Einführung in Web- und Data-Science

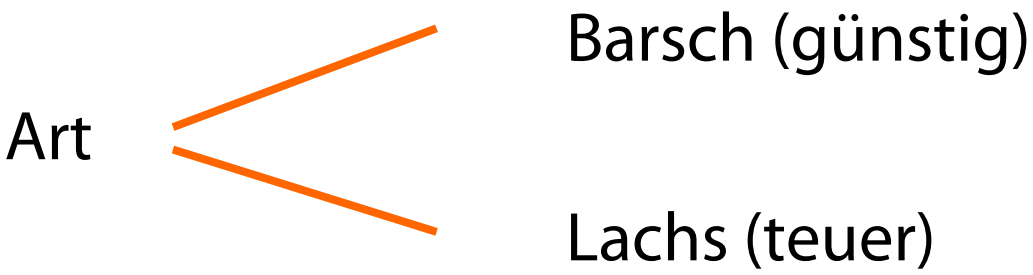
Prof. Dr. Ralf Möller

Universität zu Lübeck

Institut für Informationssysteme

Ein Anwendungsbeispiel

"Sortierung von Fischen auf einem Förderband nach Arten durch Bildverarbeitung"



Problemanalyse

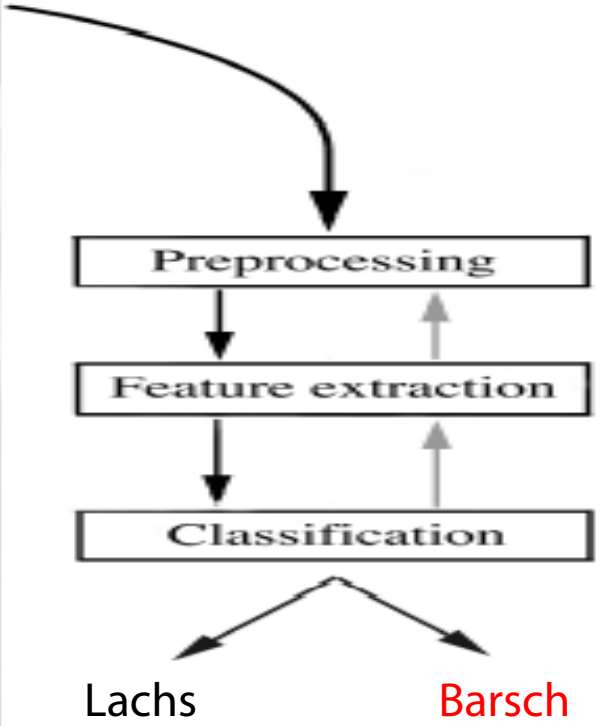
Verwende Kamera und nehme Bilder auf,
um Merkmale zu bestimmen:

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

Menge aller möglichen Merkmale

Ziel: Wähle die relevanten aus

Klassifikation



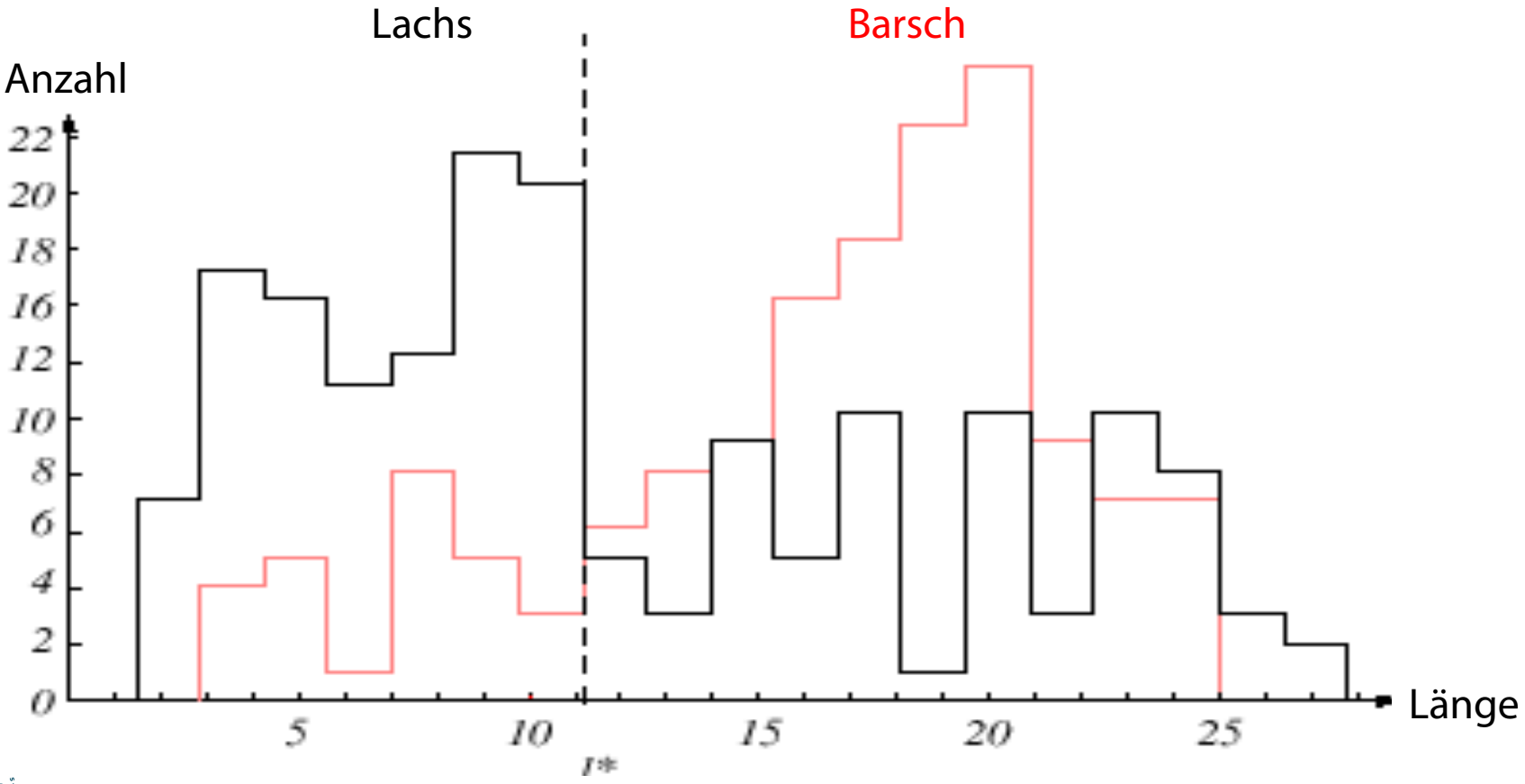
Bestimmung geeigneter Merkmale

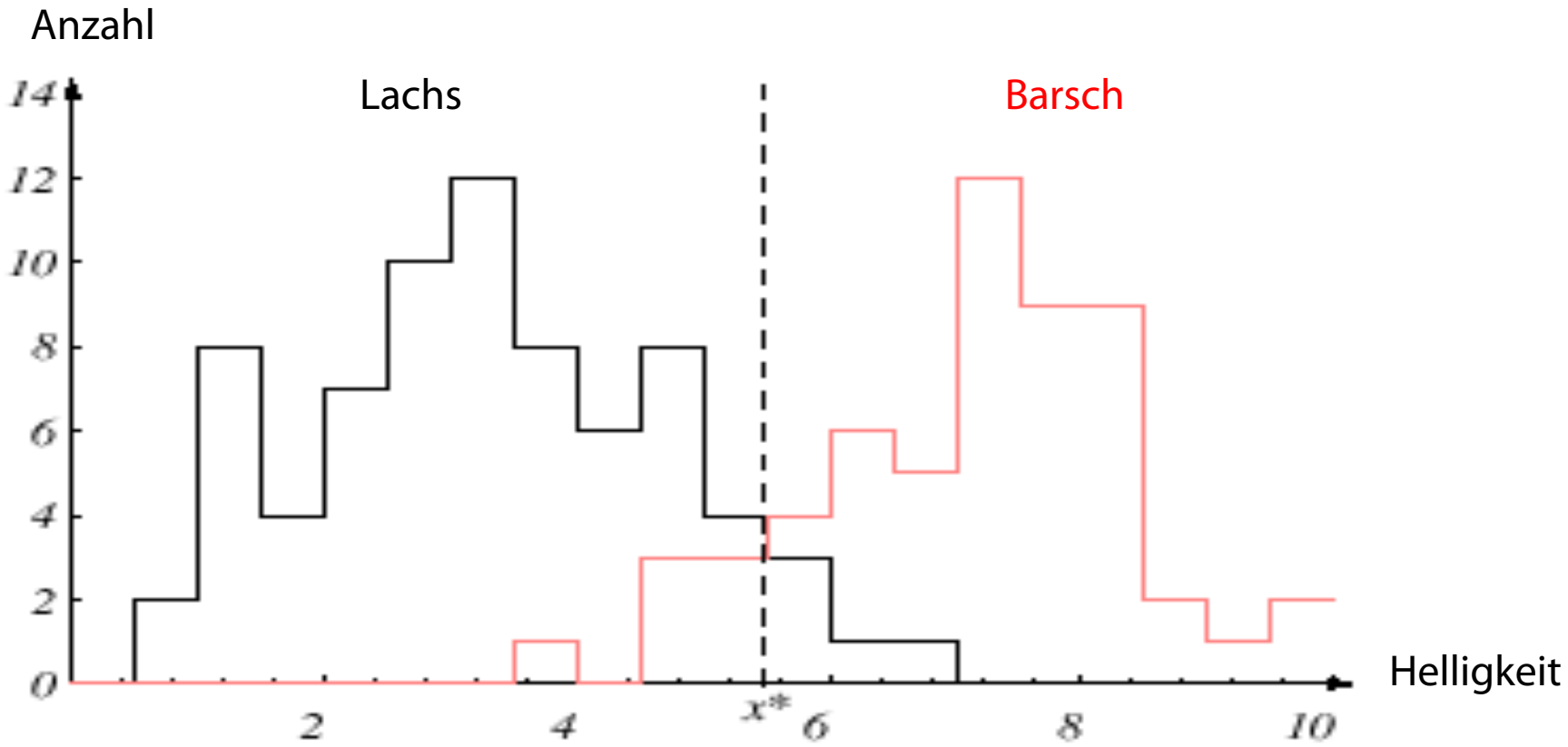
- Wir benötigen einen Experten, um die Merkmale festzulegen, mit denen man Barsche und Lachse richtig klassifizieren kann
- Wie wäre es mit Länge als Merkmal zur Unterscheidung?

Länge allein ist kein gutes Merkmal!

→ Hohe Kosten bei Fehlentscheidung

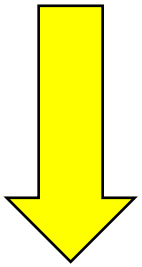
Wie wäre es mit Helligkeit?




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

Schwellwert-Entscheidungsgrenze und induzierte Kosten

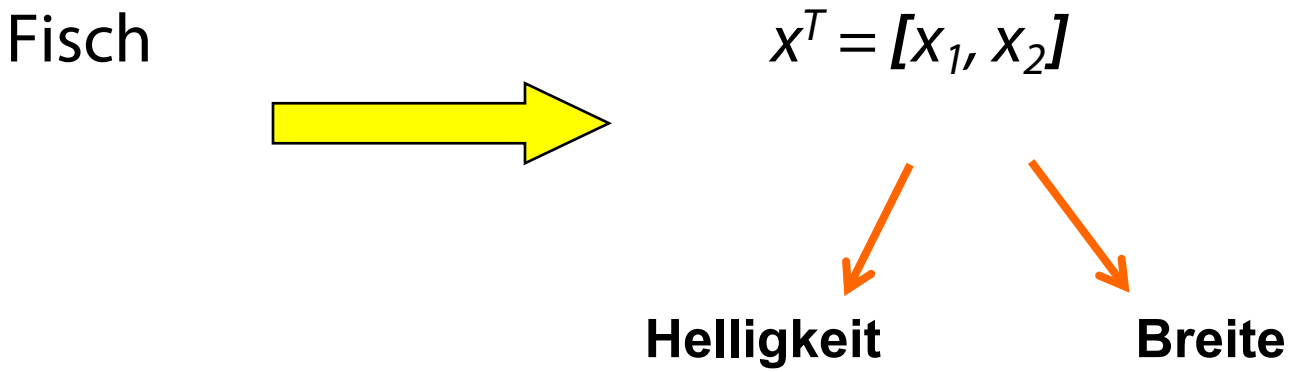
– Schwellwert-Entscheidungsgrenze in Richtung mittlerer Helligkeitswerte minimiert die Kosten (der Fehlklassifikation)

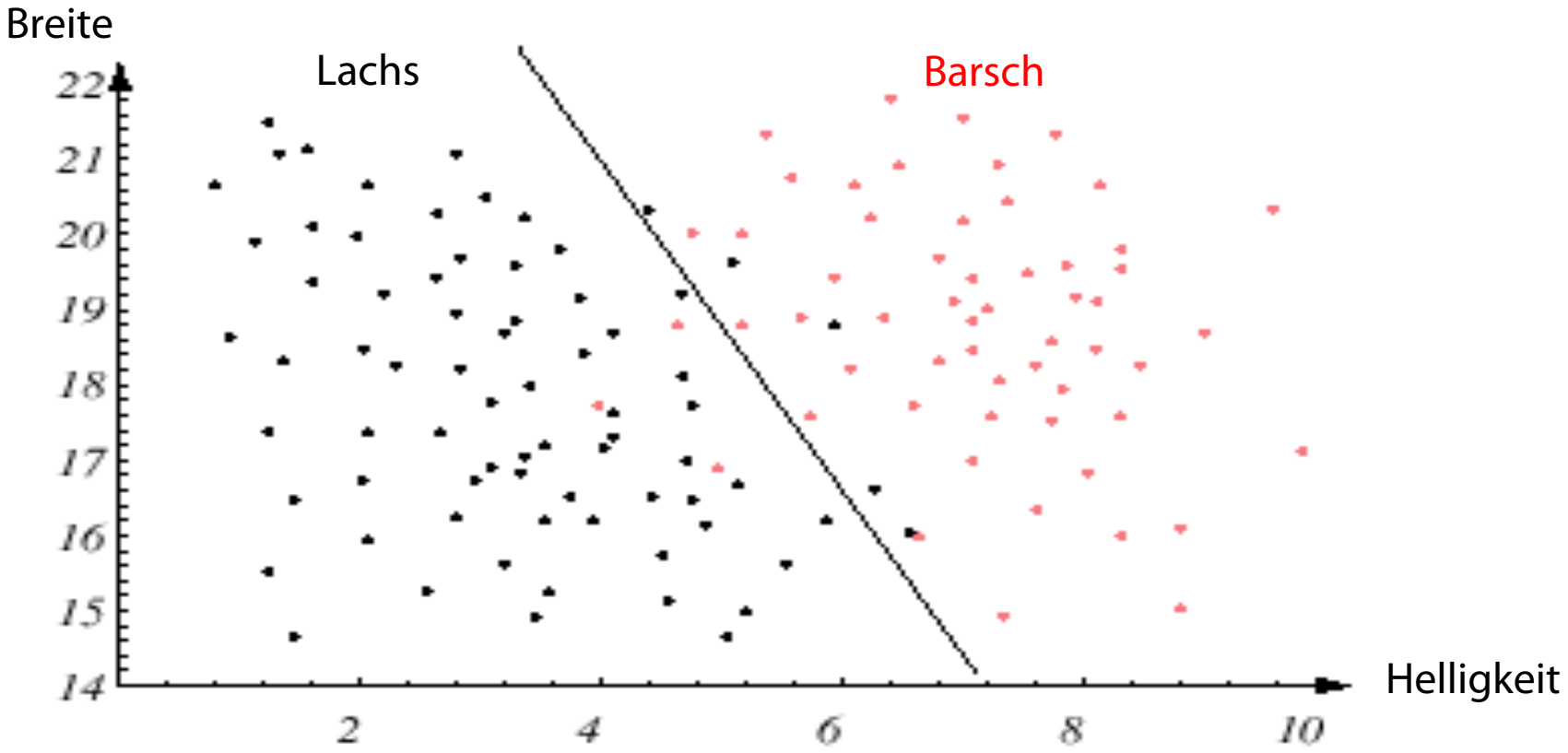


Untersuchung in der sog. Entscheidungstheorie

Ziel ist die automatische Bestimmung von Berechnungsfunktionen für geeignete Merkmale

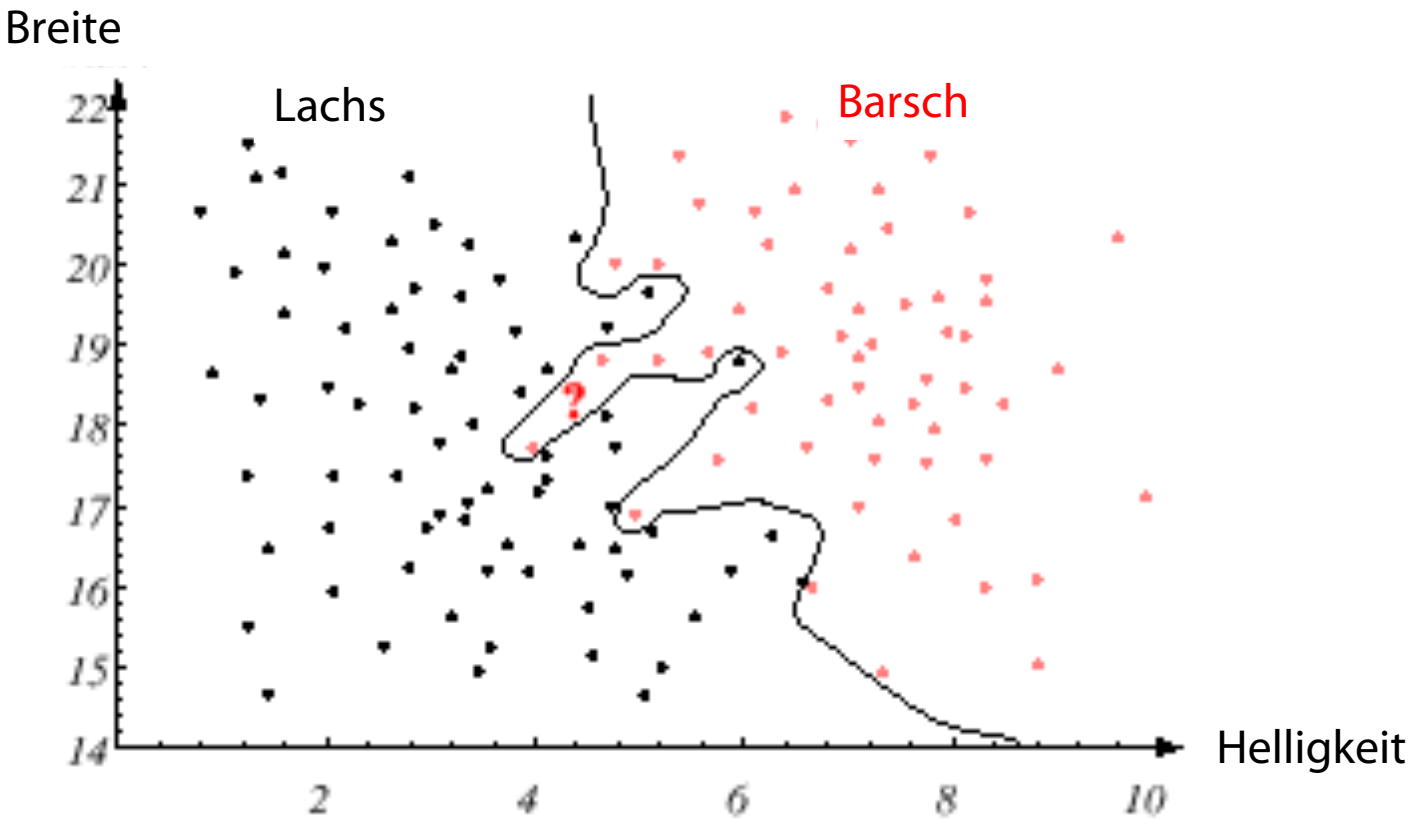
Helligkeit und zusätzlich Breite des Fisches?




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

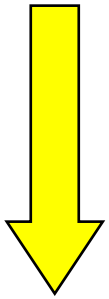
- Weitere Merkmale, die nicht direkt zu Helligkeit und Breite in Beziehung stehen, könnten hinzukommen
  - Vorsicht aber vor Reduktion durch "verrauschte Merkmale"

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

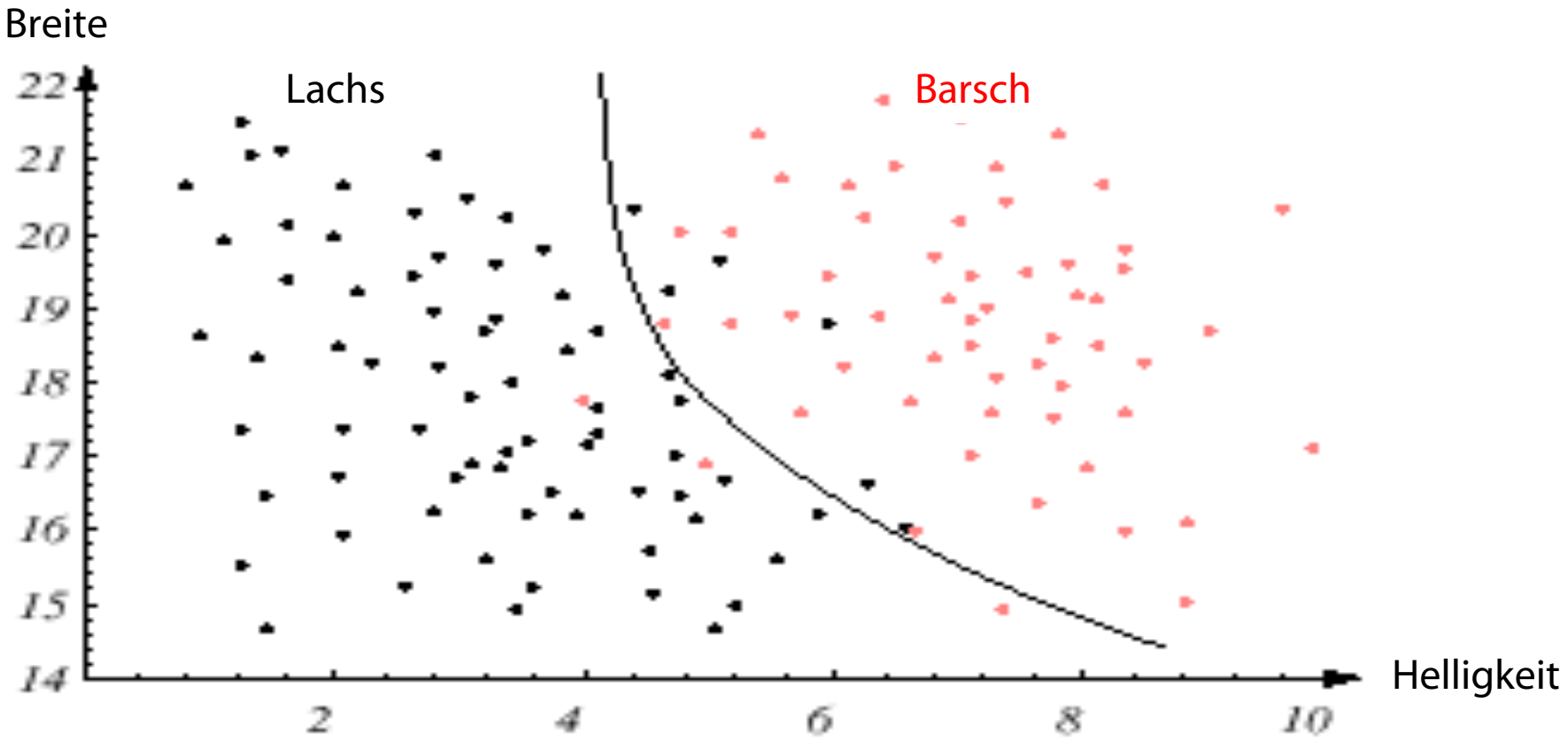

Breite
Lachs
Barsch
?
22
21
20
19
18
17
16
15
14
2
4
6
8
10
Helligkeit

Vorfreude über Klassifikationsleistung auf Testdaten kann verfrüht sein

Wichtig ist Leistung auf neuen Daten!



Generalisierungsfähigkeit zählt!


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

Support-Vektor Maschinen

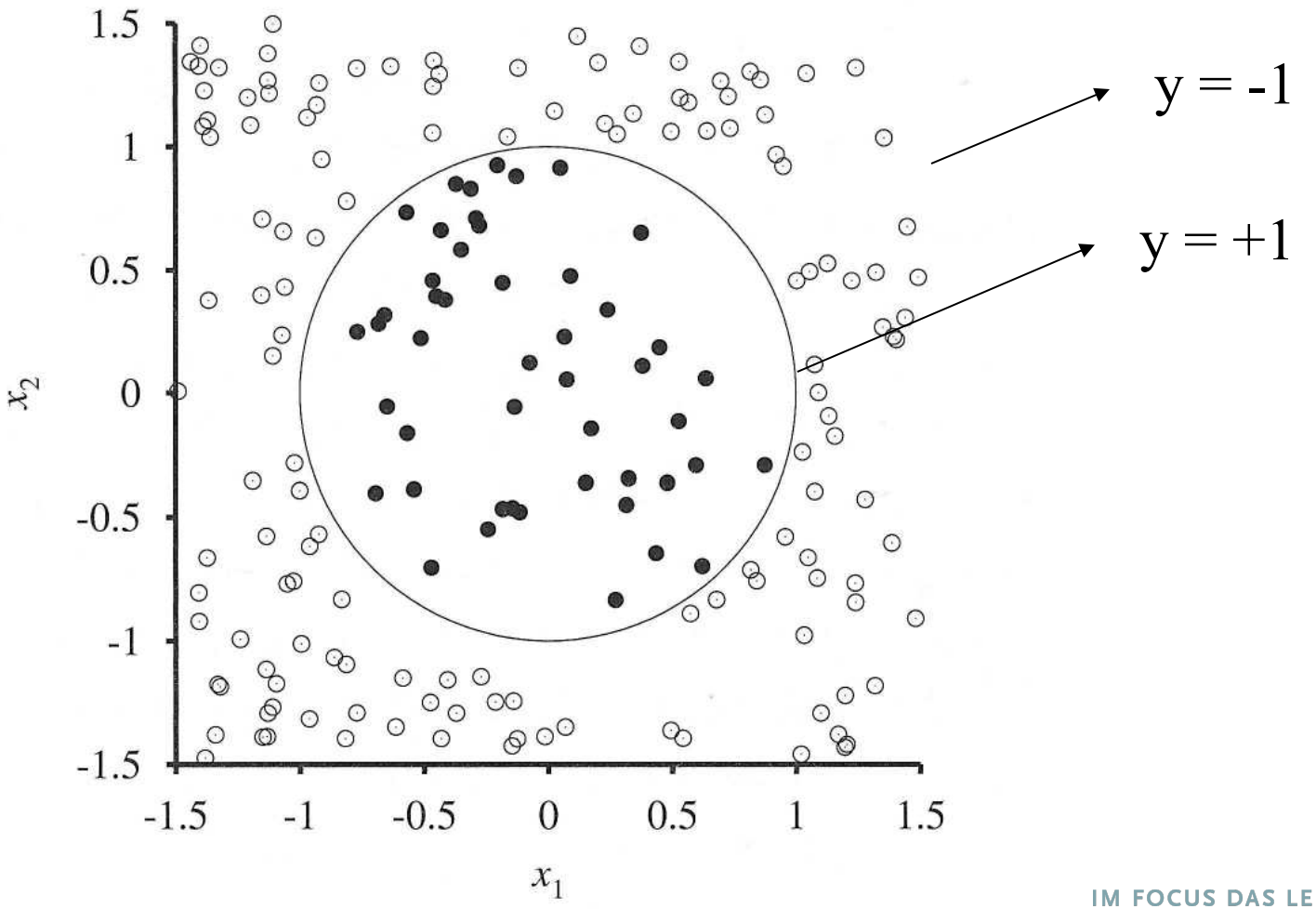
- Abbildung von Instanzen von zwei Klassen in einen Raum, in dem sie linear separierbar sind
  - Abbildungsfunktion heißt Kernel-Funktion
- Berechnung einer Trennfläche über Optimierungsproblem (und nicht iterativ wie bei Perzeptrons und mehrschichtigen Netzen)
  - Formulierung als Problem nicht Verfahren!

V. Vapnik, A. Chervonenkis, A note on one class of perceptrons. *Automation and Remote Control*, **25**, **1964**

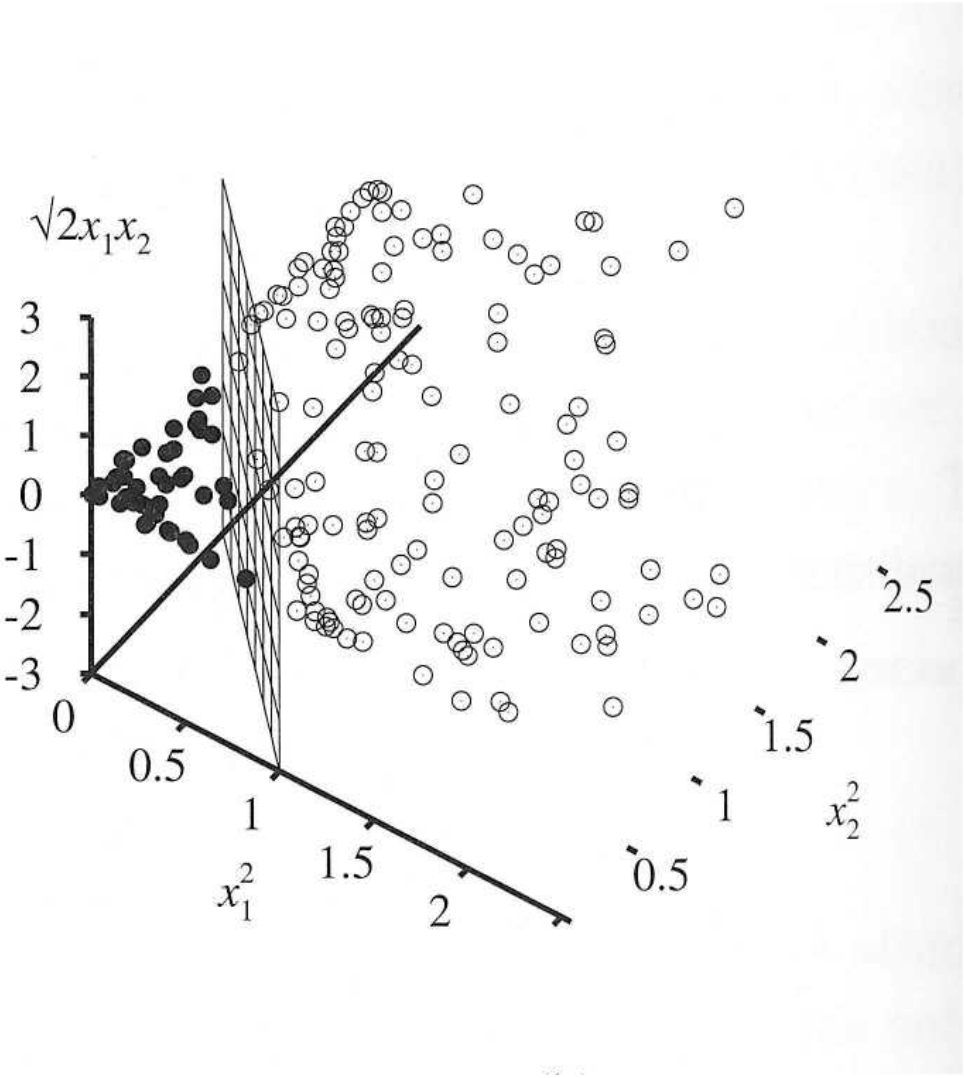
Boser, B. E.; Guyon, I. M.; Vapnik, V. N., A training algorithm for optimal margin classifiers. *Proceedings of the fifth annual workshop on Computational learning theory – COLT '92*. p. 144, **1992**

Vapnik, V., Support-vector networks, Machine Learning. 20 (3): 273–297, **1995**

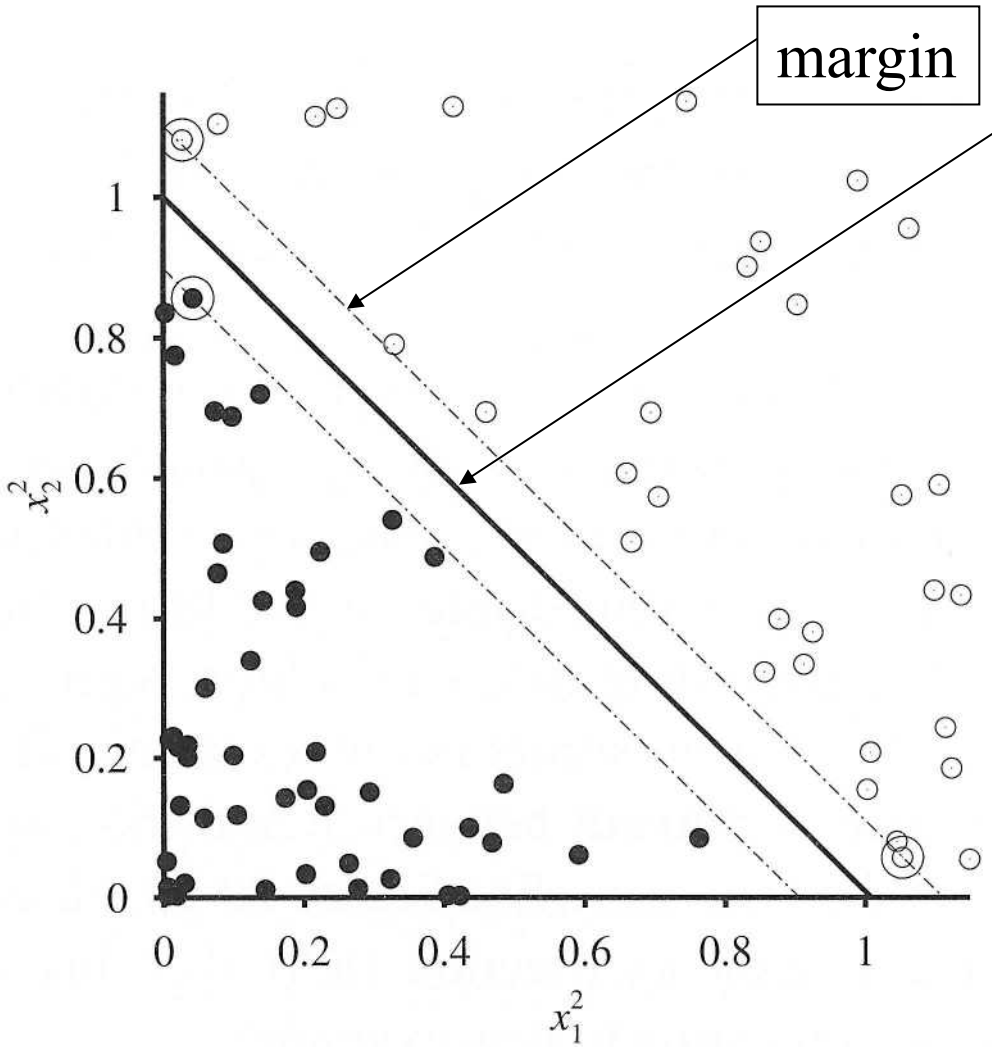
Nichtlineare Separierung



$$(x_1^2, x_2^2, \sqrt{2x_1x_2})$$



Support-Vektoren



Support-Vektoren werden über Optimierungsproblem bestimmt

support vectors

Multi-class SVMs?

- SVMs für mehrere Klassenlabel
- Kombination mehrerer SVMs
  - Einer-gegen-alle
    - Für jede Klasse gibt es eine Entscheidungsgrenze zwischen den „eigenen“ Datenpunkten und allen anderen Datenpunkten
  - Alle-gegen-alle
    - Für jede mögliche Kombination von zwei Klassen gibt es eine Entscheidungsgrenze

Klassifikatorentwicklung

- Relevante Merkmale bzw. Kombination davon automatisch bestimmbar?
  - Wir behandeln das: Deep Learning, Ensembles
- Dynamische Angepassung des Klassifikators möglich?
  - Ohne spezielle Daten mit bekannten Ausgaben (Groundtruth)
    - Transduktives Lernen
- Übertragung eines Klassifikators auf neue Anwendung?
  - Forelle vs. Lachs?
    - Transfer-Lernen

Bewertung eines Klassifikators

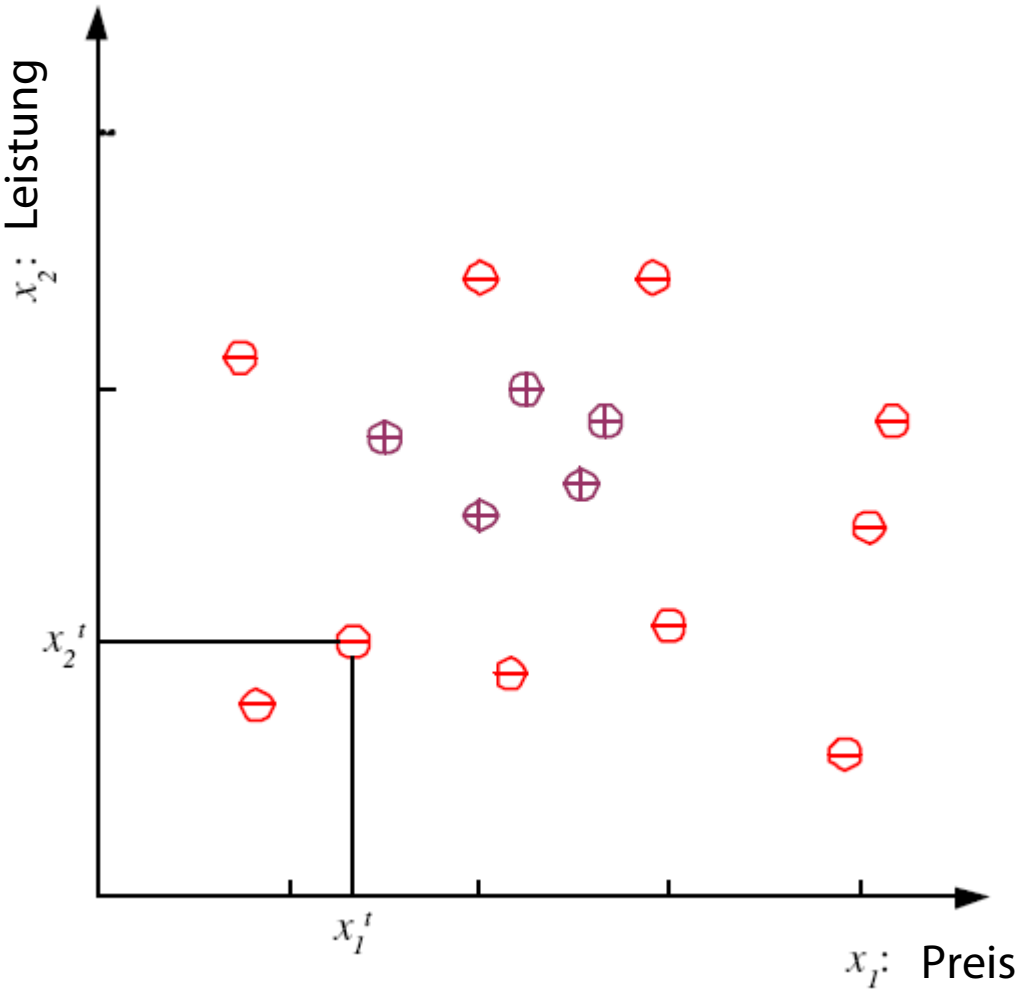
- Klasse C eines “Familienautos”
  - Vorhersage: Ist Auto x ein Familienauto?
  - Wissensextraktion:
    Was erwarten Menschen von einem Familienauto?
- Ausgabe:

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

  x_1: Preis, x_2 : Leistung

Frage: Wie gut funktioniert ein bestimmter Klassifikator?

Trainingsmenge $\mathcal{X}$



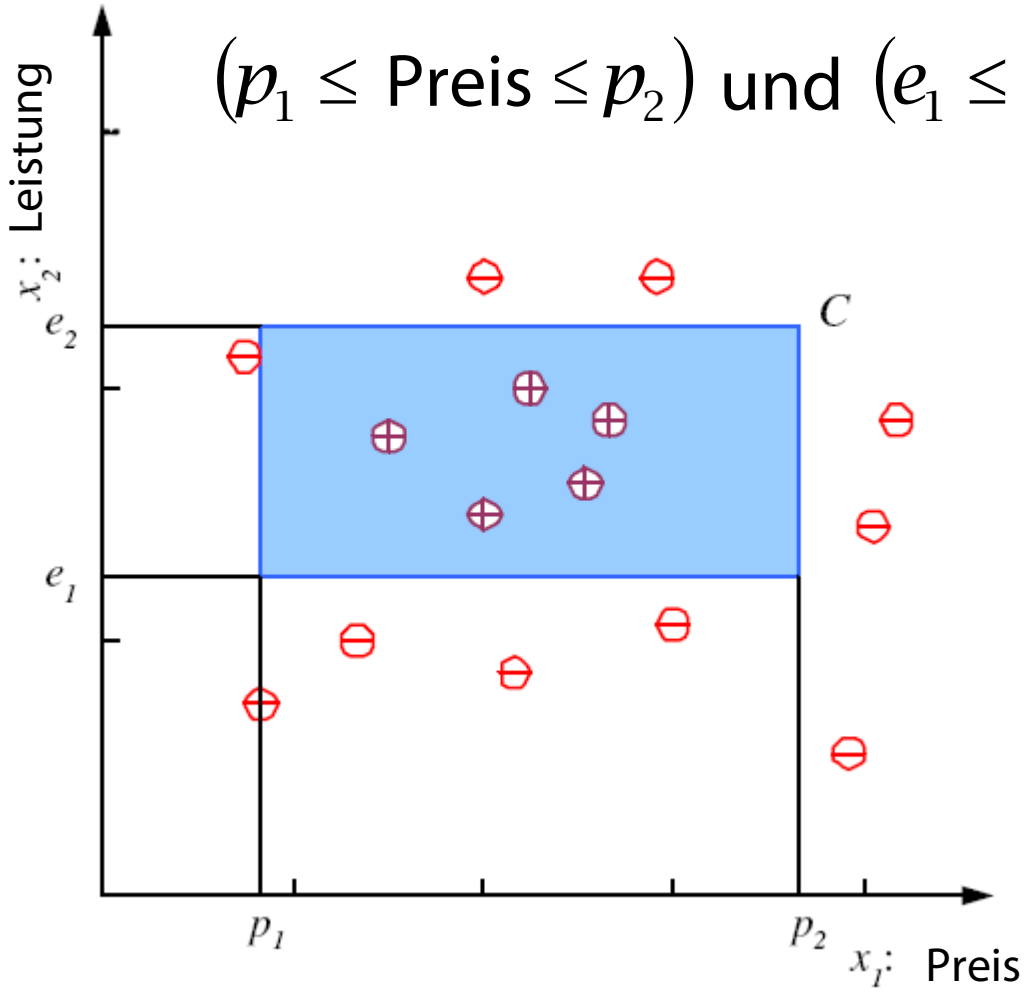
$$\mathcal{X} = \{\boldsymbol{x}^t, r^t\}_{t=1}^N$$

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

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

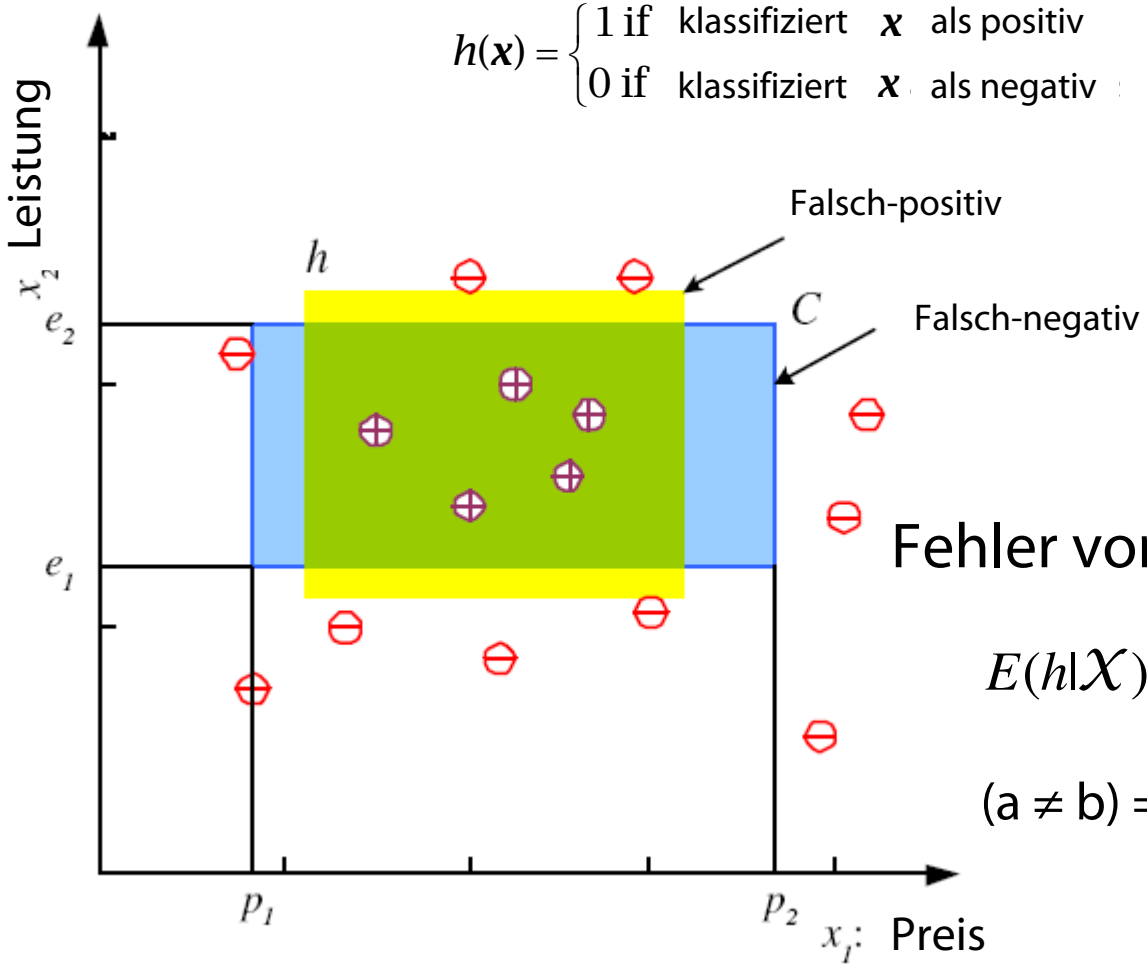
Richtige Klasse C

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



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

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

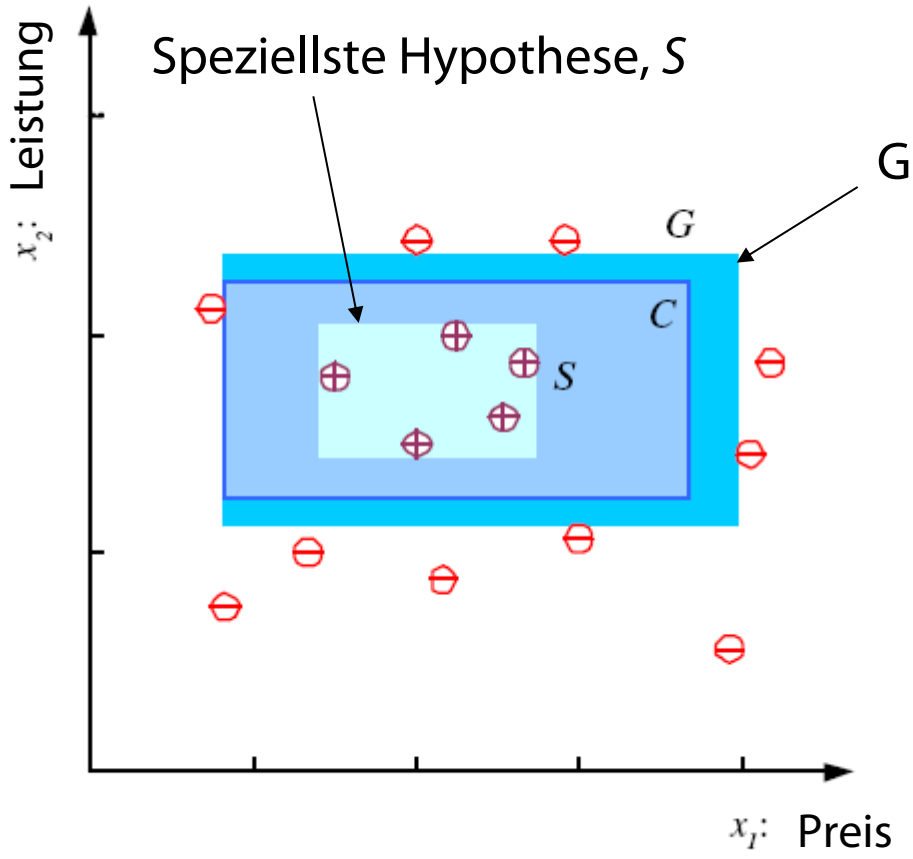


Fehler von h bzgl. $\mathcal{H}$

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

(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

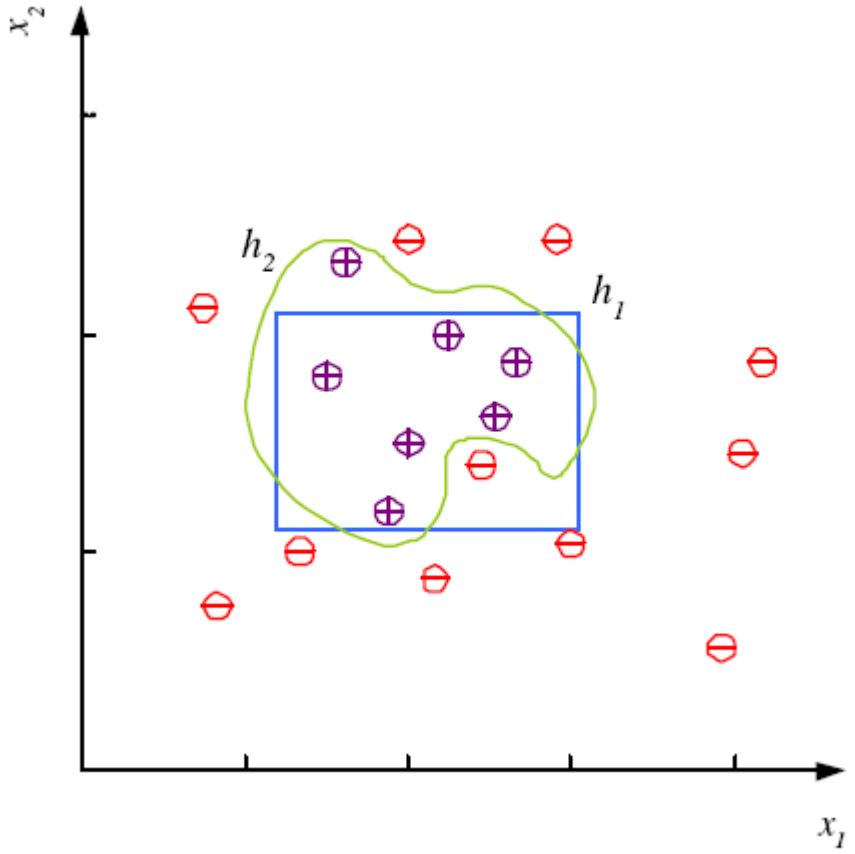
Mitchell, Tom M., Generalization as search. *Artificial Intelligence*. **18** (2): 203–226, **1982**

Rauschen und Modellkomplexität

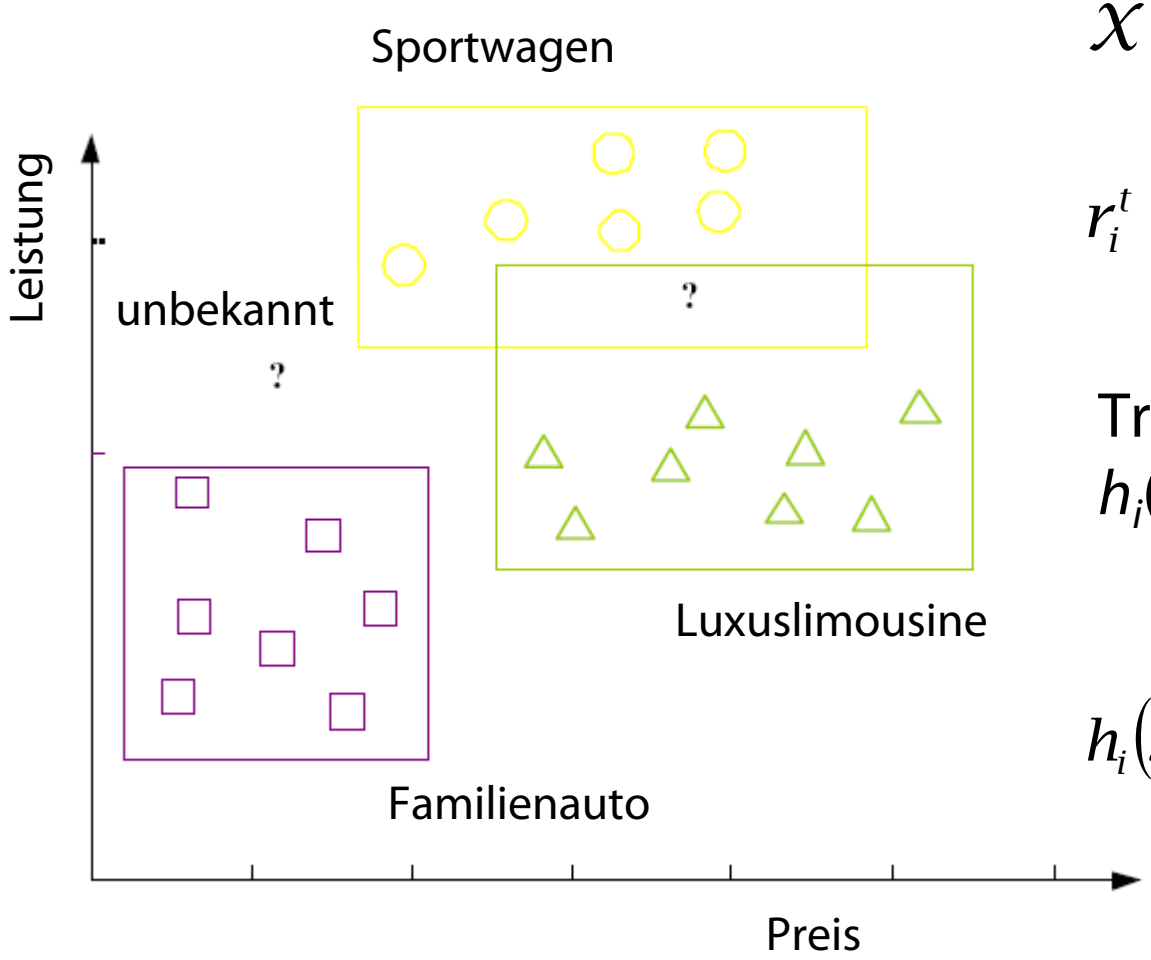
Verwende einfaches Modell:

- 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



Clusterbildung: 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

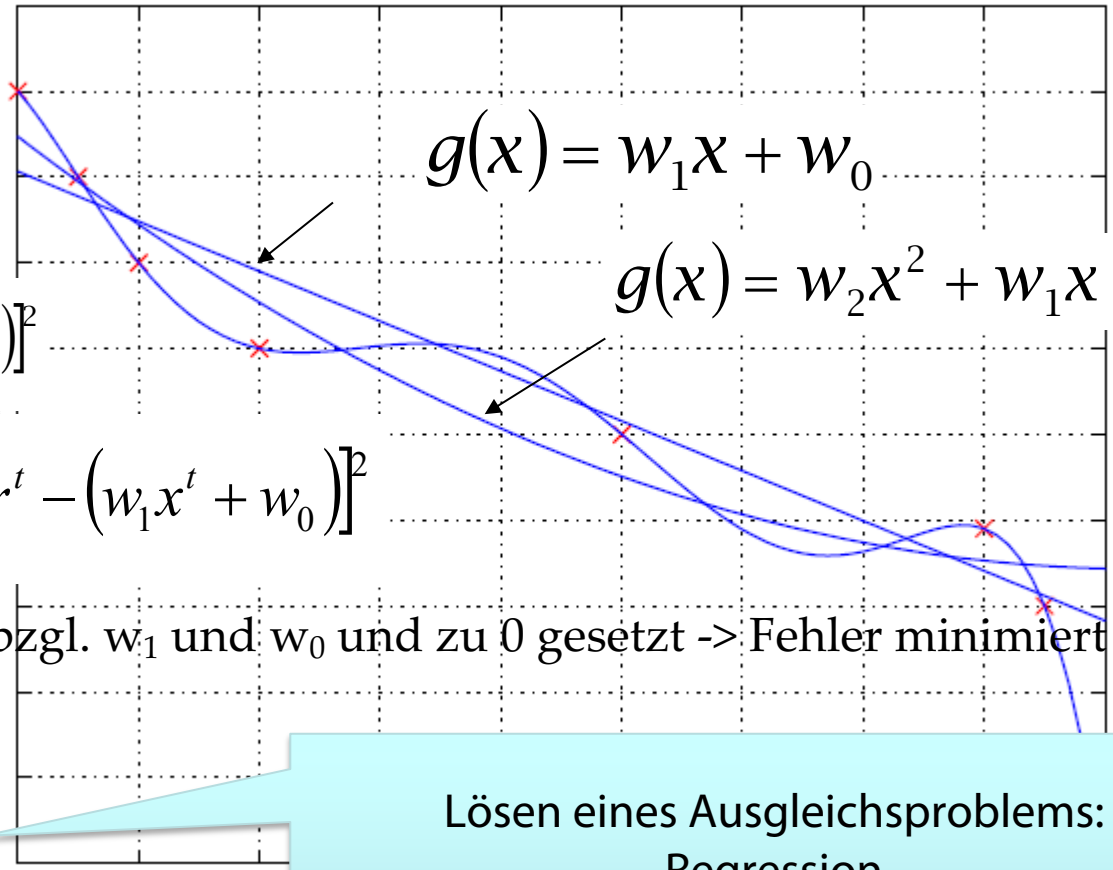
$$\mathcal{X} = \left\{ x^t, r^t \right\}_{t=1}^{N}$$

$$r^t \in \Re$$

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

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

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



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

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

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

Lösen eines Ausgleichsproblems: Regression

Regularisierung

Bisheriger Ansatz: Least Squares Error

$$E(\lambda_1, \lambda_2, \ldots, \lambda_m) \quad := \quad \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{m} \lambda_j f_j(x_i) \right)^2$$

Einführung von Bestrafungstermen (Penalized Least Squares):

- $PLS(\lambda_1, \lambda_2, \ldots, \lambda_m) := E(\lambda_1, \lambda_2, \ldots, \lambda_m) + \alpha \cdot \text{pen}(\lambda_1, \lambda_2, \ldots, \lambda_m)$
  - zu minimieren nach $\boldsymbol{\lambda} = \lambda_1, \lambda_2, \ldots, \lambda_m$
  - pen($\boldsymbol{\lambda}$) misst Komplexität der Regressionskoeffizienten
  - Glättungsparameter α misst Einfluss von pen($\boldsymbol{\lambda}$)

- Genannt: Regularisierung

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 Werte vergleichbar sind

Zusammenfassung: Überwachtes Lernen

- Beispiel: Ausgleichsrechnung (Regression[1])
  - Gegeben Datenpunkte, bestimme Parameter, so dass für gegebene x-Werte, bei i.A. minimalem die y-Werte geschätzt werden können
  - Optimierungsproblem
    Minimierung eines Normmaßes
- Beispiel: Klassifikation
  - Gegebene Datenpunkte jeweils mit Klassifikationswert, bestimme Klassifikationswert für Datenpunkte ohne diesen (binärer oder mehrwertiger Klassifikator)
  - Kann als Spezialfall der Regression angesehen werden

[1] Regression: Rückgang oder Rückschluss von Funktionswerten auf Funktion

Modellauswahl & Generalisierung

- Lernen kann als Optimierungsproblem angesehen werden:
  - Berechne Modell, so dass Fehlerfunktion minimiert
  - Parameter für Repräsentation berechnen → "Parametrisches Lernen"
- Lernen ist i.A. ein schlecht gestelltes Problem
  - In der Regel sind die Daten nicht geeignet, um eine eindeutige Lösung des Optimierungsproblems zu finden
  - Vorannahmen treffen: Annahmen bzgl. H (inductive bias)
  - Kann man optimale Hypothesenklassen automatisch bestimmen?
- Generalisierung: Wie gut arbeitet Modell auf neuen Daten?
  - Generalisierungsfehler
- Überanpassung (Overfitting): H komplexer als C bzw. f
- Unteranpassung (Underfitting): H weniger komplex als C bzw. f

H = Hypothesenklasse, C = Classifier, f = Regressionsfunktion

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

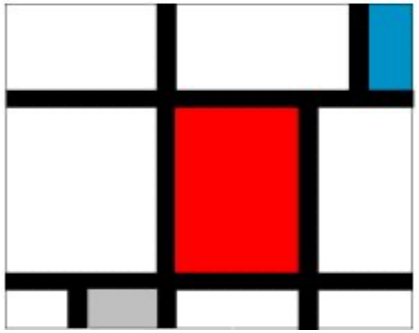
- Um den Generalisierungsfehler abzuschätzen, brauchen wir Daten, mit denen nicht trainiert wurde
- Aufteilung der Daten:
  - Trainingsmenge (50%)
  - Testmenge (z.B. für Publikation) (50%)
- Neuabtastung, wenn wenige Daten vorhanden

Betrachtungsebenen überwachtes Lernen

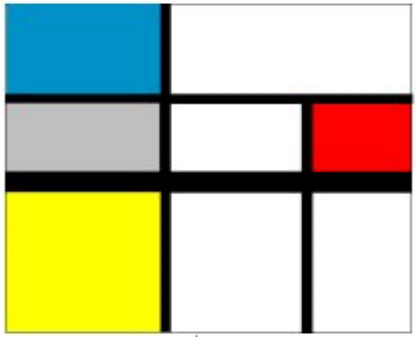
1. Modell: $g(\boldsymbol{x} \mid \theta)$

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

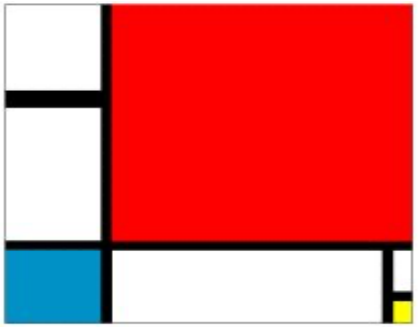
3. Optimierungs verfahren: $\theta^* = \arg\min_\theta E(\theta \mid \mathcal{X})$



1

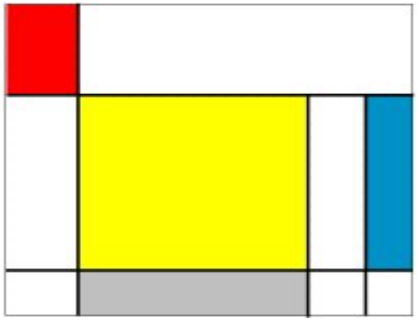


2



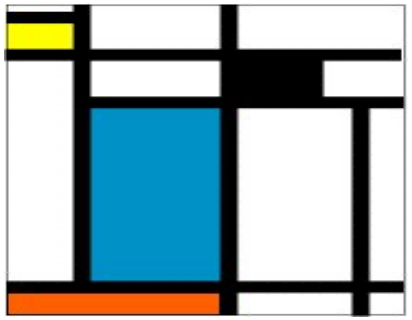
3

4

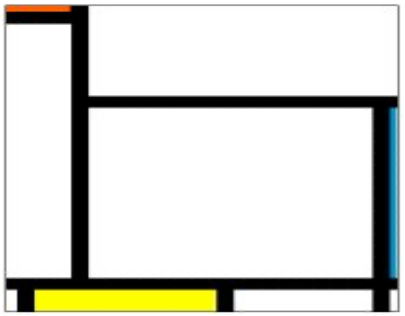


5

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: $\bar{x} = 1/n \cdot \Sigma_{i=1}^{n} x_i$
- Große und kleine Werte können sich aufheben
- Mittlere Abweichung vom Mittelwert betrachten (Varianz)
- Bestimmung der Varianz*: $var = 1/n \cdot \Sigma_{i=1}^{n} (\bar{x}-x_i)^2$
- Meist betrachtet wird die sog. Standardabweichung: $\sigma = \sqrt{var}$

*: Wird später verfeinert.

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

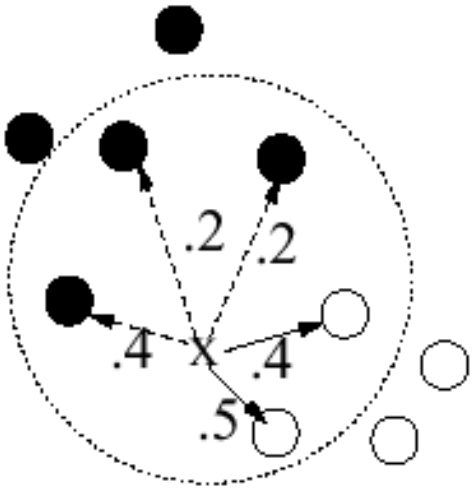
- Nähere Nachbarn haben mehr Einfluss

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

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 im Prinzip gleich alle Trainingsinstanzen (= Beispiele) nehmen

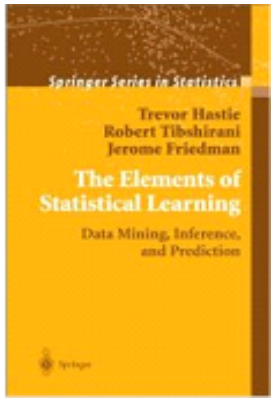
kNN: Zusammenfassung

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

Literatur (1)

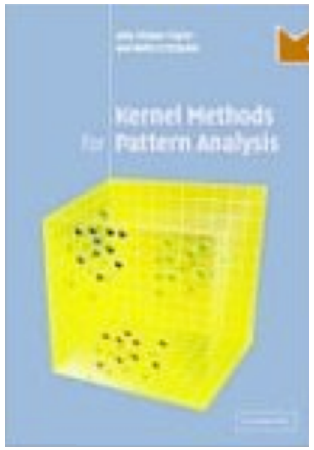
Mitchell (1989). Machine Learning.
http://www.cs.cmu.edu/~tom/mlbook.html

Duda, Hart, & Stork (2000). Pattern Classification.
http://rii.ricoh.com/~stork/DHS.html

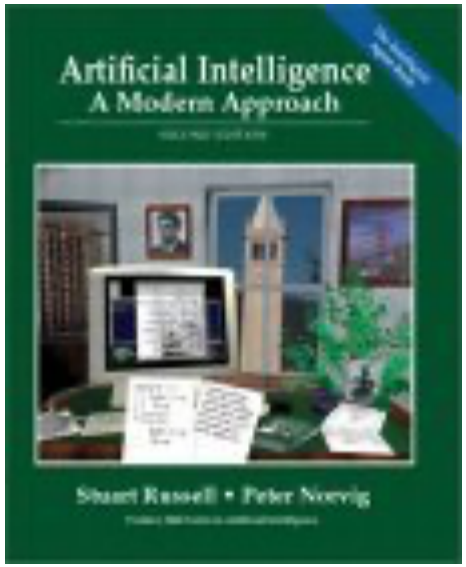


Hastie, Tibshirani, & Friedman (2001). The Elements of Statistical Learning. http://www-stat.stanford.edu/~tibs/ElemStatLearn/

Literatur (2)

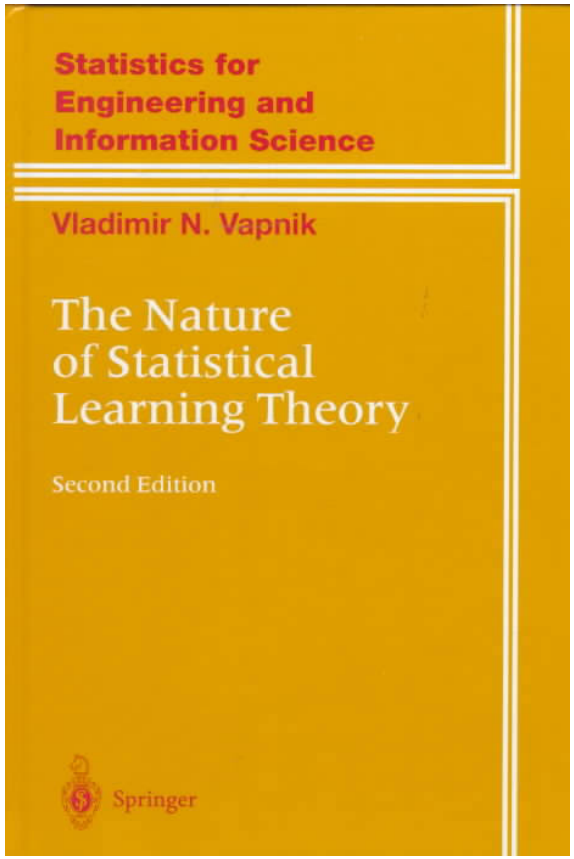


Shawe-Taylor & Cristianini. Kernel Methods for Pattern Analysis.
http://www.kernel-methods.net/



Russell & Norvig (2004). Artificial Intelligence.
http://aima.cs.berkeley.edu/

Originalliteratur SVM



VAPNIK, Vladimir N.,. The Nature of Statistical Learning Theory. Springer-Verlag New York, Inc., **1995**