Mathe III: Statistische Methoden

Teil 2

Garance PARIS

Sommersemester 2010

Data Mining

1. Juli 2010

Maschinelles Lernen

- Künstliche Generierung von Wissen aus Erfahrung
- Erkennung komplexer Muster und Regelmäßigkeiten in vorhandenen Daten
- Ziel: Verallgemeinerung
  - Über das Nachschlagen bereits gesehener Beispiele hinausgehen
  - Beurteilung unbekannter Daten
- Beispiele:
  - Die Gleichung einer Geraden an Hand zweier Punkten bestimmen
  - Einen unbekannten Text mit Wortkategorien annotieren

Maschinelles Lernen

- Künstliche Generierung von Wissen aus Erfahrung
- Erkennung komplexer Muster und Regelmäßigkeiten in vorhandenen Daten
- Ziel: Verallgemeinerung
  - Über das Nachschlagen bereits gesehener Beispiele hinausgehen
  - Beurteilung unbekannter Daten
- Beispiele:
  - Die Gleichung einer Geraden an Hand zweier Punkten bestimmen
  - Einen unbekannten Text mit Wortkategorien annotieren

Data Mining

Analyse im Voraus **strukturierter** Daten

Beispiel

Der Manager eines Golf-Clubs möchte wissen, wann er viele Kunden zu erwarten hat, damit er Studenten als Aushilfe einstellen kann, und wann keiner spielen will, damit er seinen Angestellten freigeben kann.

Zwei Wochen lang führt er Buch darüber, wie das Wetter ist und ob er viele oder wenige Kunden an dem Tag hat.

Er schreibt sich auf:

- ob das Wetter heiter, bewölkt oder regnerisch ist,
- wie warm es ist,
- wieviel Luftfeuchtigkeit es gibt,
- ob der Wind stark weht oder nicht,
- und wieviele Kunden er an dem Tag hat

Beispiel: Der Wetterdatensatz

Das Ergebnis ist für einen Menschen nicht besonders gut durchschaubar. . .

Outlook	Temp.	Humidity	Windy	Play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Was kann gelernt werden?

Einige Aufgabetypen:

Klassifikation:

Instanzen vordefinierten Kategorien zuweisen

Clustering:

Instanzen ihrer Ähnlichkeit nach in Gruppen aufteilen (ohne vordefinierte Kategorien)

Numerische Vorhersage:

Der numerische Wert eines Merkmals auf Grund der Werte aller anderen Merkmale bestimmen

Beispiel: Ein numerischer Datensatz

Outlook	Temp.	Humidity	Windy	# Cust.
sunny	85	85	false	no
sunny	80	90	true	no
overcast	83	86	false	yes
rainy	70	96	false	yes
rainy	68	80	false	yes
rainy	65	70	true	no
overcast	64	65	true	yes
sunny	72	95	false	no
sunny	69	70	false	yes
rainy	75	80	false	yes
sunny	75	70	true	yes
overcast	72	90	true	yes
overcast	81	75	false	yes
rainy	71	91	true	no

Beispiel: Ein numerischer Datensatz

Outlook	Temp.	Humidity	Windy	# Cust.
sunny	85	85	false	12
sunny	80	90	true	10
overcast	83	86	false	30
rainy	70	96	false	15
rainy	68	80	false	16
rainy	65	70	true	5
overcast	64	65	true	24
sunny	72	95	false	10
sunny	69	70	false	18
rainy	75	80	false	20
sunny	75	70	true	25
overcast	72	90	true	18
overcast	81	75	false	32
rainy	71	91	true	8

Klassifikation: Terminologie

Konzept: Das, was gelernt werden soll
Bsp.: Ob unter angegebenen Bedingungen gerne gespielt wird oder nicht

Instanz: Ein einziges Beispiel im Datensatz
Bsp.: Das Wetter an einem der 14 Tagen

Attribut: Ein Merkmal einer Instanz
Bsp.: *outlook, temperature, humidity, windy*

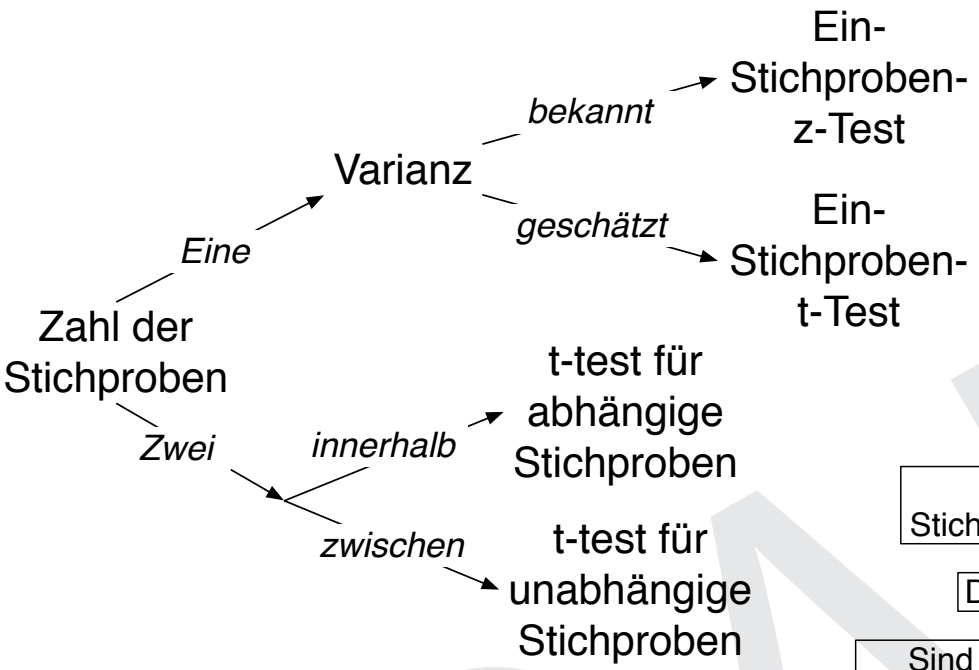
Wert: Wert eines Attributs
Bsp.: Für *outlook*: sunny, overcast, rainy

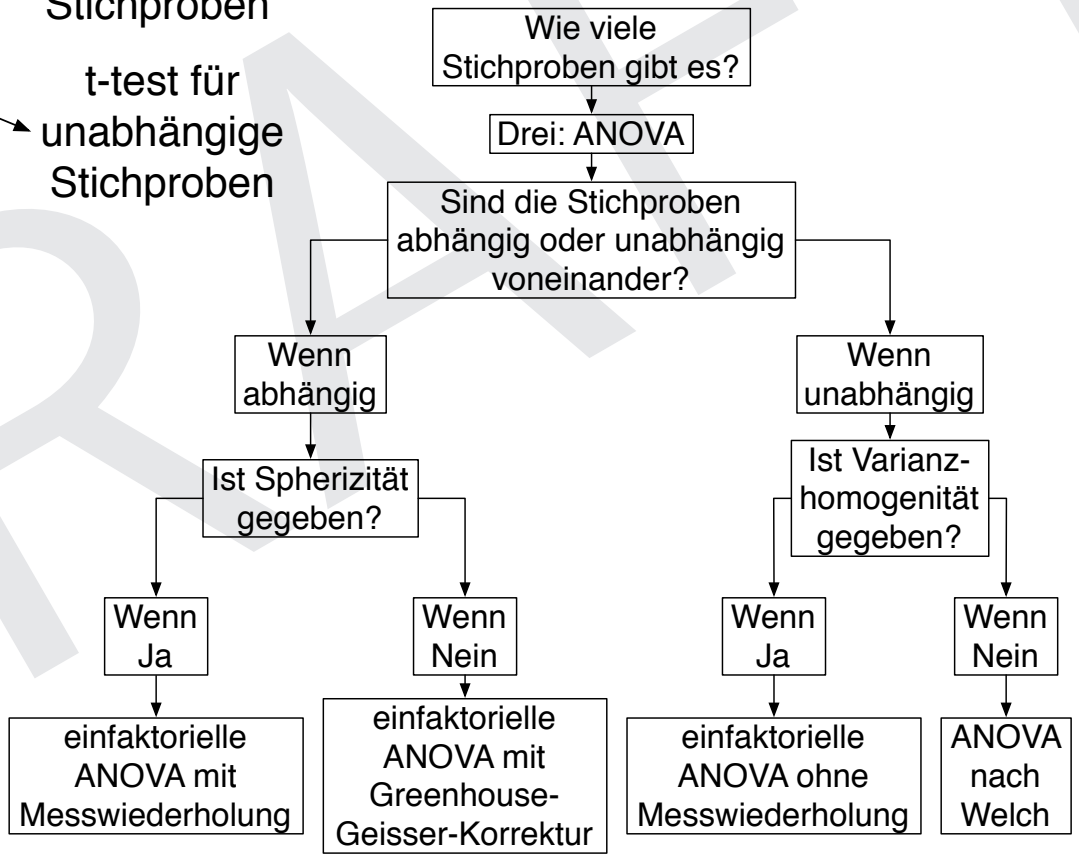
Kategorie oder Klasse:
Ziel der Klassifikation, Bsp.: yes, no

Entscheidungsbäume

1. Juli 2010

Garance PARIS





Entscheidungsbäume

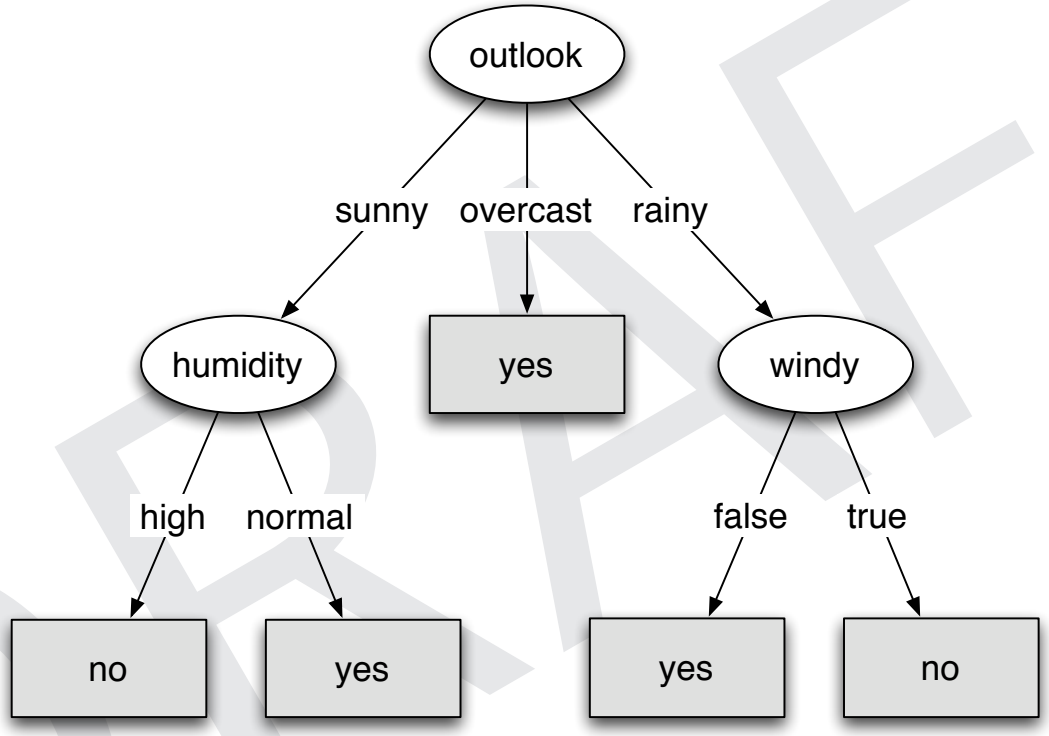
Allgemein

Ein von einem Experten erstellter Baum,
um aufeinanderfolgende, hierarchische Entscheidungen
zu veranschaulichen

Data Mining

Ein auf Basis verfügbarer Daten erstelltes Modell,
das benutzt werden kann, um Voraussagen über
neue Daten zu machen

Baum für den Wetter-Datensatz



ID3: Intuition

- Bestimmen, welches Attribut die Daten am besten klassifiziert
- Dieses Attribut als Wurzel des Baums verwenden
- Einen Zweig für jeden Wert erstellen, den das Attribut annehmen kann
- Dieses Prozess für jeden Baumzweig wiederholen, jeweils mit der Untermenge der zu klassifizierenden Daten
- Rekursiv weitermachen, bis die Klassifikation unter allen Blättern eindeutig ist

Auswahl des klassifizierenden Attribut

Frage: Wie wird das beste Attribut bestimmt?

Antwort: Es wird das Attribut gewählt, das

. . . die meiste Information beiträgt

Auswahl des klassifizierenden Attribut

Frage: Wie wird das beste Attribut bestimmt?

Antwort: Es wird das Attribut gewählt, das

. . . die meiste Information beiträgt

. . . die Entropie im Datensatz am meisten reduziert

Algorithmus: *ID3* (Kommerzieller Nachfolger: *C4.5*)

Informationsgewinn

Für jeden Attribut A:

$$Gain(S, A) = E(S) - \sum_{v \in A} \frac{|S_v|}{|S|} E(S_v)$$

S	:	Datensatz
$E(S)$	:	Entropie im Datensatz oder Untermenge davon
S_v	:	Untermenge von S für die A den Wert v hat
$\|S\|$	:	Mächtigkeit von S
$\|S_v\|$	:	Mächtigkeit von S_v

Der Informationsgewinn $Gain(S, A)$ ist die Reduzierung der Entropie, die man dadurch erwartet, dass man den Wert von Attribut A kennt

Entropie im Datensatz

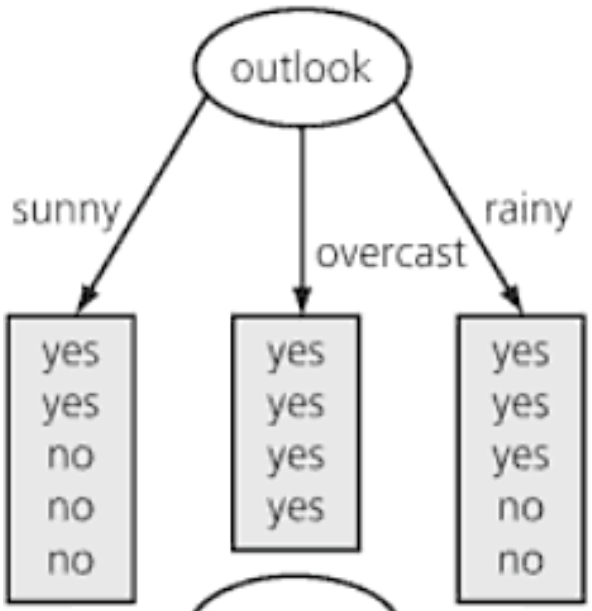
$$E(S) = -\sum_{k=1}^{c} p_k \log_2 p_k$$

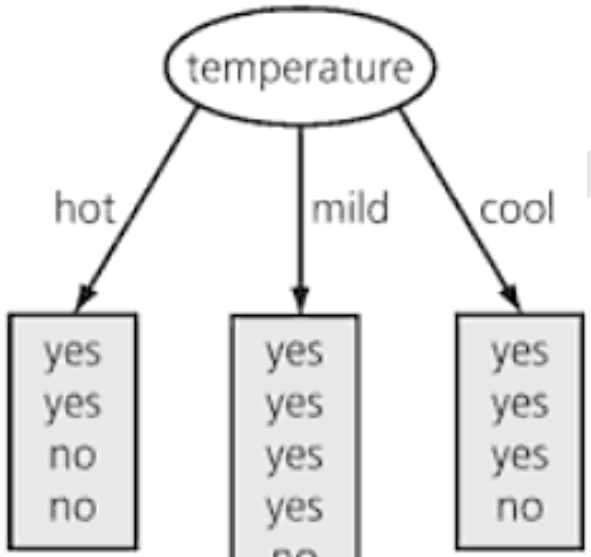
c: Anzahl Kategorien
p_k: Anteil der Instanzen in S, die Kategorie k angehören

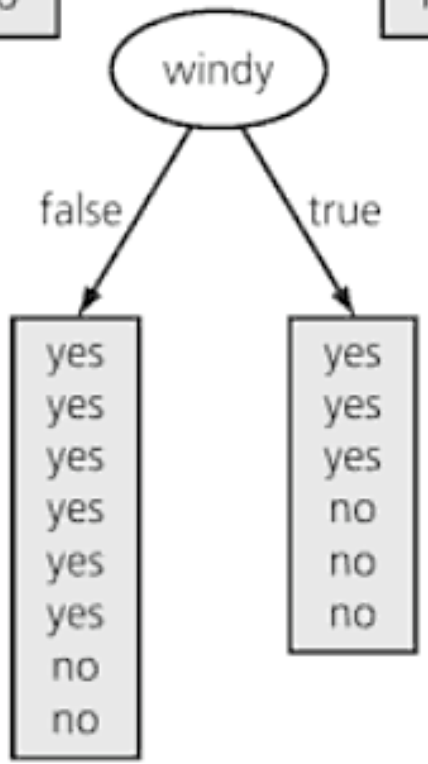
Informationstheoretische Interpretation

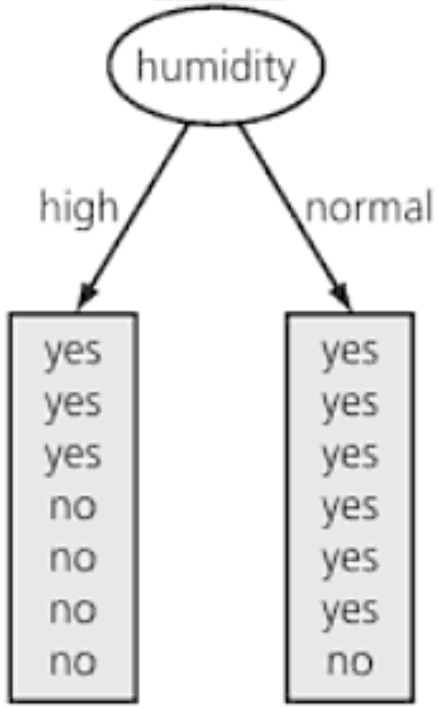
- $E(S)$: Anzahl Bits, die benötigt werden, um ein beliebiges Element aus S zu kodieren
- $E(S)$ ist gleich null, wenn alle Instanzen von S der selben Kategorie angehören
- Bei einer binären Klassifikation ist $E(S)$ gleich eins, wenn S gleich viele Instanzen aus jeder Klasse enthält

Garance PARIS









Beispiel mit dem Wetterdatensatz

- Erst die Gesamtentropie berechnen:
  Von 14 Instanzen werden 5 als *"play=yes"* klassifiziert, 9 als *"play=no"*

$$\begin{aligned} E(S) &= -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} \\ &= 0.41 + 0.53 = 0.94 \end{aligned}$$

- Dann für jedes Attribut und für jeden Wert die Entropie berechnen, z. B. für "*windy* = *false*":

$$E(S_{false}) = -\frac{6}{8}\log_2\frac{6}{8} - \frac{2}{8}\log_2\frac{2}{8}$$

Dies kann wie folgt umgeformt werden:

$$E(S_{false}) = \frac{8\log_2 8 - 6\log_2 6 - 2\log_2 2}{8} = 0.81$$

Beispiel mit dem Wetterdatensatz

- Bei "*windy* = *true*" kommen dreimal "*play* = *yes*" vor und dreimal "*play* = *no*", daher ist die Entropie gleich eins
- Daraus und aus der Gesamtentropie setzt sich $Gain(S, windy)$ zusammen:

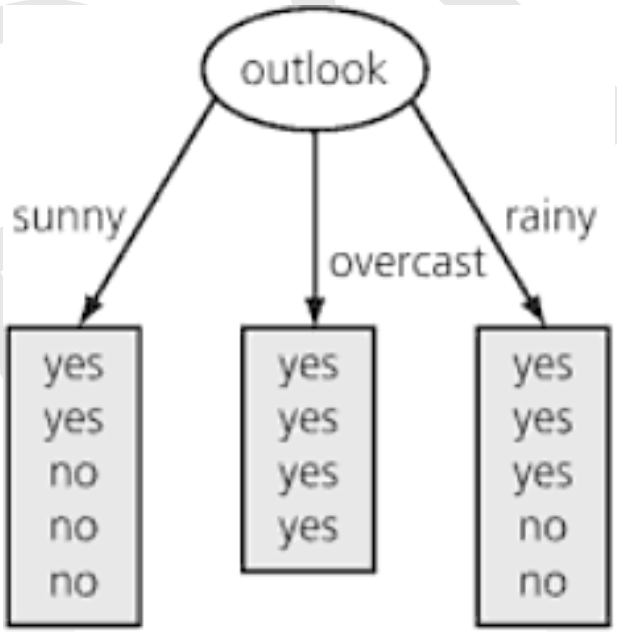
$$\begin{aligned} Gain(S, windy) &= E(S) - \frac{|S_{false}|}{|S|} E(S_{false}) - \frac{|S_{true}|}{|S|} E(S_{true}) \\ &= 0.94 - \frac{8}{14} * 0.81 - \frac{6}{14} * 1 = 0.048 \end{aligned}$$

Informationsgewinn: Beispiel

- Ähnlich werden berechnet:

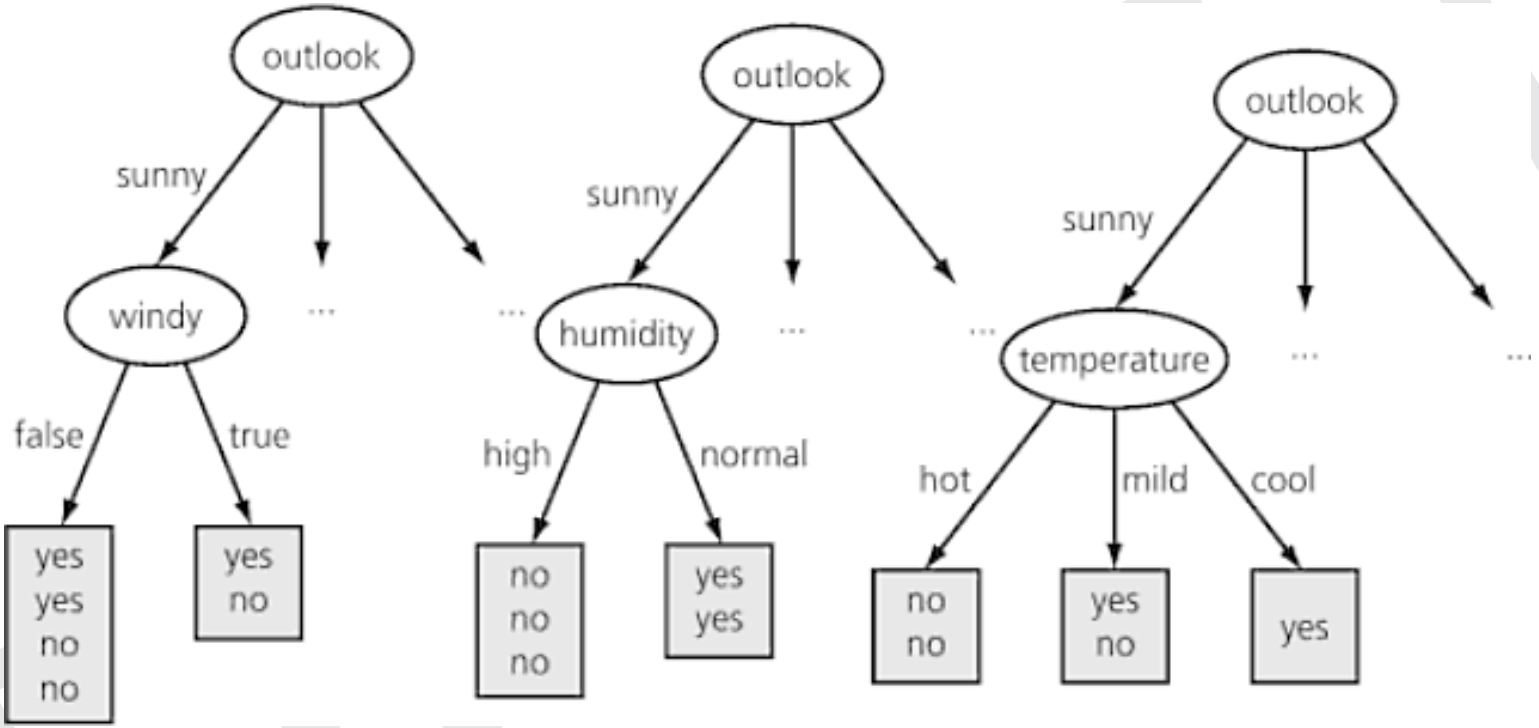
$$\textit{Gain}(S, \textit{windy}) = 0.048$$
$$\textit{Gain}(S, \textit{outlook}) = 0.247$$
$$\textit{Gain}(S, \textit{temperature}) = 0.029$$
$$\textit{Gain}(S, \textit{humidity}) = 0.152$$

- *Outlook* erzielt den größten Informationsgewinn, daher wird es als Baumwurzel gewählt



Entscheidungsbäume

Informationsgewinn: Beispiel



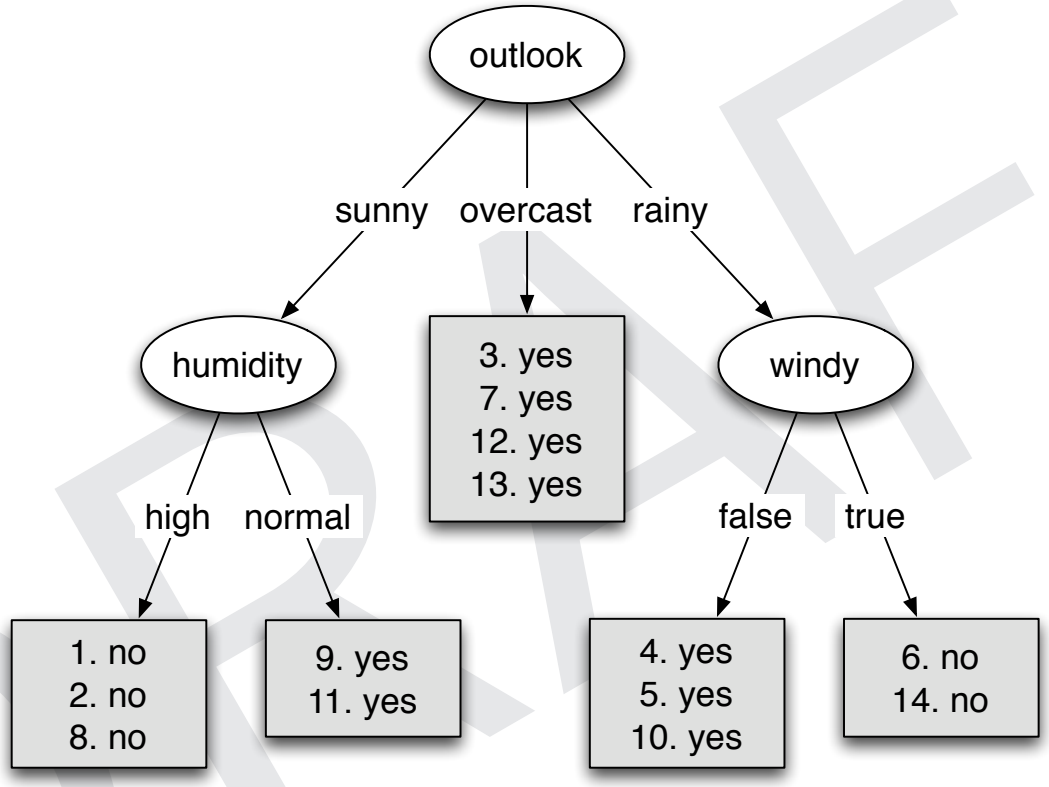
$Gain(S, windy) = 0.020$ $Gain(S, temperature) = 0.571$

$Gain(S, humidity) = 0.971$

Humidity erzielt den größten Informationsgewinn, daher wird es als nächsten Knoten unter "*outlook* = *sunny*" eingesetzt

Entscheidungsbäume

Baum für den Wetter-Datensatz



- Alle Instanzen unter einem Blatt werden gleich klassifiziert (Entropie $= 0$)
- Das Attribut *temperature* wird nicht verwendet

Entscheidungsbäume

Interpretation

- Wenn das Wetter bewölkt ist, spielen Leute immer
- Es gibt Leute, die sogar bei Regen spielen, aber nicht wenn der Wind weht
- Wenn die Sonne scheint, spielen Leute gerne, außer wenn die Luftfeuchtigkeit hoch ist

Schlußfolgerungen:

- Der Manager kann an Tagen, an denen Regen und starken Wind vorausgesagt wird, seinem Personal frei geben
- An sonnigen Tagen, an denen es Luft gibt, kann er sich überlegen, ob er zusätzlich ein paar Studenten einstellt

Entscheidungsbäume

Repräsentation

- Nicht-terminale Knoten testen ein Attribut
- Baumzweige entsprechen den Werten, die ein Attribut annehmen kann
- Blätter weisen Instanzen einer Kategorie zu
- Der Pfad zwischen Wurzel und Blatt stellt eine Konjunktion dar

Umwandlung in Entscheidungsregeln

```
IF (outlook==sunny) ∧ (humidity==high)
THEN play=no
IF (outlook==sunny) ∧ (humidity==normal)
THEN play=yes
...
```

Entscheidungsbäume

Eigenschaften des Algorithmus

- Allgemein werden flache Bäume bevorzugt
- Es findet eine Suche durch den Hypothesenraum statt
- Die Suche ist aber unvollständig, d. h. nicht alle möglichen Bäume werden in Betracht gezogen
- Es ist möglich, dass der "beste" (kleinste) Baum nicht gefunden wird
  (z.B. wenn es einen insgesamt kleineren Baum gibt, der eine andere Wurzel hat, als die, die anfangs den höchsten Informationsgewinn bietet)

Garance PARIS

Naiv-Bayes-Klassifikation

5. Juli 2010

Die Intuition

Die Wahrscheinlichkeit einer Zielkategorie für eine Instanz hängt von zwei Faktoren ab:

- die Gesamtwahrscheinlichkeit der Kategorie überhaupt, Engl. *prior probability (a-priori-Wahrscheinlichkeit)*

  Bsp.: Wie oft kommen *play=yes* bzw. *play=no* insgesamt im Datensatz vor?
- die Wahrscheinlichkeit der Kategorie bei der von der Instanz beigetragenen Evidenz oder Information, Engl. *posterior probability*

  Hier: Wahrscheinlichkeit der einzelnen Attributwerte der Instanz

Das Maximum-a-posteriori

Wir wollen für neue Instanzen die beste Kategorie gegeben der Evidenz finden, auf Englisch das *"maximum a posteriori"*:

$$c_{map} = \underset{c}{\operatorname{argmax}}\, P(c|e) \qquad \text{Bayes'scher Satz}$$

$$= \underset{c}{\operatorname{argmax}}\, \frac{P(c)P(e|c)}{P(e)} \qquad P(e) \text{ fällt weg, weil es konstant und unabhängig von der Hypothese ist}$$

$$= \underset{c}{\operatorname{argmax}}\, P(c)P(e|c)$$

Parameter-Schätzung

- $P(c)$ schätzen ist einfach: Man berechnet die relative Häufigkeit jeder Hypothese in den Trainingsdaten
- $P(e|c)$ schätzen ist wegen Sparse-Data schwieriger, da man nie genug Daten hat, damit alle mögliche Attribut-Kombinationen vorkommen
- Es wird daher angenommen, dass die Attribute unabhängig voneinander sind

$$P(e|c) = P(a_1, a_2, \ldots, a_n|c) = \prod_{i=1}^{n} P(a_i|c)$$

- $P(a|c)$: Relative Häufigkeit des Attributs a unter den Instanzen, die Kategorie c angehören

Beispiel

- Klassifizierung von Instanz *outlook* = *sunny*, *temperature* = *cool*, *humidity* = *high*, *windy* = *true*:

$$\begin{aligned} c_{map} &= \underset{c \in yes,no}{\operatorname{argmax}} P(c) \cdot \prod_i P(a_i|c) \\ &= \underset{c \in yes,no}{\operatorname{argmax}} P(c) \cdot P(sunny|c) \cdot P(cool|c) \cdot \\ &\quad P(high|c) \cdot P(true|c) \end{aligned}$$

- Berechnung der relativen Häufigkeit von $P(sunny|yes)$:

  Insgesamt gibt es 9 Instanzen mit *play* = *yes*,
  davon 2 bei denen *outlook* = *sunny*,
  also ist $P(sunny|yes) = \frac{2}{9}$

Vorgehen

- Zuerst die Wahrscheinlichkeit jeder Kategorie in den Trainingsdaten berechnen

  Gesamthäufigkeiten für "play": $yes = 9$, $no = 5$

  Wahrscheinlichkeiten: $yes = \frac{9}{14}$, $no = \frac{5}{14}$
- Für jede Kategorie die Wahrscheinlichkeit der einzelnen Attribute berechnen

		freq.		P	
		yes	*no*	*yes*	*no*
Outlook	sunny	2	3	$\frac{2}{9}$	$\frac{3}{5}$
	overcast	4	0	$\frac{4}{9}$	0
	rainy	3	2	$\frac{3}{9}$	$\frac{2}{5}$
	$\sum$	9	5	1	1
Temperature	...				
Humidity	...				
Windy	...				

Vorgehen, Teil 2

- Wahrscheinlichkeit jeder Kategorie für die vorliegende Instanz berechnen

$$P(yes) \cdot P(sunny|yes) \cdot P(cool|yes) \cdot P(high|yes) \cdot P(true|yes) = \frac{9}{14} * \frac{2}{9} * \frac{3}{9} * \frac{3}{9} * \frac{3}{9} = 0.0053$$

$$P(no) \cdot P(sunny|no) \cdot P(cool|no) \cdot P(high|no) \cdot P(true|no) = \frac{5}{14} * \frac{3}{5} * \frac{1}{5} * \frac{4}{5} * \frac{3}{5} = 0.0206$$

- Kategorie zuweisen

$$\begin{aligned} c_{map} &= \underset{c \in yes,no}{\operatorname{argmax}} P(c) \cdot \prod_i P(a_i|c) \\ &= no \end{aligned}$$

Textklassifikation

Bei der Klassifikation von Dokumenten werden die Wörter des Textes als Attribute eingesetzt:

$$\begin{aligned} c_{map} &= \underset{c}{\operatorname{argmax}}\, P(c)P(c|d) \\ &= \underset{c}{\operatorname{argmax}}\, P(c)\prod_i P(w_i|c) \end{aligned}$$

$P(w|c)$: Wahrscheinlichkeit, dass Wort w in einem
: Dokument mit Klasse c vorkommt

Klassifikation nach Bayes

Anwendung: Spam herausfiltern

- Datensatz: Ein Korpus von E-Mail-Nachrichten, annotiert als “Spam” oder “Ham”
- Aufgabe: Eine neue Nachricht als “Spam” oder “Ham” klassifizieren
- Attribute: Vokabular der E-Mail-Nachrichten im Trainingskorpus

Bsp.: Eine E-Mail mit dem Inhalt “Get rich fast!!!”

$$
\begin{aligned}
c_{map} &= \underset{c \in spam, ham}{\operatorname{argmax}} \; P(c) \cdot \prod_i P(a_i|c) \\
&= \underset{c \in spam, ham}{\operatorname{argmax}} \; P(c) \cdot \\
&\quad P(get|c) \cdot P(rich|c) \cdot P(fast|c) \cdot P(!!!|c)
\end{aligned}
$$

Anwendung: Spam Herausfiltern

- Problem: “rich” und “!!!” sind in Ham-E-Mails selten
- Null-Werte führen dazu, dass Kategorien ununterscheidbar werden

$$\begin{aligned} P(ham) &= P(c) \cdot P(get|c) \cdot P(rich|c) \cdot P(fast|c) \cdot P(!!!|c) \\ &= P(c) \cdot P(get|c) \cdot 0 \cdot P(fast|c) \cdot 0 = \boxed{0} \text{ !} \end{aligned}$$

- Mögliche Lösung: Kleine Werte zu Zähler und Nenner hinzufügen, damit beide ungleich null werden

$$P(a|c) = \frac{n_a + \frac{1}{k}}{n + 1}$$

n : Anzahl Trainingsinstanzen in Kategorie c
n_a : Anzahl Trainingsinstanzen mit Attribut a in Kategorie c
k : Anzahl Werte, die Attribut a annehmen kann

Garance PARIS

Clusteranalyse

6. Juli 2010

Clustering vs. Klassifikation

- Bei der Klassifikation werden Instanzen vordefinierten Klassen zugeordnet
- Beim Clustering entdeckt der Algorithmus "natürliche" Klassen, die die Instanzen in Gruppen mit ähnlichen Eigenschaften teilen
- Deutscher Begriff: *Ballungsanalyse*

Betreutes und unbetreutes Lernen

Betreutes Lernen (Engl. “supervised learning”):

Der Algorithmus lernt, indem er Eingabe-Ausgabe-Beispiele analysiert, die die Rolle eines “Lehrers” übernehmen

Unbetreutes Lernen (Engl. “unsupervised learning”):

Vorkategorisierte Beispiele sind nicht verfügbar

Betreutes und unbetreutes Lernen

Halb-betreutes Lernen (Engl. "semi-supervised learning")

- Nur ein Teil der Trainingsbeispiele ist bereits mit einer Kategorie annotiert
- Kombination aus betreutes und unbetreutes Lernen

Bestärkendes Lernen (Engl. "reinforcement learning")

- Die Lernkomponente ist Teil eines Systems oder Agents, das lernen soll, wie in potenziell auftretenden Situationen zu handeln ist
- Der Algorithmus erfährt aber nicht explizit, was die richtige Kategorisierung wäre
- Stattdessen lernt er durch Feedback von der Umwelt, das ihm in der Form von Belohnung und Bestrafung gegeben wird, den Erfolg des Agenten zu maximieren

Wozu Clustering verwenden?

Explorative Datenanalyse

Um ein Gefühl für die vorhandenen Daten
und ihre Eigenschaften zu gewinnen

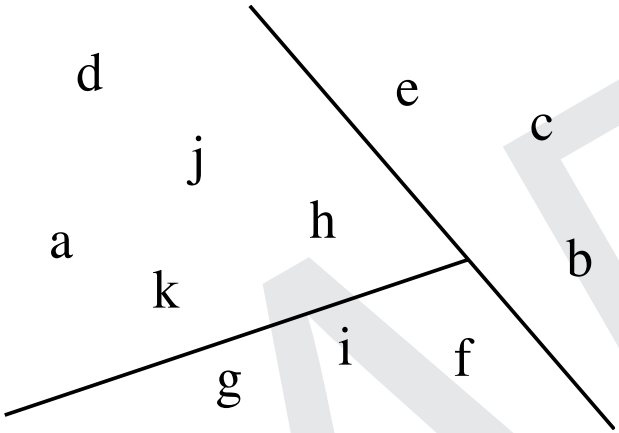
Binning

Instanzen entdecken, die sich ähnlich verhalten
und daher ähnlich behandelt werden können,
um Abhilfe bei Sparse-Data-Problemen zu schaffen

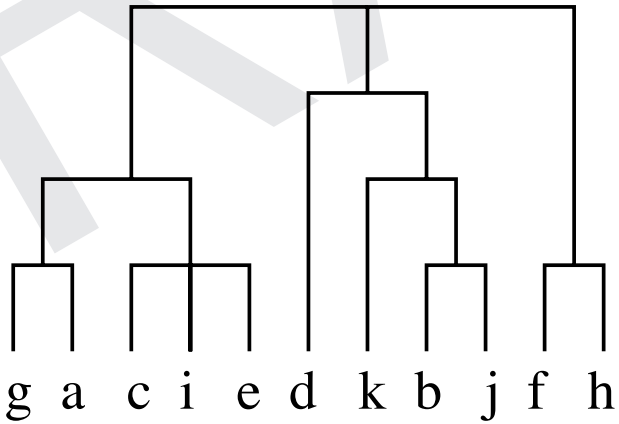
- In einem Korpus findet man die Sequenzen
  "am Donnerstag" und *"am Freitag"*
  sowie *donnertags* und *freitags*
- Außerdem hat man *"am Montag"*,
  aber *montags* kommt nicht vor
- Wenn wir wissen, dass *Donnerstag*, *Freitag*
  und *Montag* sich syntaktisch ähnlich verhalten,
  können wir *montags* inferieren

Verschiedene Clustering-Arten

Flach



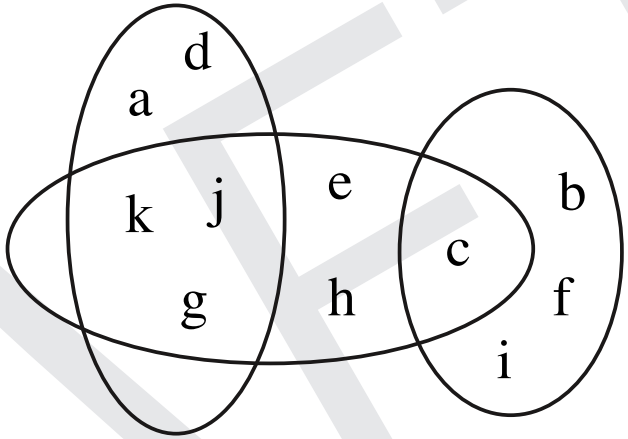
Hierarchisch (Dendrogramme)



Verschiedene Clustering-Arten

Disjunktiv

Instanzen können mehreren Cluster angehören



Probabilistisch oder "soft"

Für jeden Cluster wird eine Wahrscheinlichkeit angegeben, dass eine Instanz ihm zugeordnet wird

	1	2	3
a	0.4	0.1	0.5
b	0.1	0.8	0.1
c	0.3	0.3	0.4
d	0.1	0.1	0.8
e	0.4	0.2	0.4
f	0.1	0.4	0.5
g	0.7	0.2	0.1
h	0.5	0.4	0.1

Iteratives distanz-basiertes Clustering

Intuition bei k-means

- Bestimme k, die Anzahl der gewünschten Cluster
- Wähle k beliebige Punkte als Cluster-Zentren aus
- Weise jede Instanz dem nächsten Cluster-Zentrum zu
- Berechne den Mittelpunkt für jeden Cluster und verwende ihn als neues Zentrum
- Weise alle Instanzen wieder dem nächsten Cluster-Zentrum zu
- Iteriere, bis alle Cluster stabil sind

Iteratives distanz-basiertes Clustering

Der Algorithmus

- Jede Instanz $\vec{x}$ im Training Set wird als Vektor mit einem Wert pro Attribut repräsentiert

$$\vec{x} = (x_1, x_2, \ldots, x_n)$$

- Die Distanz zwischen zwei Vektors $\vec{x}$ and $\vec{y}$ ist definiert als (euklidische Distanz):

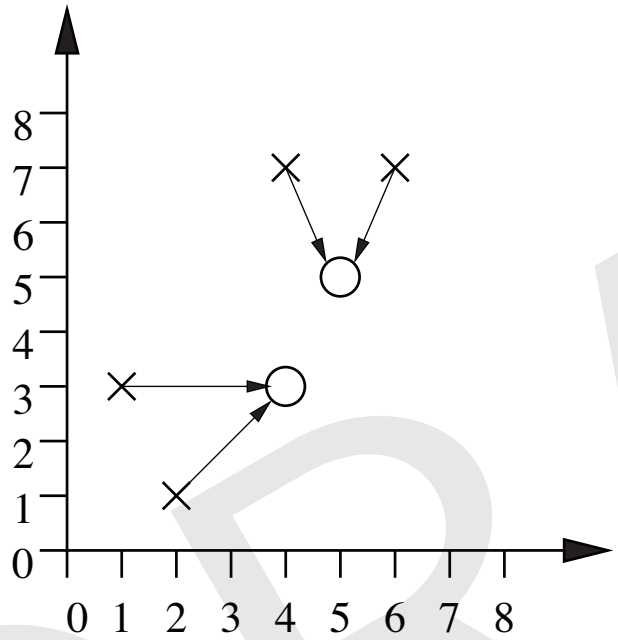
$$|\vec{x} - \vec{y}| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

- Der Mittelpunkt $\vec{\mu}$ einer Menge Vektoren c_j ist definiert als:

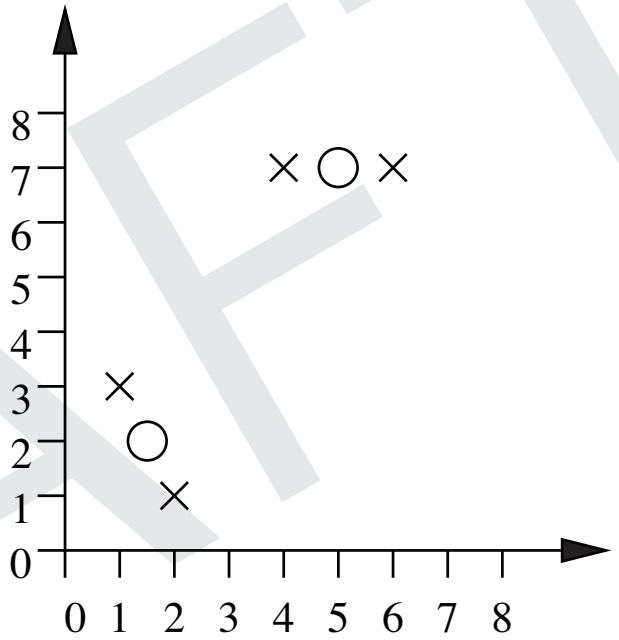
$$\vec{\mu} = \frac{1}{|c_j|} \sum_{\vec{x} \in c_j} \vec{x}$$

Iteratives distanz-basiertes Clustering

Beispiel



Die Instanzen (Kreuzchen) werden anfangs zum nächsten Cluster-Zentrum (Kreise) zugewiesen



Der Mittelpunkt jedes Cluster wird dann berechnet und als neuen Zentrum verwendet

Iteratives distanz-basiertes Clustering

Eigenschaften von k-means

- Flaches Clustering-Verfahren
- Konzeptuell einfach
- Effizient bei großen Datenmengen
- Nicht geeignet für Nominaldaten
- Findet nur ein lokales Maximum, keine globales
- Die Cluster hängen sehr von der initialen Wahl der Cluster-Zentren
- Kann man für hierarchisches Clustering verwenden
- Andere Distanzmaße können verwendet werden (z. B. der Cosinus)
- Alternative: der EM-Algorithmus
  - Legt iterativ die Parameter eines Modells so fest, dass es die gesehenen Daten optimal erklärt
  - HMMs sind eine Anwendung des EM-Algorithmus