Form

$$E(f(a,b)) := \sum_{i=1}^{4} (y_i - f(x_i))^2 = \sum_{i=1}^{4} (y_i - (ax_i + b))^2$$

Lineare Ausgleichsprobleme

Das Fehlerfunktional $E(f(a,b))$ soll minimal werden, d.h. die partiellen Ableitungen nach den Parametern a und b müssen verschwinden:

$$0 = \frac{\partial E(f(a,b))}{\partial a} = -2\sum_{i=1}^{n}(y_i - (ax_i + b))x_i$$

$$0 = \frac{\partial E(f(a,b))}{\partial b} = -2\sum_{i=1}^{n}(y_i - (ax_i + b))$$

Dies liefert 2 Gleichungen für die beiden Unbekannten a und b. Nach wenigen Umformungen erhält man:

$$a\sum_{i=1}^{n}x_i^2 + b\sum_{i=1}^{n}x_i = \sum_{i=1}^{n}y_i x_i$$

$$a\sum_{i=1}^{n}x_i + b\sum_{i=1}^{n}1 = \sum_{i=1}^{n}y_i$$

79 / 904

Lineare Ausgleichsprobleme

Dies ist ein lineares Gleichungssystem für die beiden Unbekannten a und b, das sich in Matrix-Vektor-Form schreiben lässt als:

$$\begin{pmatrix} \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & n \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} y_i x_i \\ \sum_{i=1}^{n} y_i \end{pmatrix}$$

Nach Einsetzen der Werte für x_i und y_i erhält man:

$$\begin{pmatrix} 30 & 10 \\ 10 & 4 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 91.6 \\ 33.3 \end{pmatrix} \quad \Rightarrow \quad a = 1.67, b = 4.15$$

Die Ausgleichsgerade ist: $y = 1.67x + 4.15$.

Lineare Ausgleichprobleme

Definition

Gegeben sind Basisfunktionen $f_1, \ldots, f_m$,
$\mathcal{F} := \{\lambda_1 f_1 + \lambda_2 f_2 + \ldots \lambda_m f_m | \lambda_i \in \mathbb{R} \text{ für alle } i = 1, \ldots, m\}$
sowie n Wertepaare $(x_i, y_i), i = 1, \ldots, n$. Man sagt, dass ein **lineares Ausgleichsproblem** vorliegt.

Weiter sei für $f = \sum_{j=1}^{m} \lambda_j f_j \in \mathcal{F}$:

$$
\begin{aligned}
E(\lambda_1, \lambda_2, \ldots, \lambda_m) &:= \sum_{i=1}^{n} (y_i - f(x_i))^2 \\
&= \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{m} \lambda_j f_j(x_i) \right)^2 \\
&= ||\mathbf{y} - \mathbf{A}\boldsymbol{\lambda}||_2^2.
\end{aligned}
$$

Lineare Ausgleichsprobleme

Hierbei ist

$$\mathbf{A} = \begin{pmatrix} f_1(x_1) & f_2(x_1) & \dots & f_m(x_1) \\ f_1(x_2) & f_2(x_2) & \dots & f_m(x_2) \\ \vdots & \vdots & \vdots & \vdots \\ f_1(x_n) & f_2(x_n) & \dots & f_m(x_n) \end{pmatrix} \text{ und}$$

$$\mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \boldsymbol{\lambda} = \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_m \end{pmatrix}.$$

Das System $\mathbf{A}\,\boldsymbol{\lambda} = \mathbf{y}$ heißt **Fehlergleichungssystem**.

Normalgleichungen

Definition

Die Gleichungen

$$0 = \frac{\partial E(f(\lambda_1, \lambda_2, \ldots, \lambda_m))}{\partial \lambda_i}, \quad i = 1, \ldots, m$$

heißen **Normalgleichungen** zum linearen Ausgleichsproblem.

Das System der Normalgleichungen heißt **Normalgleichungssystem**; es lässt sich als lineares Gleichungssystem in der Form

$$\mathbf{A}^T \mathbf{A} \boldsymbol{\lambda} = \mathbf{A}^T \mathbf{y}$$

schreiben.

Lineare Ausgleichsprobleme

Beispiel

Gesucht ist das lineare Gleichungssystem bestehend aus den Normalgleichungen zu den Wertepaaren:

x_i	1	2	3	4
y_i	6	6.8	10	10.5

Lösung: Gegeben sind $f_1(x) = x$, $f_2(x) = 1$ und $n = 4$ Wertepaare. Die Matrix $\mathbf{A}$ ist also eine 4×2 Matrix

$$\mathbf{A} = \begin{pmatrix} f_1(x_1) & f_2(x_1) \\ f_1(x_2) & f_2(x_2) \\ f_1(x_3) & f_2(x_3) \\ f_1(x_4) & f_2(x_4) \end{pmatrix} = \begin{pmatrix} f_1(1) & f_2(1) \\ f_1(2) & f_2(2) \\ f_1(3) & f_2(3) \\ f_1(4) & f_2(4) \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \end{pmatrix}$$

Lineare Ausgleichsprobleme

$$\Rightarrow \mathbf{A}^T\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \end{pmatrix} = \begin{pmatrix} 30 & 10 \\ 10 & 4 \end{pmatrix},$$

$$\mathbf{A}^T\mathbf{y} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 6 \\ 6.8 \\ 10 \\ 10.5 \end{pmatrix} = \begin{pmatrix} 91.6 \\ 33.3 \end{pmatrix}$$

Damit ist $\mathbf{A}^T\mathbf{A}\boldsymbol{\lambda} = \mathbf{A}^T\mathbf{y}$ dasselbe Gleichungssystem wie aus dem vorherigen Beispiel.

Lineare Ausgleichsprobleme

Beispiel

Gegeben sind die Messdaten

x_i	0	1	2	3	4
y_i	6	12	30	80	140

Gesucht ist eine Funktion $f(x) = ae^x + b$, die die Daten bestmöglich bzgl. der kleinsten Fehlerquadrate approximiert.

Lösung: Die Ansatzfunktionen lauten $f_1(x) = e^x$ und $f_2(x) = 1$. Das Fehlergleichungssystem hat dann die Form:

$$\mathbf{A}\boldsymbol{\lambda} = \mathbf{y} \Longleftrightarrow \begin{pmatrix} 1.0 & 1.0 \\ 2.7183 & 1.0 \\ 7.3891 & 1.0 \\ 20.0865 & 1.0 \\ 54.5982 & 1.0 \end{pmatrix} \boldsymbol{\lambda} = \begin{pmatrix} 6 \\ 12 \\ 30 \\ 80 \\ 140 \end{pmatrix}$$

Lineare Ausgleichsprobleme

Hieraus ergibt sich das zugehörige Normalgleichungssystem

$$\mathbf{A}^T\mathbf{A}\boldsymbol{\lambda} = \mathbf{A}^T\mathbf{y} \Longleftrightarrow \begin{pmatrix} 3447.37 & 85.79 \\ 85.79 & 5.0 \end{pmatrix} \boldsymbol{\lambda} = \begin{pmatrix} 9510.88 \\ 268.0 \end{pmatrix}$$

Dieses System hat die Lösung

$$\begin{pmatrix} 2.49 \\ 10.93 \end{pmatrix}$$

und damit die Lösung für die Ausgleichsfunktion
$f(x) = 2.49e^x + 10.93.$

Penalized Least Squares (PLS)

- $$E(\lambda_1, \lambda_2, \ldots, \lambda_m) := \sum_{i=1}^{n} (y_i - f(x_i))^2$$

$$E(\boldsymbol{\lambda}) = \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{m} \lambda_j f_j(x_i) \right)^2$$

- $PLS(\boldsymbol{\lambda}) = E(\boldsymbol{\lambda}) + \alpha \cdot \text{pen}(\boldsymbol{\lambda})$ zu minimieren nach $\boldsymbol{\lambda}$
- pen($\boldsymbol{\lambda}$) misst Komplexität der Regressionkoeffizienten
- Glättungsparameter α misst Einfluss von pen($\boldsymbol{\lambda}$)

Regularisierung bei der Regression

- Ridge Regression (First, Grat, Bergrücken)
  - $\text{pen}(\lambda_1, \lambda_2, \ldots, \lambda_m) = \Sigma_{j=1}^{m} \lambda_j^2$
  - Schrumpfung der Koeffizienten *gegen* 0
- LASSO (Least Absolute Shrinkage and Selection Operator)
  - $\text{pen}(\lambda_1, \lambda_2, \ldots, \lambda_m) = \Sigma_{j=1}^{m} |\lambda_j|$
  - Schrumpfung der Koeffizienten *auf* 0
  - Verwendung zur Konstruktion möglichst einfacher Modelle
- Bei allen Regularisierungsverfahren ist die Annahme, dass die Koeffizienten bzgl. Ihrer Wert vergleichbar sind

Modellauswahl & Generalisierung

- Lernen kann als Optimierungsproblem angesehen werden:
  - Berechne Modell, so dass Fehlerfunktion minimiert
  - Parameter für Repräsentation berechnet → "Parametrisches Lernen"
- Lernen ist ein schlecht gestelltes Problem
  - In der Regel reichen die Daten nicht, eine eindeutige Lösung des Optimierungsproblems zu finden
  - Vorannahmen treffen: Annahmen bzgl. H (inductive bias)
  - Regularisierung: Repräsentation für alle optimalen Lösungen
- Generalisierung: Wie gut arbeitet Modell auf neuen Daten?
- Überanpassung (Overfitting): H komplexer als C oder f
- Unteranpassung (Underfitting): H weniger komplex als C oder f

Drei-Wege-Austauschbeziehung

(Dietterich, 2003):

1. Komplexität von $\mathcal{H}$, $c(\mathcal{H})$,
2. Trainingsmengengröße *N*,
3. Generalisierungsfehler, *E*, auf neuen Daten

- Wenn $N\uparrow$, $E\downarrow$
- Wenn $c(\mathcal{H})\uparrow$, gilt zuerst $E\downarrow$ und dann $E\uparrow$

Kreuzvalidierung

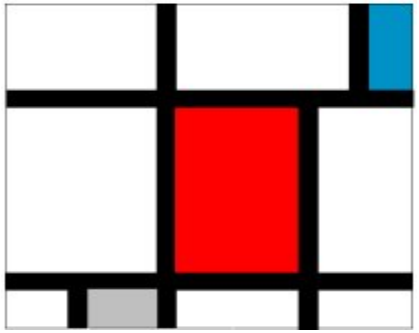
- Um den Generalisierungsfehler abzuschätzen, brauchen wir Daten, mit denen nicht trainiert wurde
- Aufteilung der Daten:
  - Trainingsmenge (50%)
  - Validierungsmenge (25%)
  - Testmenge (z.B. für Publikation) (25%)
- Neuabtastung, wenn wenige Daten vorhanden

Betrachtungsebenen überwachtes Lernen

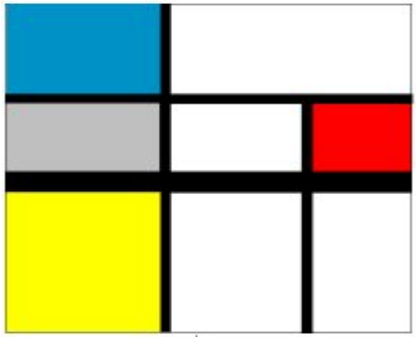
1. Modell: $g(\boldsymbol{x} \mid \theta)$

2. Fehlerfunktion (Verlustfunktion): $E(\theta \mid \mathcal{X}) = \sum_t L(r^t, g(\boldsymbol{x}^t \mid \theta))$

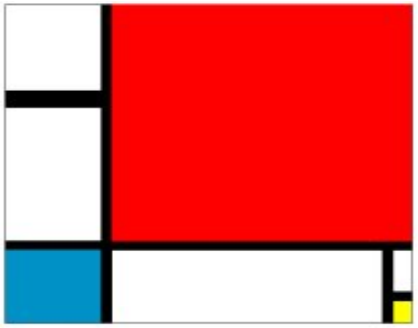
3. Optimierungs verfahren: $\theta^* = \arg\min_\theta E(\theta \mid \mathcal{X})$



1

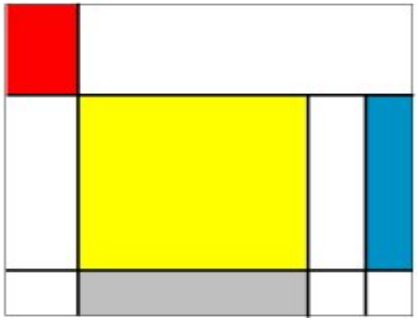


2



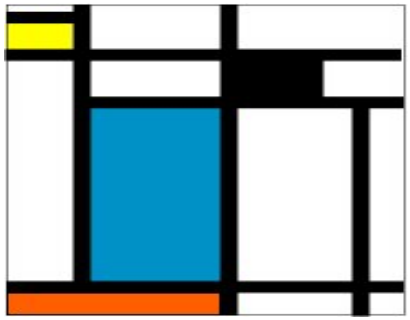
3

4

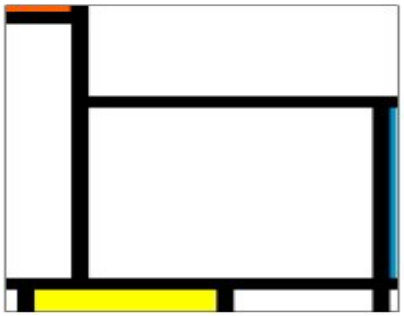


5

6



7



8 ?

Daten in Tabellarischer Form

Number	Lines	Line types	Rectangles	Colours	Mondrian?
1	6	1	10	4	*No*
2	4	2	8	5	*No*
3	5	2	7	4	Yes
4	5	1	8	4	Yes
5	5	1	10	5	*No*
6	6	1	8	6	Yes
7	7	1	14	5	*No*

Anfrage

Number	Lines	Line types	Rectangles	Colours	Mondrian?
8	7	2	9	4	

Analyse von Daten

- Betrachtung einer Spalte x mit n Werten
- Bestimmung des Mittelwerts: $\overline{x} = 1/n \cdot \sum_{i=1}^{n} x_i$
- Große und kleine Werte können sich aufheben
- Mittlere Abweichung vom Mittelwert betrachten (Varianz)
- Bestimmung der Varianz: $var = 1/n \cdot \sum_{i=1}^{n} (x - \overline{x}_i)^2$
- Meist betrachtet wird die sog. Standardabweichung: $\sigma = \sqrt{var}$

Halte Daten in normalisierter Form vor

Eine Möglichkeit zur Normalisierung:

$$x_t' \equiv \frac{x_t - \bar{x}_t}{\sigma_t}$$

Gemittelte Abweichung vom Mittel

Normalisierte Trainingsdaten

Number	Lines	Line types	Rectangles	Colours	Mondrian?
1	0.632	-0.632	0.327	-1.021	*No*
2	-1.581	1.581	-0.588	0.408	*No*
3	-0.474	1.581	-1.046	-1.021	Yes
4	-0.474	-0.632	-0.588	-1.021	Yes
5	-0.474	-0.632	0.327	0.408	*No*
6	0.632	-0.632	-0.588	1.837	Yes
7	1.739	-0.632	2.157	0.408	*No*

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{t=1}^{T} [x_{it} - x_{jt}]^2}$$

Number	Lines	Line types	Rectangles	Colours	Mondrian?
8	1.739	1.581	-0.131	-1.021	

Normalisierte Trainingsdaten

Number	Lines	Line types	Rectangles	Colours	Mondrian?
1	0.632	-0.632	0.327	-1.021	*No*
2	-1.581	1.581	-0.588	0.408	*No*
3	-0.474	1.581	-1.046	-1.021	Yes
4	-0.474	-0.632	-0.588	-1.021	Yes
5	-0.474	-0.632	0.327	0.408	*No*
6	0.632	-0.632	-0.588	1.837	Yes
7	1.739	-0.632	2.157	0.408	*No*

$$\sqrt{(0 + 4{,}89 + 5{,}23 + 2{,}04)} = 3{,}489$$

Number	Lines	Line types	Rectangles	Colours	Mondrian?
8	1.739	1.581	-0.131	-1.021	

Distanz der Testinstanz von den Trainingsdaten

Beispiel	Distanz zum Test	Mondrian?
1	2.517	*No*
2	3.644	*No*
3	2.395	Yes
4	3.164	Yes
5	3.472	*No*
6	3.808	Yes
7	3.490	*No*

Klassifikation

1-NN	Yes
3-NN	Yes
5-NN	*No*
7-NN	*No*

Was verwenden wir bei reellwertiger Zielfuntion als Ausgabe?

- Mittel der k-nächsten Nachbarn

Variante von kNN: Distanzgewichtetes kNN

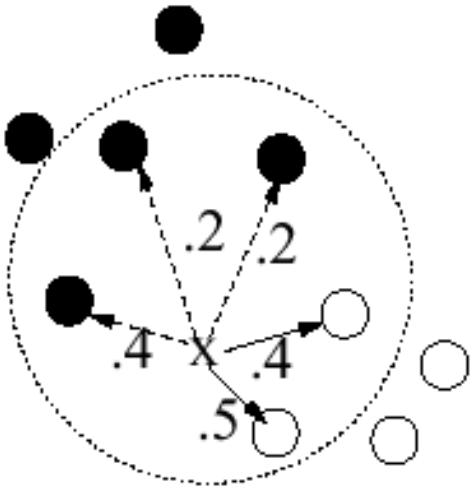
- Nähere Nachbarn haben mehr Einfluss

$$f(\mathbf{x}_q) := \frac{\sum_{i=1}^{k} w_i f(\mathbf{x}_i)}{\sum_{i=1}^{k} w_i} \quad \text{where } w_i = \frac{1}{d(\mathbf{x}_q, \mathbf{x}_i)^2}$$

Dudani, S.A. The distance-weighted k-nearest-neighbor rule. IEEE Trans. Syst. Man Cybern., SMC-6:325–327, **1976**

Variante von kNN: Distanzgewichtetes kNN

k-NN using a weighted-sum voting scheme



kNN (k = 5)

Assign "white" to x because the weighted sum of "whites" is larger then the sum of "blacks".

Each neighbor is given a weight according to its nearness.

Dann könnten wir statt nur k gleich alle Trainingsinstanzen (= Beispiele) nehmen

kNN: Zusammenfassung

- Sehr einfacher Ansatz, nicht-parametrisch
  - Klassifikation (ggf. mit Schwellwert)
  - Regression
- Verhält sich auch noch gutartig, wenn Daten nicht einfach separiert werden können
- Rang 7 der 10 wichtigsten Data-Mining-Verfahren