Big Data Anwendungen

Klassifikation

Klassifikationsprobleme

- Idee
  - Bestimmung einer „Klasse“, d.h. eines unbekannten
    - kategorischen Attributwertes
    - (ordinale Attributwerte mit Einschränkung)
  - Unter Benutzung beliebiger
    - bekannter Attributwerte

					Klasse
	Einkommen	**Ausgaben**	**Haus**	**Alter**	**Kreditausfall**
Lern-daten	10.000 €	15.000 €	1	17	nein
	20.000 €	10.000 €	1	20	nein
	50.000 €	100.000 €	0	55	ja
Vorhersage	10.000 €	15.000 €	0	55	?

- Beispiele:
  - Response auf Werbemittel
  - Vorhersage von Kundenverhalten wie Kündigungen (Churn)
  - Vorhersage von Retouren
  - …

Agenda

- Einführung
- Deskriptive Methoden zur Datenexploration
- Datenqualität
- Klassifikation
  - Naiver Bayes Klassifikator
  - Künstliche Neuronale Netze
  - Entscheidungsbäume
  - Support Vector Maschinen
  - Evaluation von Klassifikatoren
  - Overfitting & Pruning
  - Kombinierte Klassifikatoren
- Recommender Systems
- Clusteringverfahren
- Stream Mining
- Social Network Analysis
- Technische Lösungen
- Datenschutz und gesellschaftliche Aspekte

Bayes Klassifikation I

- Idee
  - Ereignis X: Fall x tritt ein
  - Hypothese H_i: Fall x gehört zu Klasse i

	Einkommen	Ausgaben	Haus	Alter	Klasse: Kreditausfall
Lern-daten	10.000 €	15.000 €	1	17	nein
	20.000 €	10.000 €	1	20	nein
	50.000 €	100.000 €	0	55	ja
Vorhersage	10.000 €	15.000 €	0	55	?

X

gesucht: Wahrscheinlichkeit für ja bzw. nein gegeben X

- Formal
  - Maximiere $p(H_\mathrm{i} \mid X)$ für $i \in \{1 \ldots n\}$ mit n ist Anzahl Klassen
    $p(H_\mathrm{i} \mid X)$ ist bedingte Wahrscheinlichkeit, dass H_i stimmt, gegeben X
  - $p(H_\mathrm{i} \mid X)$ unbekannt
  - Anwendung des Satz von Bayes

$$p(H_\mathrm{i} \mid X) = \frac{p(X|H_\mathrm{i}) \cdot p(H_\mathrm{i})}{p(X)}$$

Bayes Klassifikation II

- Benötigte Wahrscheinlichkeiten
  - $p(X)$: Wahrscheinlichkeit, dass Fall x eintritt
  - $p(H_\mathrm{i})$: Wahrscheinlichkeit, dass Klasse i eintritt
  - $p(X\,|H_\mathrm{i})$: Wahrscheinlichkeit, dass Fall x eintritt, wenn H_i eintrat

- Vorgehen
  „Ablesen“ aller benötigten Wahrscheinlichkeiten durch Vereinfachungen

					Klasse
	Einkommen	**Ausgaben**	**Haus**	**Alter**	**Kreditausfall**
Lern-daten	10.000 €	15.000 €	1	17	nein
	20.000 €	10.000 €	1	20	nein
	50.000 €	100.000 €	0	55	ja
Vorhersage	10.000 €	15.000 €	0	55	?

$p(\mathrm{X}|H_\mathrm{i})$ $p(H_\mathrm{i})$

Bayes Klassifikation III

- Vereinfachungen
  - Reduktion der Berechnungen
    Wahrscheinlichkeit, dass Fall x eintritt ($p(X)$) nicht nötig, ...
    ... da Nenner aller Terme
  - Wahrscheinlichkeit, dass Fall Klasse i eintritt ($p(H_\mathrm{i})$)
    - Abschätzung: $p(H_i) = \frac{|Fälle\ in\ Klasse\ i|}{|\text{Alle Fälle im Trainingsset}|}$
  - Wahrscheinlichkeit, dass Fall x eintritt, wenn H_i eintrat ($p(X\ |H_\mathrm{i})$)
    - Unabhängigkeit der Klassen angenommen, ...
      ... dann gilt $p(X\ |\mathrm{H}_i) = \prod_{k=1}^{n} p(x_k\ |H_i)$
    - ... mit $p(x_k\ |\mathrm{H}_i) = \frac{|Fälle\ in\ Klasse\ i\ mit\ Attributwert\ x_k|}{|\text{Alle Fälle der Klasse i im Trainingsset}|}$

- Vorgehen
  - Berechnung des Zählers $p(X|H_\mathrm{i}) \cdot p(H_\mathrm{i})$ [von $p(H_\mathrm{i} \mid X)$] für alle Klassen i
  - Wahl der Klasse i mit maximalem $p(X|H_\mathrm{i}) \cdot p(H_\mathrm{i})$

Naiver Bayes Klassifikator - Beispiel

- Ziel
  - Vorhersage Kauf: ja oder nein
  - Für Fall $x = (ja, hoch, modern, ?)$
- Bedingte Wahrscheinlichkeiten
  - Klassen für Kauf
    $p(H_{ja}) = \frac{3}{8};\ p(H_{nein}) = \frac{5}{8}$
  - Zulassung
    $p('ja'|H_{ja}) = \frac{3}{3};\ p('ja'|H_{nein}) = \frac{1}{5}$
  - Verbrauch
    $p('hoch'|H_{ja}) = \frac{1}{3};\ p('hoch'|H_{nein}) = \frac{4}{5}$
  - Alter
    $p('modern'|H_{ja}) = \frac{2}{3};\ p('modern'|H_{nein}) = \frac{3}{5}$
- Vorhersage
  - $p(H_{ja}|x) = \frac{p(X|H_{\text{ja}})\cdot p(H_{\text{ja}})}{p(X)} = \frac{\left(\frac{3}{3}\cdot\frac{1}{3}\cdot\frac{2}{3}\right)\cdot\frac{3}{8}}{p(X)} = \frac{\frac{6}{72}}{p(X)}$
  - $p(H_{nein}|x) = \frac{p(X|H_{\text{nein}})\cdot p(H_{\text{nein}})}{p(X)} = \frac{\left(\frac{1}{5}\cdot\frac{4}{5}\cdot\frac{3}{5}\right)\cdot\frac{5}{8}}{p(X)} = \frac{\frac{3}{50}}{p(X)}$

Beispiel

(Zulassung, Verbrauch, Alter | Kauf)

1 (nein, hoch, modern | nein)
2 (nein, gering, modern | nein)
3 (nein, hoch, alt | nein)
4 (nein, hoch, modern | nein)
5 (ja, gering, alt | ja)
6 (ja, gering, modern | ja)
7 (ja, hoch, alt | nein)
8 (ja, hoch, modern | ja)

Ergebnis

- $p(H_{ja}|x) > p(H_{nein}|x)$
- Vorhersage: ja!

Agenda

- Einführung
- Deskriptive Methoden zur Datenexploration
- Datenqualität
- Klassifikation
  - Naiver Bayes Klassifikator
  - Künstliche Neuronale Netze
  - Entscheidungsbäume
  - Support Vector Maschinen
  - Evaluation von Klassifikatoren
  - Overfitting & Pruning
  - Kombinierte Klassifikatoren
- Recommender Systems
- Clusteringverfahren
- Stream Mining
- Social Network Analysis
- Technische Lösungen
- Datenschutz und gesellschaftliche Aspekte

Künstliche Neuronale Netze

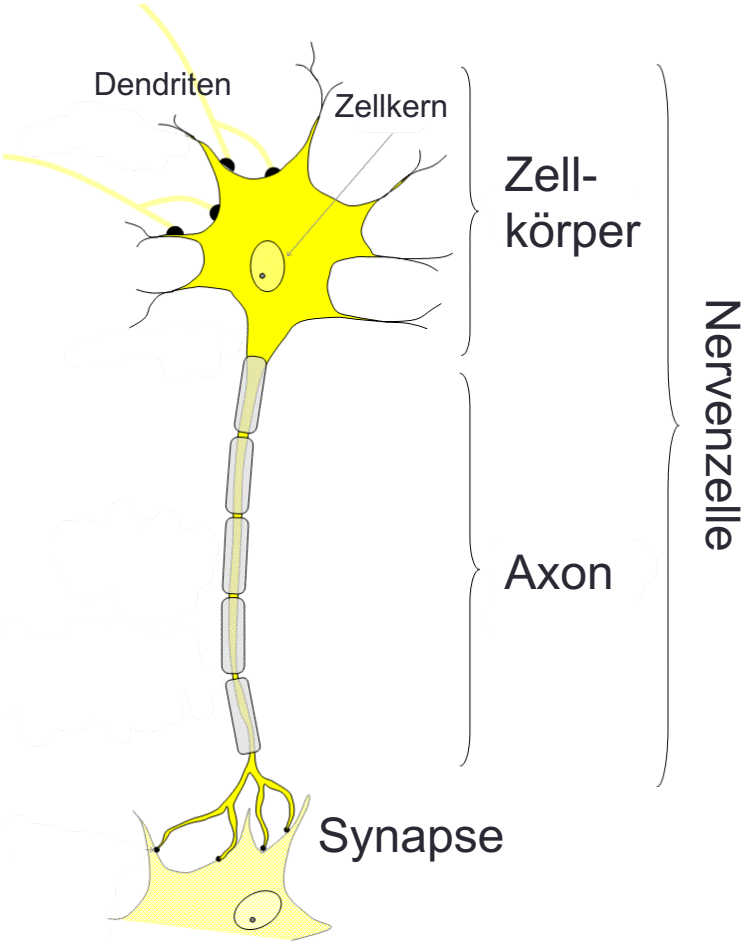
- Nachteile des Naiven Bayes Klassifikators
  - Deterministisches Verfahren
    (Trainingsset muss „alle“ möglichen Fälle repräsentativ abbilden)
  - Kaum Möglichkeiten der Modelloptimierung
    (Details dazu am Ende der Vorlesung)

- Idee
  - Ableiten von Verfahren mit Möglichkeiten der Modellanpassung
  - Ein Ansatz: Künstliche Neuronale Netze (später noch weitere)

- Eigenschaften Künstliche Neuronale Netze
  - Simulation des Informationsverarbeitungsprozesses...
    ... in „biologischen“ Gehirnen
  - Netzwerk einfacher Verarbeitungseinheiten (Nervenzellen)
  - Austausch von Informationen zwischen...
    ... Nervenzellen über Netzwerk
  - Erlauben nicht nur Klassifikation

Biologische Nervenzelle

Illustration



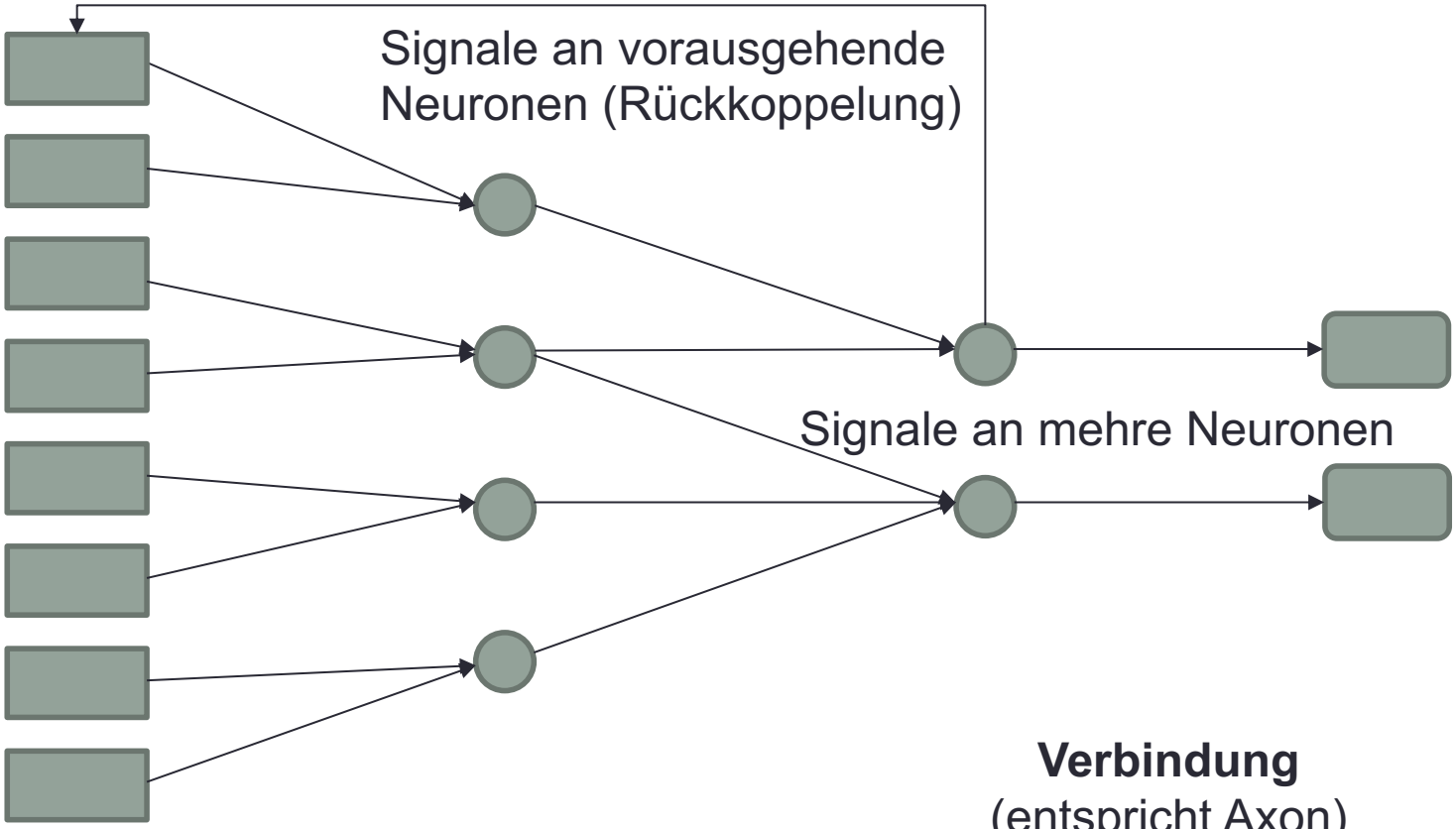
Funktionsweise (stark vereinfacht)

- Innerhalb einer Nervenzelle
  - Dendriten empfangen Signal
  - Jedes Signal ändert „Potential“
  - Erreicht „Potential“ Schwellwert wird Axon ausgelöst
  - D.h. Signal wird über Synapsen an nächste Zelle weitergeleitet
- Im gesamten Gehirn
  - Viele Nervenzellen $(10^{10}/10^{11})$
  - Signale werden von Rezeptoren über Nervenzellen weitergeleitet
  - Nervenzellen erhalten Signale von Nervenzellen $(10^{3}/10^{4})$
  - Intelligenz über Kombination... ... der Nervenzellen

Künstliche Neuronale Netze – Funktionsweise

- Idee: Nervenzelle im menschlichen Gehirn
  - „Verknüpft“ Eingabegröße mit Zielgröße
    Beispiel: Auge sieht Bier, Gehirn meldet Durst
  - Definition
    Binäres Schaltelement mit zwei Zuständen (aktiv, inaktiv)
- Wiederholung Klassifikationsproblem
  - Eingabegrößen: Verschiedene Attribute
  - Zielgröße: Vorhersageklasse geg. Attribute
- Vorgehen Klassifikation
  - Initial
    - Netzwerk aus Nervenzellen
    - Alle Nervenzellen inaktiv, senden keine Signale
  - Eingabegrößen reizt Nervenzelle $\Rightarrow$ Gereizte Nervenzellen senden Signale
  - Signale werden über Netzwerk zum Ausgabeneuron weitergeleitet
  - Ausgabeneuron mit „höchstem Reiz“ definiert Klasse

Künstliches Neuronales Netz – Aufbau I



Eingabeneuron
(entspricht Rezeptoren)

Versteckte Neuronen
(entspricht Zellkörper)

Ausgabeneuron
(entspricht Aktion)

Künstliches Neuronales Netz – Aufbau II

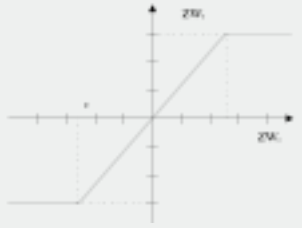
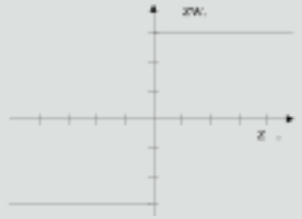
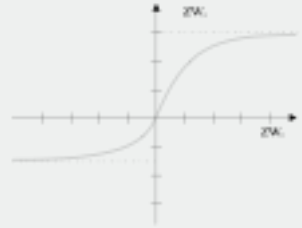
- Drei Schichten
  - Eingabeschicht (bestehend aus Eingabeneuronen)
    - Nimmt Signale aus Attributen auf
    - Leitet (aggregierte) Signale an verborgene Schicht weiter
  - Verborgene Schicht (bestehend aus versteckten Neuronen)
    - Verarbeitet eingehende Signale
    - Leitet Signale an Ausgabeschicht weiter
  - Ausgabeschicht (bestehend aus Ausgabeneuronen)
    - Übernahme der Signale aus verborgener Schicht
    - Ausgabe der Signale

- Weiter Eigenschaften
  - Jede Schicht kann beliebig viele Neuronen enthalten
  - Konvergenz:
    Jedes Neuron kann Signale von mehreren Neuronen empfangen
  - Divergenz
    Jedes Neuron kann Signale an mehrere Neuronen senden

Künstliche Neuronale Netze – Formal

- Inputsignale am Neuron: e_j mit $j = 1 \ldots n$
  - Bilden Attribute der Beobachtungen ab
  - Können kategorisch oder kontinuierlich sein
- Propagierungsfunktion
  - Inputsignale e_j werden zu einer Zahl ϵ zusammengefasst
  - Gewichtung sämtlicher Inputsignale e_j mit spezifischem Gewicht w_j
  - Typische Eingangsfunktion: $\epsilon = \sum_{j=1}^{n} w_j \cdot e_j$
- Aktivierungsfunktion
  - Propagierungswert ϵ löst Aktivität c des Neurons aus
  - Meist lineare Aktivierungsfunktion mit Steigung χ: $c = \chi \cdot \epsilon$
  - Auch denkbar: Diskontierte Berücksichtigung früherer Signale
- Ausgangsfunktion
  - Berechnung des Ausgangswerts a auf Basis der Aktivität c
  - Abbildung ist monoton steigend
  - Wird oft mit Aktivierungsfunktion zu „Transferfunktion“ kombiniert

Künstliche neuronale Netze – Aktivierungsfunktionen

Form	Formal	Visualisierung	Sonstiges
linear	$c = \chi \cdot \epsilon + \delta$ mit $\chi > 0; \epsilon, \delta \in \mathbb{R}$		Identitätsfunktion (mit $\chi = 1; \delta = 0$)
linear, begrenzt	$c = \begin{cases} \beta & \epsilon \geq \sigma \\ \alpha & \epsilon \leq \sigma' \\ \chi \cdot \epsilon + \delta & sonst \end{cases}$ mit $\chi > 0; \epsilon, \delta, \alpha, \beta, \sigma, \sigma' \in \mathbb{R}$		Ausgabe erst wenn σ' überschritten; ab σ keine Anpassung der Ausgabe
Treppenfunktion	$c = \begin{cases} \beta & \epsilon \geq \sigma \\ \alpha & sonst \end{cases}$ mit $\epsilon, \alpha, \beta, \sigma \in \mathbb{R}$		Sprungartiger Anstieg der Aktivität bei σ
Sigmoidfunktion	$c = \frac{1}{1 + e^{-\delta \cdot \epsilon}}$ mit $\delta > 0; \epsilon \in \mathbb{R}$		Langsames Konvergieren gegen untere/obere Grenze

Künstliche Neuronale Netze – Lernen

- Allgemein
  Trainingsdaten werden genutzt um Propagierungs-, Aktivierungs- und Ausgangsfunktion (mit Fokus auf Gewichte!) zu parametrisieren

- Algorithmus
  - Initialisierung aller Parameter mit zufälligen Werten
  - Vergleich Ausgangsvektor mit Sollausgabe (gemäß Trainingsdatensatz)
  - Anpassung der Verbindungsgewichte, ...
    ... so dass Ausgangsvektor sich Sollausgabe annähert
  - Verschiedene Lernregeln möglich
    - Verstärkungslernen
      Gewichte werden angepasst wenn Sollausgabe != Ausgangsvektor
    - Korrigierendes Lernen
      Wie Verstärkungslernen + Berücksichtigung von Fehlern

Künstliche Neuronale Netze – Parametrisierung

- Parameterraum
  - Rückkoppelung kann erlaubt / verboten werden
  - Anzahl der Input- / Outputneutronen ist frei wählbar
  - Anzahl der verborgenen Schichten ist frei wählbar
  - Anzahl der Knoten pro verborgener Schicht ist frei wählbar
  - Aktivierungsfunktion ist frei wählbar
  - Lernrate ist frei wählbar

- Implikationen
  - Mehr „Freiheit“ als bei Naivem Bayes Klassifikator
  - Parametrisierung hat oft großen Einfluss auf Güte des Modells
  - Es gibt keine „beste“ Parametrisierung
    - Erfahrungswissen notwendig
    - Ausprobieren notwendig
  - Anwendungsgebiete sind vielseitig

Künstliche Neuronale Netze – Bewertung

- Vorteile
  - Gutes Verhalten bei neuen und verrauschten Daten
  - Kann auch auf Regressionsprobleme angewendet werden
  - Finden auch bei fehlerbehafteten Daten gute Vorhersagen
  - Ähnliche Inputs erzeugen ähnliche Outputs
    Dadurch auch Anwendung auf unbekannten Inputdaten möglich

- Nachteile
  - Lernen oft vergleichsweise aufwändig
  - Ergebnis schwer zu interpretieren

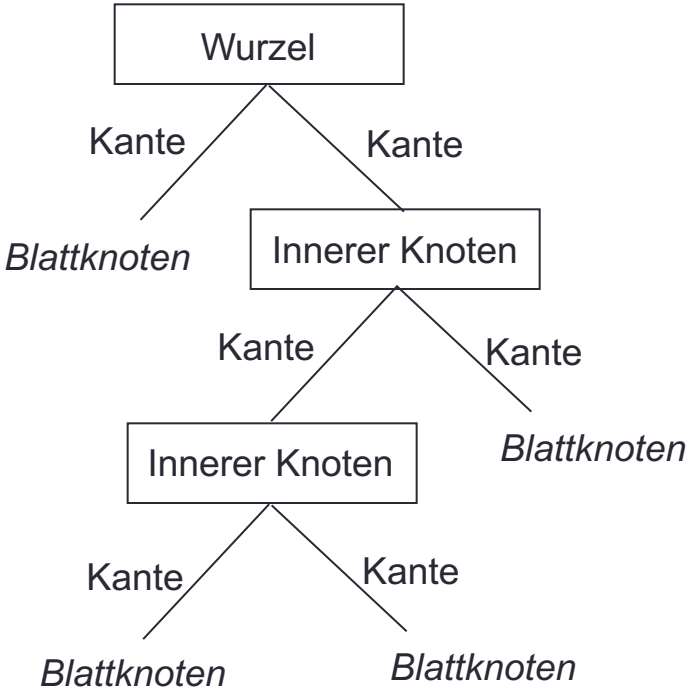
Agenda

- Einführung
- Deskriptive Methoden zur Datenexploration
- Datenqualität
- Klassifikation
  - Naiver Bayes Klassifikator
  - Künstliche Neuronale Netze
  - Entscheidungsbäume
  - Support Vector Maschinen
  - Evaluation von Klassifikatoren
  - Overfitting & Pruning
  - Kombinierte Klassifikatoren
- Recommender Systems
- Clusteringverfahren
- Stream Mining
- Social Network Analysis
- Technische Lösungen
- Datenschutz und gesellschaftliche Aspekte

Entscheidungsbäume

- Klassifikation mit Hilfe von Entscheidungsbäumen
  - „Baum mit: Wurzel, Blattknoten, innere Knoten und Kanten
  - Blattknoten sind Klasse (ggfs. mehrere Blattknoten pro Klasse)
  - Innere Knoten: Attribut (ggfs. mehrere innere Knoten pro Attribut)
  - Kanten: Attributwerte

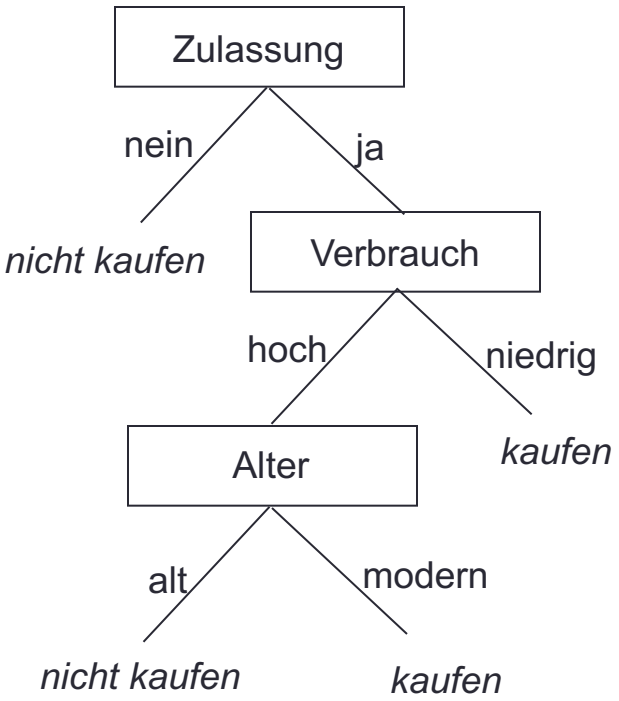
Baum:



Entscheidungsbaum – Beispiel

- Entscheidungsbaum zur Vorhersage „Kauf eines Autos“

Repräsentation als Baum



Repräsentation als Regeln

Wenn
Zulassung nicht vorhanden
dann nicht kaufen.

Wenn
Zulassung vorhanden und
Verbrauch gering
dann kaufen.

Wenn
Zulassung vorhanden und
Verbrauch hoch und
Alter alt
dann nicht kaufen.

Wenn
Zulassung vorhanden und
Verbrauch hoch und
Alter modern
dann kaufen.

Konstruktion

- Konstruktion von Entscheidungsbäumen
  - Vorgehen zur Bestimmung von Entscheidungsbäumen (Top-Down)
  - Ziele
    - Baum der mit allen verfügbaren Daten „möglichst gut“ zusammenpasst
    - Blattknoten sind homogen zur vorhergesagten Klasse

- Vorgehen
  - Alle Trainingsdaten liegen im Wurzelknoten
  - Bestimmung zusätzlicher innerer Knoten
    - Auswahl des Attributs das alle Daten im Knoten...
      ... „möglichst gut“ bzgl. Klasse separiert
    - Aufteilung der Daten im Kinderknoten
    - Ist „Stoppkriterium“ erreicht:
      - Zuordnung aller Fälle zur Majoritätsklasse (Blattknoten!)
    - Sonst
      - Rekursives Bestimmen weiterer innerer Knoten

Entscheidungsbaum – Beispiel

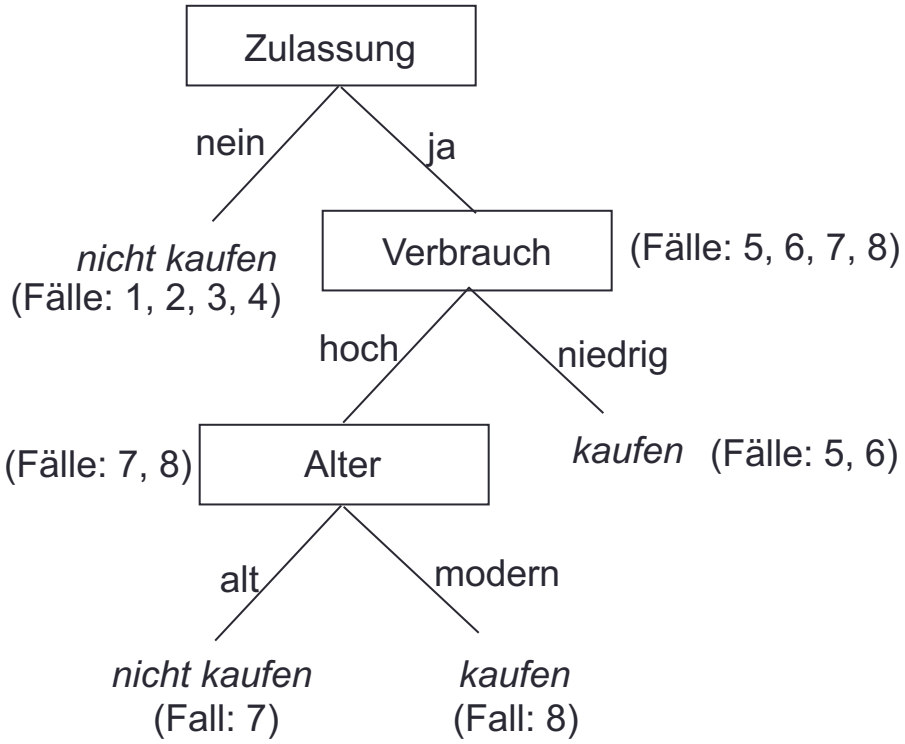
- Entscheidungsbaum zur Vorhersage „Kauf eines Autos“

Trainingsdaten
(Zulassung, Verbrauch, Alter, Kauf)

~~1 (nein, hoch, modern, nein)~~
~~2 (nein, gering, modern, nein)~~
~~3 (nein, hoch, alt, nein)~~
~~4 (nein, hoch, modern, nein)~~
~~5 (ja, gering, alt, ja)~~
~~6 (ja, gering, modern, ja)~~
~~7 (ja, hoch, alt, nein)~~
~~8 (ja, hoch, modern, ja)~~

- Algorithmus (nur für Beispiel tauglich!)
  - Auswahlmaß
    Sagt eine Klasse perfekt voraus
  - Stoppkriterium
    Separiert Klassen perfekt

Repräsentation als Baum



Vorgehen bei der Konstruktion

- Existierende Algorithmen unterscheiden sich in
  - … der Wahl des Stoppkriteriums
  - … der Art des Splits (einer, mehrere, ...)
  - … dem Vorgehen zur Wahl des Splitattributs („Auswahlmaß“)

- Diverse Verfahren
  - CHAID
  - C4.5/C5.0
  - CART
  - etc.

  Unterschiede:
  - Anzahl der Blattknoten
  - Bildung der Knoten
  - Vermeidung von Overfitting

- Gemeinsamkeit von Entscheidungsbäumen
  - Lösen Klassifikationsprobleme
  - Ergebnis sind leicht interpretierbar
  - Übersetzbar in Regelsystem

- In der Praxis oft: Random Forest (Kombination von Bäumen)

Stoppkriterien

- Natürliche Stoppkriterien
  - Knoten enthält (fast) nur Tupel einer Klasse
  - Alle Klassifikationsattribute ausgeschöpft

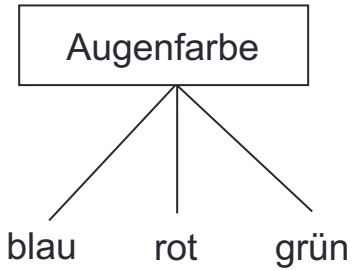
- Weitere Kriterien
  - Minimale Fallzahl je Knoten
    - Weniger als n Fälle im Knoten
    - Weniger als $p\%$ der Fälle im Baum sind im Knoten
  - Minimaler Anteil falsch klassifizierter Tupel
    - Weniger als n falsch klassifizierte Fälle im Knoten
    - Weniger als $p\%$ der Fälle im Knoten falsch klassifiziert
  - Maximale Baumtiefe
    - Anzahl der Kanten zwischen Wurzel und Blattknoten größer n
    - Anzahl der Kanten zwischen bis Blattknoten zu unterschiedlich
  - Maximale Knotenanzahl
    - Anzahl der Knoten im Baum übersteigt n

Art des Splits

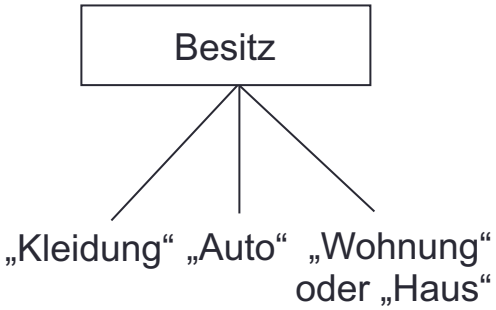
- Diskrete vs. kontinuierliche Attribute
  - Diskret
    Ein Knoten pro Attributwert
  - Ordinal
    Separierung in Klassen
  - Kontinuierlich
    Ein Knoten pro Attributintervall

- Binäre vs. n-äre Bäume
  - Zwei oder mehr Ausgangskanten

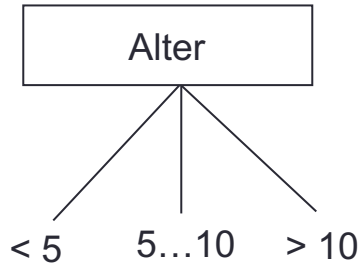
Diskret



Ordinal



Kontinuierlich



Auswahlmaße

- Ziel
  Wahl des Attributs, ...
  ... Das verbleibende Fälle am besten separiert

- Im Beispiel: verschiedene binäre Splits
  - Zulassung
    - Ja: (ja, ja, nein, ja)
    - Nein: (nein, nein, nein, nein)
  - Verbrauch
    - Hoch: (nein, nein, nein, nein, ja)
    - Gering: (nein, ja, ja)
  - Alter
    - Modern: (nein, nein, nein, ja, ja)
    - Alt: (nein, ja, nein)

- Algorithmus vergleicht alle möglichen Splits...
  ... und Auswahlmaß wählt dann „bestes“ Splitattribut

Beispiel

(Zulassung, Verbrauch, Alter | Kauf)

1 (nein, hoch, modern | nein)
2 (nein, gering, modern | nein)
3 (nein, hoch, alt | nein)
4 (nein, hoch, modern | nein)
5 (ja, gering, alt | ja)
6 (ja, gering, modern | ja)
7 (ja, hoch, alt | nein)
8 (ja, hoch, modern | ja)

Gini Index

- Idee
  Minimierung der „Heterogenität“ innerhalb der neuen Knoten
- Mathematisch
  Wahrscheinlichkeit, dass wiederholtes Ziehen mit Zurücklegen...
  ... von Fällen im Knoten zu unterschiedlichen Klassen führt
- Formal
  - Allgemein: $1 - \sum_{i=1}^{n} p_i^2$ mit n ist Anzahl der Klassen
  - 2-Klassen: $1 - p_0^2 - p_1^2$
- Wertebereich:
  - $\left[0 \ \ldots 1 - \frac{1}{k}\left[= 1 - k\left(\frac{1}{k}\right)^2\right]\right]$
- Interpretation
  - Gini Index = 0: Perfekte Klassifikation
  - Gini Index = $1 - \frac{1}{k}$: Alle Klassen gleich häufig vertreten

Gini Index - Beispiel

- Gesamtdaten
  - Gini-Index: $1 - p_0^2 - p_1^2 = 1 - \left(\frac{5}{8}\right)^2 - \left(\frac{3}{8}\right)^2 = 0.47$
- Split Zulassung
  - Nein: (n, n, n, n) $\Rightarrow 1 - \left(\frac{4}{4}\right)^2 - \left(\frac{0}{4}\right)^2 = 0.00$
  - Ja: (j, j, n, j) $\Rightarrow 1 - \left(\frac{1}{4}\right)^2 - \left(\frac{3}{4}\right)^2 = 0.38$
  - Gewichtetes Mittel: $\frac{4}{8} \cdot 0.00 + \frac{4}{8} \cdot 0.38 = 0.19$
- Split Verbrauch
  - Hoch: (n, n, n, n, j) $\Rightarrow 1 - \left(\frac{4}{5}\right)^2 - \left(\frac{1}{5}\right)^2 = 0.32$
  - Gering: (n, j, j) $\Rightarrow 1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2 = 0.44$
  - Gewichtetes Mittel: $\frac{5}{8} \cdot 0.32 + \frac{3}{8} \cdot 0.44 = 0.37$
- Beide Splits Verbesserung vgl. mit Gesamtdaten
  Split Zulassung besser

Beispiel

(Zulassung, Verbrauch, Alter | Kauf)

1 (nein, hoch, modern | nein)
2 (nein, gering, modern | nein)
3 (nein, hoch, alt | nein)
4 (nein, hoch, modern | nein)
5 (ja, gering, alt | ja)
6 (ja, gering, modern | ja)
7 (ja, hoch, alt | nein)
8 (ja, hoch, modern | ja)

Informationsgewinn

- Idee
  - Bewertung der Reduktion der „Unordnung“ durch Split
  - „Shannon Entropie“ als Maß für „Ordnung“ eines Systems

- Ursprung Informationstheorie
  - Durch „Kodierung“ von Daten mit mehr Parametern: mehr Information

- Formal
  - $I = \text{Entropie}(\text{Gesamtdaten}) - \text{Entropie}(\text{Knoten nach Split})$
    mit Entropie: $-\sum_{i=1}^{n}(p_i \cdot \log_2 p_i)$ und n ist Anzahl der Klassen

- Interpretation
  - Je größer der Informationsgewinn...
    ... umso besser das Splitattribut

Informationsgewinn - Beispiel

- Entropie Gesamtdaten
  - $-\sum_{i=1}^{|C|}(p_i \cdot \log_2 p_i) = -\left(\frac{5}{8}\cdot \log_2 \frac{5}{8} + \frac{3}{8}\cdot \log_2 \frac{3}{8}\right) = 0.95$
- Split Zulassung
  - Ja: (j, j, n, j) $\Rightarrow -\left(\frac{1}{4}\cdot \log_2 \frac{1}{4} + \frac{3}{4}\cdot \log_2 \frac{3}{4}\right) = 0.81$
  - Nein: (n, n, n, n) $\Rightarrow -\left(\frac{4}{4}\cdot \log_2 \frac{4}{4} + \frac{0}{4}\cdot \log_2 \frac{0}{4}\right) = 0.00$
  - Gewichtetes Mittel: $0.41 \Rightarrow$ Informationsgew.: 0.54
- Split Verbrauch
  - Hoch: (n, n, n, n, j) $\Rightarrow -\left(\frac{4}{5}\cdot \log_2 \frac{4}{5} + \frac{1}{5}\cdot \log_2 \frac{1}{5}\right) = 0.72$
  - Gering: (n, j, j) $\Rightarrow -\left(\frac{1}{3}\cdot \log_2 \frac{1}{3} + \frac{2}{3}\cdot \log_2 \frac{2}{3}\right) = 0.92$
  - Gewichtetes Mittel: $\frac{5}{8}\cdot 0.72 + \frac{3}{8}\cdot 0.92 = 0.80 \Rightarrow$ Informationsgew.: 0.15
- Beide Splits Verbesserung vgl. mit Gesamtdaten
  Split Zulassung besser

Beispiel

(Zulassung, Verbrauch, Alter |Kauf)

1 (nein, hoch, modern | nein)
2 (nein, gering, modern | nein)
3 (nein, hoch, alt | nein)
4 (nein, hoch, modern | nein)
5 (ja, gering, alt | ja)
6 (ja, gering, modern | ja)
7 (ja, hoch, alt | nein)
8 (ja, hoch, modern | ja)

χ^2 – Maß

- Idee
  - Werte lassen sich besonders gut in Klassen unterteilen, wenn abhängig
  - Zwei Verteilungen sind unabhängig, wenn $p(x_i \cap y_j) = p(x_i) \cdot p(y_j)$
  - Prüfung typischerweise mit χ^2-Test

- Mathematisch
  Hypothesentest selbst wird nicht ausgeführt, ...
  ... sondern nur die Teststatistik ermittelt und für Splits verglichen

- Formal
  - Prüfung über χ^2-Teststatistik: $\chi^2 = \sum_{j=1}^{k} \frac{(n_j - \overline{n_j})^2}{\overline{n_j}}$

    mit k ist Zahl der (Klasse, Attributwert)-Kombinationen
    und n_j ($\overline{n_j}$) Zahl der Beob. (erwartet) einer (Klasse, Attributwert)-Kombi.

- Interpretation
  - Je größer die χ^2-Teststatistik umso zuverlässiger kann davon... ausgegangen werden, dass Größen nicht unabhängig

χ^2 – Maß - Beispiel

- Absolute Häufigkeiten

Zulassung		Kauf	
		ja	nein	
	ja	3	1	4
	nein	0	4	4
		3	5	8

Verbrauch		Kauf	
		ja	nein	
	hoch	1	4	5
	gering	2	1	3
		3	5	8

Beispiel

(Zulassung, Verbrauch, Alter |Kauf)

1 (nein, hoch, modern | nein)
2 (nein, gering, modern | nein)
3 (nein, hoch, alt | nein)
4 (nein, hoch, modern | nein)
5 (ja, gering, alt | ja)
6 (ja, gering, modern | ja)
7 (ja, hoch, alt | nein)
8 (ja, hoch, modern | ja)

- Split Zulassung
  - $\chi^2 = \sum_{j=1}^{n} \frac{(n_j - \overline{n_j})^2}{\overline{n_j}} = \frac{\left(3-\frac{4\cdot 3}{8}\right)^2}{\frac{4\cdot 3}{8}} + \frac{\left(1-\frac{4\cdot 5}{8}\right)^2}{\frac{4\cdot 5}{8}} + \frac{\left(0-\frac{4\cdot 3}{8}\right)^2}{\frac{4\cdot 3}{8}} + \frac{\left(4-\frac{4\cdot 5}{8}\right)^2}{\frac{4\cdot 5}{8}} = 4.80$
- Split Verbrauch
  - $\chi^2 = \sum_{j=1}^{n} \frac{(n_j - \overline{n_j})^2}{\overline{n_j}} = \frac{\left(1-\frac{5\cdot 3}{8}\right)^2}{\frac{5\cdot 3}{8}} + \frac{\left(4-\frac{5\cdot 5}{8}\right)^2}{\frac{5\cdot 5}{8}} + \frac{\left(2-\frac{3\cdot 3}{8}\right)^2}{\frac{3\cdot 3}{8}} + \frac{\left(1-\frac{3\cdot 5}{8}\right)^2}{\frac{3\cdot 5}{8}} = 1.74$
- Split Zulassung besser

Auswahlmaße – Überblick

- Hier vorgestellt
  - Gini-Index
    - Fokus auf Homogenität
    - Findet früh im Baum reine Klassen
  - Informationsgewinn
    - Fokus auf Reduktion der Heterogenität
    - Erzeugt „balancierte“ Bäume
  - χ^2 – Maß
    - Fokus auf stochastische Unabhängigkeit
    - Mathematisch ableitbar

- Weitere Ansätze
  - Minimale Beschreibungslänge („minimum description length“)
    - Ähnlich Informationsgewinn
    - Zusätzlich „Strafe“ für zunehmende Komplexität des Baums
  - ...

Algorithmus CHAID (Chi Squared Automatic Interaction Detection)

- Eigenschaften
  - Berücksichtigt ausschließlich kategorische Attribute
  - Erzeugt binäre Bäume

- Rekursives Anwenden zweier Schritte
  - Schritt 1: Zusammenfassung von Kategorien
    - Identifikation der Attribute mit mehr als zwei Attributwerten
    - Ermittlung aller möglichen Attributwertkombinationen
    - Zusammenfassung aller Attributwertekombinationen, für die χ^2 Unabhängigkeitstest signifikanten Zusammenhang findet
  - Schritt 2: Wahl des Splits
    - Ermittlung der χ^2 Teststatistik für alle Attribute
    - Split anhand des Attributs mit höchster Teststatistik
    - Auflösung der Attributwertkombinationen

- Stoppkriterium
  - Abbruch, wenn keine weiteren Splits möglich

Algorithmus CART (Classification and Regression Trees)

- Eigenschaften
  - Berücksichtigt kategorische und metrische Attribute
  - Erzeugt binäre Bäume

- Rekursives Anwenden eines Schritts
  - Einsatz des Gini-Index als Auswahlmaß
  - Zuordnung von Beobachtungen zu Kindknoten
    - Metrische Attribute: Kindknoten 1 bei $Fall \leq X_j$, sonst Kindknoten 2
    - Kategorische Attribute: Kindknoten 1 bei Fall $\in B_j$, sonst Kindeknoten 2 (dabei ist B_j eine Teilmenge der Attributwerte)

- Stoppkriterium
  - Abbruch, wenn keine weiteren Splits möglich

- Hinweis
  CART, CHAID und C4.5/ID3 sind nicht-parametrische Verfahren, ...
  ... d.h. keine a priori-Annahmen über Verteilung der Fehler

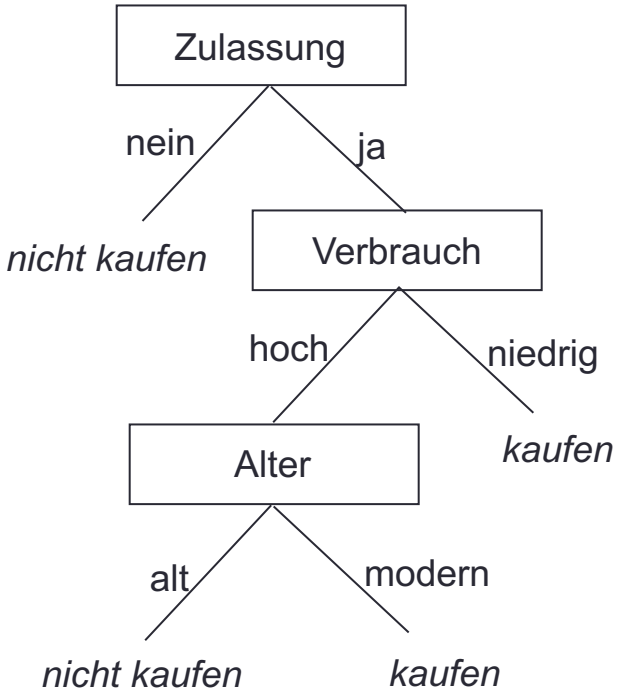
Regelbasierte Klassifikatoren

- Vorgehen
  - Lernen eines Entscheidungsbaums
  - Überführen des Entscheidungsbaums in Regeln
  - Einführung von „Default“-Regeln für Regeln...
    ... mit wenigen Daten

- Beispiel:
  - Entscheidungsbaum rechts wird zu

    Default Regel
    Auto nicht kaufen

    Wenn
    Zulassung vorhanden und
    Verbrauch gering
    dann kaufen.



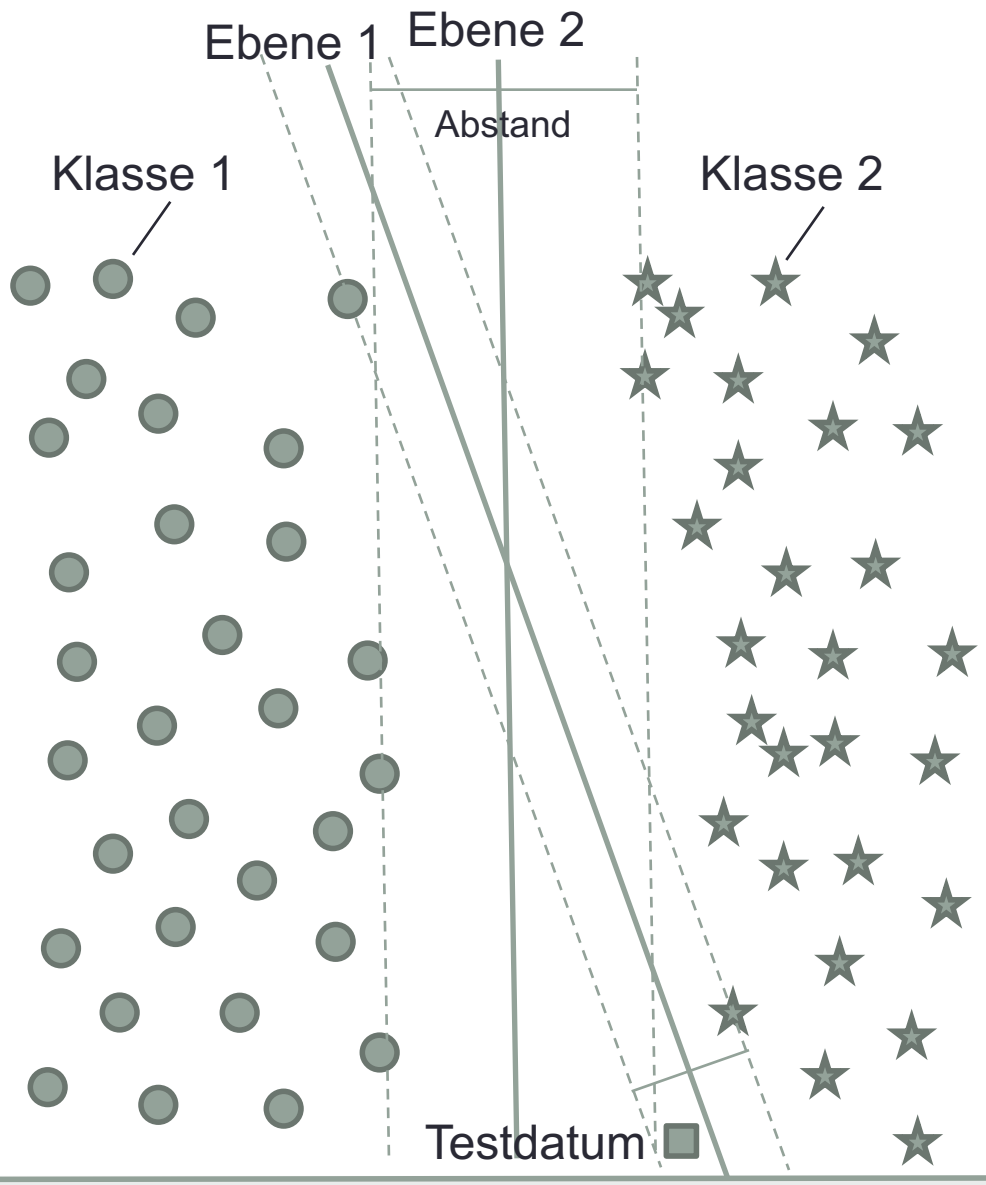
- Vorteile
  - Stärkere Generalisierung
    (da weniger Regeln als Entscheidungsbäume)
  - Leicht verständlich

Agenda

- Einführung
- Deskriptive Methoden zur Datenexploration
- Datenqualität
- Klassifikation
  - Naiver Bayes Klassifikator
  - Künstliche Neuronale Netze
  - Entscheidungsbäume
  - Support Vector Maschinen
  - Evaluation von Klassifikatoren
  - Overfitting & Pruning
  - Kombinierte Klassifikatoren
- Recommender Systems
- Clusteringverfahren
- Stream Mining
- Social Network Analysis
- Technische Lösungen
- Datenschutz und gesellschaftliche Aspekte

Linear separierbare Daten

- Idee
  - Wahl einer Ebene, ...
    ... die Klassen trennt
  - Berechnung des Abstands...
    ... Zwischen Ebene und nächstem Punkt jeder Seite (Support Vector)
  - Ebene mit höchstem Abstand...
    ... trennt Klassen bestmöglich
- Prüfung der Idee
  - Betrachtung Testdatum
  - Ebene 1: Klasse 1
  - Ebene 2: Klasse 2
  - Zuordnung zu Klasse 2...
    ... sinnvoller!



Finden der Ebene – Formal I

- Trainingsdaten:
  $(\bar{X}_i, y_i)$ mit $\bar{X}_i$ ist Vektor der Attributwerte und $y_i = \{-1, +1\}$ ist Klasse

- Ziel: Allgemeine Form der Ebene: $\bar{W} \cdot \bar{X} + b = 0$

- Vorgehen (Idee)
  - Ableiten von $\bar{W}$ (Richtung der Ebene) und...
  - Ableiten von b (Abstand zum Ursprung)...
    ... durch „Einsetzen“ der Trainingsdaten

Finden der Ebene – Vorgehen

- Datenpunkte der Klassen sollen auf gegenüberliegenden Seiten liegen:
  - $\overline{W} \cdot \overline{X_i} + b \geq 0 \; \forall i: y_i = +1$
  - $\overline{W} \cdot \overline{X_i} + b \leq 0 \; \forall i: y_i = -1$
- Da des Vorgehens ist den Abstand zwischen Ebene und Datenpunkten zu maximieren, Einführung des Parameters c
  - $\overline{W} \cdot \overline{X_i} + b \geq +c \; \forall i: y_i = +1$
  - $\overline{W} \cdot \overline{X_i} + b \leq -c \; \forall i: y_i = -1$
- Koordinatensystem lässt sich immer so Skalieren, dass gilt $c = 1$
  - $\overline{W} \cdot \overline{X_i} + b \geq +1 \; \forall i: y_i = +1$
  - $\overline{W} \cdot \overline{X_i} + b \leq -1 \; \forall i: y_i = -1$
- Überführung beider Bedingungen in eine Gleichung
  - $y_i(\overline{W} \cdot \overline{X_i} + b) \geq +1 \; \forall i$
- Jetzt:
  - Maximierung Abstand zwischen parallelen Ebenen (Support Vectoren)...
    ... unter dieser Nebenbedingung ($y_i(\overline{W} \cdot \overline{X_i} + b) \geq +1 \; \forall i$)

SVM - Bewertung

- Herausforderungen
  - Standard Algorithmus funktioniert nur für binäre Klassifikationsprobleme
  - Anwendung auf allgemeine Klassifikationsprobleme:
    Lernen mehrerer SVM und Zusammenführung der Ergebnisse

- Vorteile
  - Oft hervorragende Ergebnisse
  - Kann auch auf Regressionsprobleme angewendet werden

- Nachteile
  - Skaliert schlecht für große Lerndatensätze
  - Ergebnis oft schwer zu interpretieren

- Häufige Anwendungen:
  - Handschrifterkennung, Objekterkennung, Zuordnung Sprache zu Person

Agenda

- Einführung
- Deskriptive Methoden zur Datenexploration
- Datenqualität
- Klassifikation
  - Naiver Bayes Klassifikator
  - Künstliche Neuronale Netze
  - Entscheidungsbäume
  - Support Vector Maschinen
  - Evaluation von Klassifikatoren
  - Overfitting & Pruning
  - Kombinierte Klassifikatoren
- Recommender Systems
- Clusteringverfahren
- Stream Mining
- Social Network Analysis
- Technische Lösungen
- Datenschutz und gesellschaftliche Aspekte

Evaluation von Klassifikatoren

- Konfusionsmatrix

		Vorhersage	
		Ja	Nein
Tatsächliche Klasse	Ja	True Positives (TP)	False Negatives (FN)
	Nein	False Positives (FP)	True Negatives (TN)

- Akkuratheit = $\frac{(TP + TN)}{(TP + FN + FP + TN)}$
- Sensitivität = $\frac{TP}{(TP + FN)}$
- Spezitivität = $\frac{TN}{(FP + TN)}$
- Präzision = $\frac{TP}{(TP + FP)}$
- Lift = $\frac{Präzision}{P(Tats.ja)}$ mit $P(Tats.ja) = \frac{(TP + FN)}{(TP + FN + FP + TN)}$

Beispiel 1 (annähernd perfekter Klassifikator)

- Konfusionsmatrix

		Vorhersage		
		Ja	Nein	
Tatsächliche Klasse	Ja	490 (TP)	10	(FN)
	Nein	10 (FP)	490	(TN)

- Akkuratheit = $\frac{(TP + TN)}{(TP + FN + FP + TN)} = 0.98$
- Sensitivität = $\frac{TP}{(TP + FN)} = 0.98$
- Spezitivität = $\frac{TN}{(FP + TN)} = 0.98$
- Präzision = $\frac{TP}{(TP + FP)} = 0.98$
- Lift = $\frac{Präzision}{P(Tats.ja)} = 1.96$ mit $P(Tats.ja) = \frac{(TP + FN)}{(TP + FN + FP + TN)} = 0.50$

Beispiel 2 (ungünstiger Klassifikator)

- Konfusionsmatrix

		Vorhersage		
		Ja	Nein	
Tatsächliche Klasse	Ja	10 (TP)	90 (FN)	
	Nein	95 (FP)	805 (TN)	

- Akkuratheit = $\frac{(TP + TN)}{(TP + FN + FP + TN)} = 0.82$
- Sensitivität = $\frac{TP}{(TP + FN)} = 0.10$
- Spezitivität = $\frac{TN}{(FP + TN)} = 0.89$
- Präzision = $\frac{TP}{(TP + FP)} = 0.10$
- Lift = $\frac{Präzision}{P(Tats.ja)} = 1.00$ mit $P(Tats.ja) = \frac{(TP+FN)}{(TP + FN + FP + TN)} = 0.10$

Beispiel 2a (kein Klassifikator)

- Konfusionsmatrix

		Vorhersage		
		Ja	Nein	
Tatsächliche Klasse	Ja	0 (TP)	100 (FN)	
	Nein	0 (FP)	900 (TN)	

- Akkuratheit = $\frac{(TP + TN)}{(TP + FN + FP + TN)} = 0.90$ (besser!)
- Sensitivität = $\frac{TP}{(TP + FN)} = 0.00$ (schlechter)
- Spezitivität = $\frac{TN}{(FP + TN)} = 1.00$ (besser!)
- Präzision = $\frac{TP}{(TP + FP)}$ = undef.
- Lift = $\frac{Präzision}{P(Tats.ja)}$ = undef.

Beispiel 3 (durchschnittlicher Klassifikator)

- Konfusions-Matrix

		Vorhersage			
		Ja		Nein	
Tatsächliche Klasse	Ja	259	(TP)	1,077	(FN)
	Nein	578	(FP)	21,664	(TN)

- Akkuratheit = $\frac{(TP + TN)}{(TP + FN + FP + TN)} = 0.93$
- Sensitivität = $\frac{TP}{(TP + FN)} = 0.19$
- Spezitivität = $\frac{TN}{(FP + TN)} = 0.97$
- Präzision = $\frac{TP}{(TP + FP)} = 0.31$
- Lift = $\frac{Präzision}{P(Tats.ja)} = 5.46$ mit $P(Tats.ja) = \frac{(TP + FN)}{(TP + FN + FP + TN)} = 0.06$

Cross-Validation

- Bestimmung der Evaluationsmaße auf Trainingsdaten nicht sinnvoll...
  ... deshalb meist Trennung in Test- und Trainingsdaten

- Unterteilung der Ausgangsdaten in k Partitionen
  - Typischerweise wird $k = 10$ gewählt
  - Eine Partition bildet Test Set
  - $k-1$ Partitionen bilden Training Set

- Berechnung und Evaluation von k Klassifikatoren:
  - In k Runden wird jedes Datentupel $k - 1$ mal zum Lernen verwendet und genau ein mal klassifiziert.

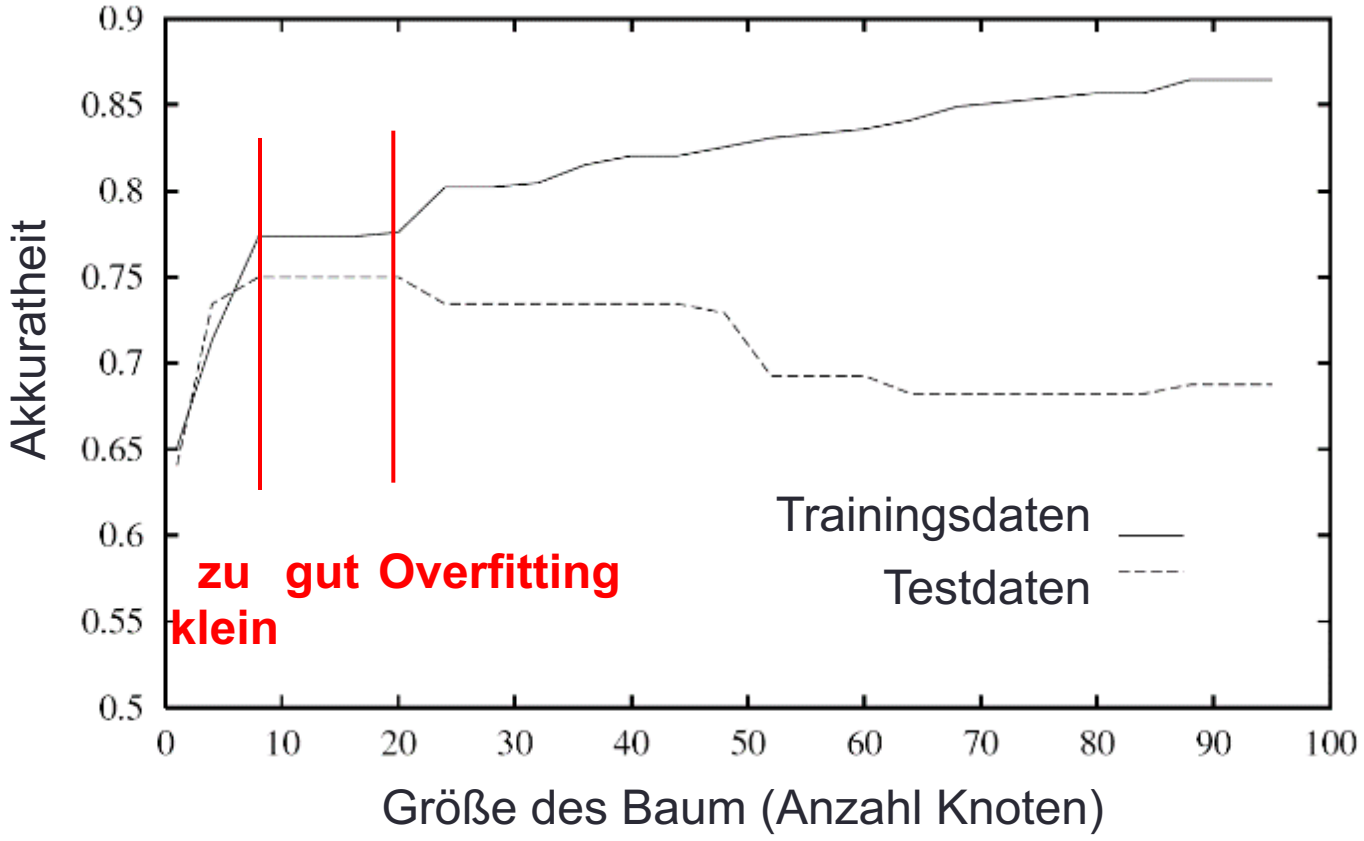
- Stratifizierte Cross-Validation ist in vielen Fällen die zu empfehlende Evaluationstechnik, besonders aber bei kleinen Datensätzen.
  - Achtung: Cross-Validation ist sehr Rechenaufwändig

- „Leave-One-Out“ ist Spezialfall für $k = n$

Agenda

- Einführung
- Deskriptive Methoden zur Datenexploration
- Datenqualität
- Klassifikation
  - Naiver Bayes Klassifikator
  - Künstliche Neuronale Netze
  - Entscheidungsbäume
  - Support Vector Maschinen
  - Evaluation von Klassifikatoren
  - Overfitting & Pruning
  - Kombinierte Klassifikatoren
- Recommender Systems
- Clusteringverfahren
- Stream Mining
- Social Network Analysis
- Technische Lösungen
- Datenschutz und gesellschaftliche Aspekte

Overfitting



- Klassifikator ist optimiert auf Trainingsdaten, die nicht Realität widerspiegeln

Pruningverfahren

- Gründe für Pruning komplexer Bäume
  - Einfachheit / Verständlichkeit
  - Verhinderung von Overfitting / Generalisierungsfähigkeit

- Pre-Pruning: Stopkriterien bei Baumerstellung

- Post-Pruning: Nachträgliches Stutzen
  - Subtree Replacement
    - Ersetzen von Entscheidungsknoten durch Blattknoten...
      ... wenn Klassifikationsbeitrag gering
    - Optimal: Entscheidung zum Ersetzen mit „frischen“ Daten evaluieren
  - Subtree Raising
    - Verschiebung von Teilbäumen nach oben
    - Insbesondere dann, wenn es Attribute gibt, die einzeln wenig, aber in Kombination sehr stark zur Klassifikation beitragen.
    - Solche Attribute rutschen sonst sehr leicht weit nach unten.

Agenda

- Einführung
- Deskriptive Methoden zur Datenexploration
- Datenqualität
- Klassifikation
  - Naiver Bayes Klassifikator
  - Künstliche Neuronale Netze
  - Entscheidungsbäume
  - Support Vector Maschinen
  - Evaluation von Klassifikatoren
  - Overfitting & Pruning
  - Kombinierte Klassifikatoren
- Recommender Systems
- Clusteringverfahren
- Stream Mining
- Social Network Analysis
- Technische Lösungen
- Datenschutz und gesellschaftliche Aspekte

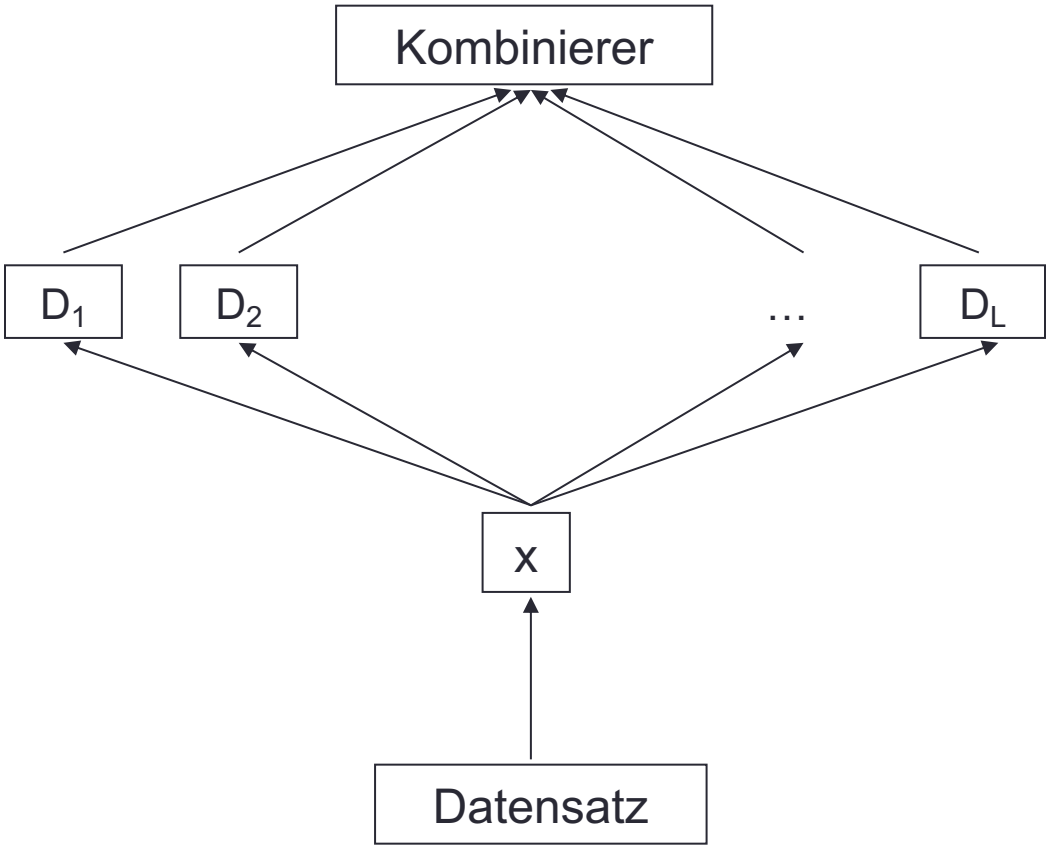
Kombinierte Klassifikatoren - Motivation

- Im „banalen Leben“
  - Bei wichtiger Entscheidung
    - Konsultation mehrer Experten
    - Beispiel: Ärzte vor kritischer OP, Freunde vor Pferdewette
  - Entscheidungsfindung
    - Mehrheit der Experten oder
    - Vertrauenswürdigste Experten

- Bei Klassifikationsproblemen
  - Bei wichtiger Entscheidung
    - Mehrere Klassifikatoren
  - Entscheidungsfindung
    - Kombination der Klassifikatoren oder
    - Ermittlung mehrerer Klassifikatoren und Wahl des „besten“

- Ziel: Erhöhung der Akkuratheit

Kombinierte Klassifikatoren - Ansatzpunkte



Kombinations-Ebene:
Einsatz verschiedener Kombinationstechniken

Klassifikator-Ebene:
Einsatz verschiedener Klassifikatoren

Feature-Ebene:
Einsatz verschiedener Feature-Mengen

Daten-Ebene:
Einsatz verschiedener Teilmengen

Daten-Ebene: Bagging & Boosting

- Ursprünglicher Datensatz D, $d = |D|$

- Bagging
  - Zufällige Auswahl von k Lerndatensätzen
    - Vorgehen: Ziehen mit Zurücklegen von d Tupeln
  - Lernen je eines Klassifikators pro Lerndatensatz
  - Resultierende k Klassifikatoren oft erstaunlich unterschiedlich

- Boosting
  - Ähnlich Bagging
  - Ausnahme $(i+1)$ter Klassifikator:
    Fokus auf falsch klassifizierte Tupel in (i)tem Klassifikator

- Optionaler Schritt
  - Evaluation aller k Klassifikatoren
  - Ergebnisse gewichtet (z.B. mit Akkuratheit)

Feature-Ebene: Feature Selection

- Problem: „Curse of Dimensionality“
  - Lernen sehr aufwändig
  - Viele Attribute irrelevant
- Optimal:
  - Domänen-Experte identifiziert relevante Features (Attribute)
- Alternativ: Feature Selection
  - Meist Entropie-basierte Algorithmen
  - (Kombination verschiedener Selektionsstrategien denkbar)
- Bei kombinierten Klassifikatoren
  - Verschiedene Klassifikatoren durch verschiedene Attribut-Mengen von verschiedenen Feature Selection Strategien

Klassifikator-Ebene

- Alternativen:
  - Einsatz eines Klassifikators mit verschiedenen Parametern (z.B. maximale Baumhöhe, …)
  - Verwendung verschiedener Klassifikatoren (z.B. Entscheidungsbaum, Neuronales Netzwerk, Naive Bayes, …)
  - Ein Klassifikator für jede Klasse (bei mehr als 2 Klassen)

- Ziel:
  - Klassifikatoren mit möglichst unterschiedlichen Ergebnissen

Kombinations-Ebene: Strategien

- Problem:
  - Unterschiedliche Vorgehensweisen zur Wahl der Vorhersageklasse
- Alternativen
  - Majority Vote
    - Vorhersageklasse: Ergebnis der meisten Klassifikatoren
  - Weighted Majority Vote
    - Gewichtung mit Konfidenzwerten
    - (z.B. von Entscheidungsbäumen, Nearest Neighbour)
  - Stacking
    - Ein weiterer Klassifikator zur Vorhersage der endgültigen Klasse
  - Scoring
    - Bei binären Entscheidungsproblemen, wenn Konfidenzen bekannt
    - score = confidence if class=pos
      score = 1-confidence if class=neg
      Gesamt-Score: Mittel der Scores aller Klassifikatoren
    - Setzen eines Schwellwertes zur Klassifikation
  - Weitere Strategien in der Literatur…