Begriffslernen

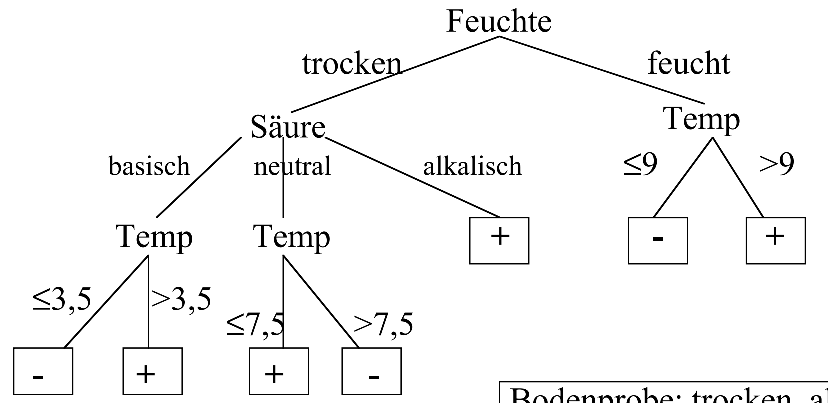
Gegeben: eine Menge klassifizierter Beobachtungen (Beispiele) für Klassen (Begriffe)

Finde: einen Entscheidungsbaum, der eine neue Beobachtung klassifiziert

Alternativ: finde eine Entscheidungsfunktion, die eine Beobachtung klassifiziert (Regressionsbaum)

- Aussagenlogik (Attribut- Werte)
- Numerische Werte

Lernen von Entscheidungsbäumen



Bodenprobe: trocken, alkalisch, 7
wird als geeignet klassifiziert (+)

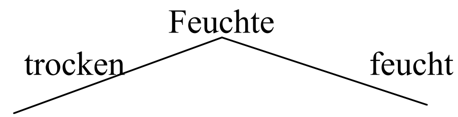
Bodeneignung für Rotbuchen

Beispiele

+ ID	Feuchte	Säure	Temp	- ID	Feuchte	Säure	Temp
1	trocken	basisch	7	2	feucht	neutral	8
3	trocken	neutral	7	4	feucht	alkal.	5
6	trocken	neutral	6	5	trocken	neutral	8
9	trocken	alkal.	9	7	trocken	neutral	11
10	trocken	alkal.	8	8	trocken	neutral	9
12	feucht	neutral	10	11	feucht	basisch	7
13	trocken	basisch	6	14	feucht	alkal.	7
16	trocken	basisch	4	15	trocken	basisch	3

Ohne weiteres Wissen können wir als Vorhersage immer - sagen.
Der Fehler ist dann 8/16.

Aufteilen nach Bodenfeuchte



1 basisch 7 +
3 neutral 7 +
5 neutral 8 -
6 neutral 6 +
7 neutral 11 -
8 neutral 9 -
9 alkal. 9 +
10 alkal. 8 +
13 basisch 6 +
15 basisch 3 -
16 basisch 4 +

2 neutral 8 -
4 alkal. 5 -
11 basisch 7 -
12 neutral 10 +
14 alkal. 7 -

Vorhersage der häufigsten Klasse:
11/16 trocken +: Fehler 4/11
5/16 feucht -: Fehler 1/5
Fehler bei Information über Feuchte:
11/16 4/11 + 5/16 1/5 = 5/16

Bedingte Wahrscheinlichkeit

- Wahrscheinlichkeit, dass ein Beispiel zu einer Klasse gehört, gegeben der Attributwert
- $P(A|B) = P(A \cap B) / P(B)$
- Annäherung der Wahrscheinlichkeit über die Häufigkeit
- Gewichtung bezüglich der Oberklasse

Beispiel:

P(+ |feucht) = 1/5 P(- |feucht) = 4/5 gewichtet mit 5/16

P(+ |trocken) = 7/11 P(- | trocken) = 4/11 gewichtet mit 11/16

Wahl des Attributes mit dem höchsten Wert (kleinsten Fehler)

Information eines Attributs

- Wir betrachten ein Attribut als Information.
- Wahrscheinlichkeit p_i, dass das i-te Symbol auftritt, entspricht der Wahrscheinlichkeit, dass das Beispiel der i-ten Klasse entstammt.

  $I(p_1,...p_m)= \sum_{i=1} -p_i \log p_i$ (Entropie)
- Ein Attribut mit *k* Werten teilt eine Menge von Beispielen E in *k* Untermengen auf. Für jede dieser Mengen berechnen wir die Entropie.

  Güte(Attribut, E):= $- \sum^{k}_{i=1} |E_i| / |E| \; I(p_1,...p_m)$

Beispiel: Feuchte

Güte des Attributs *Feuchte* mit den 2 Werten *trocken* und *feucht*:

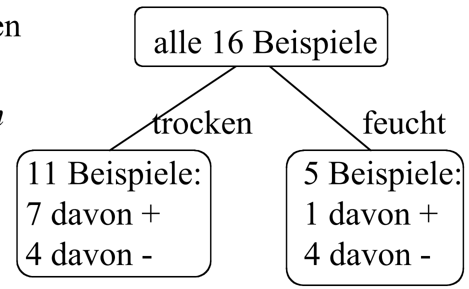
- (11/16 I(+,-)   *trocken*

  + 5/16 I(+,-)) =   *feucht*

- (11/16 (-7/11 log 7/11 +
  -4/11 log 4/11)

  + 5/16 (-1/5 log 1/5 +
  -4/5 log 4/5)) =

- 0,27



Säure

Güte des Attributs *Säure* mit den 3 Werten *basisch, neutral* und *alkalisch*:

- (5/16 I(+,-)   *basisch*

  + 7/16 I(+,-)   *neutral*

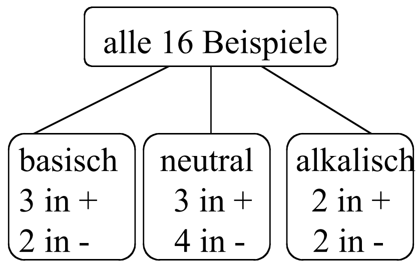
  + 4/16 I(+,-)) =   *alkalisch*

  -0,3

basisch: - 3/5 log 3/5 +
-2/5 log 2/5

neutral: -3/7 log 3/7 +
-4/7 log 4/7

alkalisch: - 2/4 log 2/4 +
-2/4 log 2/4



Temperatur

- Numerische Attributwerte werden nach Schwellwerten eingeteilt.
- 9 verschiedene Werte in der Beispielmenge, also 8 Möglichkeiten, zu trennen.
- Wert mit der kleinsten Fehlerrate bei Vorhersage der Mehrheitsklasse liegt zwischen 6 und 7.
- 5 Beispiele mit Temp < 7, davon 3 in +,
  11 Beispiele Temp ≥ 7, davon 6 in -.
- Die Güte der Temperatur als Attribut ist - 0,29.

Beispiel: Attributselektion

- Gewählt wird das Attribut, dessen Werte am besten in (Unter-)mengen aufteilen, die geordnet sind.
- Das Gütekriterium Information bestimmt die Ordnung der Mengen.
- Im Beispiel hat *Feuchte* den höchsten Gütewert.

Algorithmus ID3

Rekursive Aufteilung der Beispielmenge nach Selektion eines Attributs:

TDIDT(E, Attribute)

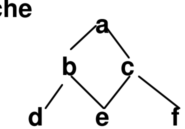
- E enthält nur Beispiele einer Klasse --> Blatt
- E enthält Beispiele verschiedener Klassen:
  - Güte (Attribute,E)
  - Wahl des Attributs *a* in *Attribute* mit *k* Werten
  - Aufteilung von E in $E_1, E_2, ..., E_k$
  - für i=1, ..., k: TDIDT(E_i, Attribute \ a}
- Resultat ist aktueller Knoten mit den Teilbäumen $T_1, ..., T_k$

conceptual clustering

Gegeben: eine Menge von Beobachtungen
Ziel: eine Hierarchie von Begriffsdefinitionen, die neue Beobachtungen vorhersagen

STAR Methode:

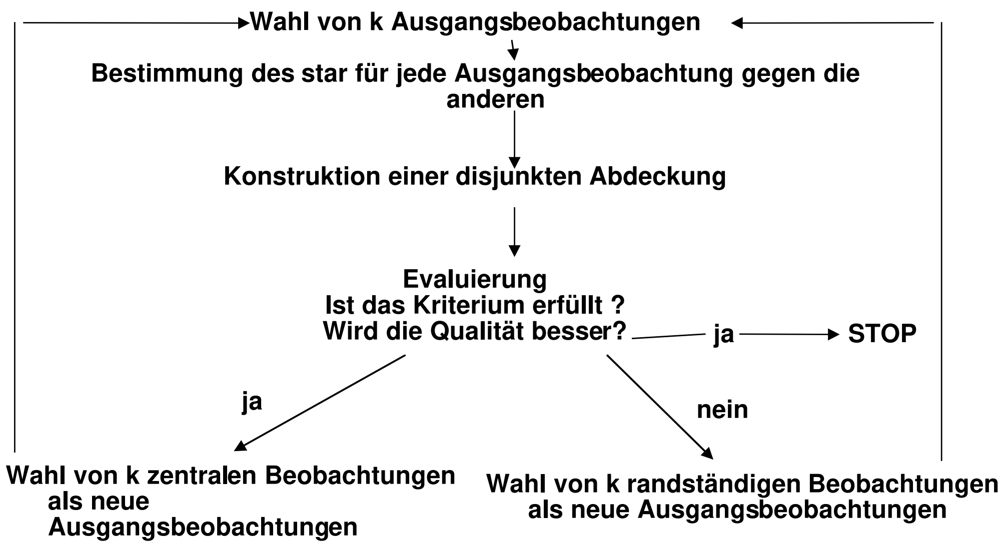
Eine Beobachtung ist eine Beschreibung.
Die Beschreibungssprache ist eine Prädikatenlogik, wobei Wertebereiche vorgegeben werden. Z.B.: lineare Wertebereiche, nominale Wertebereiche, hierarchische Wertebereiche



Ein *cluster* ist eine intensional definierte Menge von Beobachtungen.
Ein *star* ist eine abgrenzende Beschreibung (elementares *cluster*).

Ryszard Michalski, Robert Stepp 1983

Algorithmus

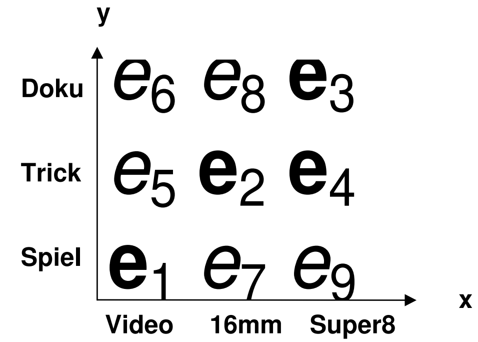


Beispiel

Beobachtungen:
e1: (Video, Spielfilm)
e2: (16mm, Trickfilm)
e3: (Super8, Dokumentarfilm)
e4: (Super8, Trickfilm)

Wertebereiche:
X: {Video, 16mm, Super8 }
Y: { Spielfilm, Trickfilm,Dokumentarfilm}

mögliche, nicht beobachtete Objekte:
e5: (Video, Trickfilm)
e6: (Video, Dokumentarfilm)
e7: (16mm,Spielfilm)
e8: (16mm, Dokumentarfilm)
e9: (Super8, Spielfilm)



star bilden
Generalisierung

Was ist bei einer Beobachtung das Besondere bezüglich einer anderen?
alle möglichen Charakterisierungen

G(e1|e4): [x≠ Super8] OR [y≠ Trick]
= [x= Video or 16mm] OR [y= Spiel or Doku]

G(e4|e1): [x≠ Video] OR [y≠ Spiel]
= [x= 16mm or Super8] OR [y= Trick or Doku]

Die <u>Disjunktion</u> G(e1|e4) schließt nur e4 aus und deckt maximal viele andere Beobachtungen ab, nämlich e1, e2, e3.
Jede einzelne Beschreibung schließt mehr aus:
[x= Video or 16mm] deckt e1, e2 ab, schließt e3, e4 aus.
[y= Spiel or Doku] deckt e1, e3 ab, schließt e2, e4 aus.
Eine Konjunktion der Beschreibungen würde nur e1 abdecken, e2, e3, e4 ausschließen. Die Konjunktion ist nicht die maximale Generalisierung.

star reduzieren
Spezialisierung

Die einzelnen Beschreibungen des elementaren *stars* werden nun spezialisiert.
[x=Video or 16mm] kommt bei den Beobachtungen nur zusammen mit [y= Spiel or Trick] vor.
Nehmen wir an, das sei eine Gesetzmäßigkeit und spezialisieren wir zu [x=Video or 16mm] AND [y=Spiel or Trick] . Das deckt e1,e2 ab und schließt e3, e4 aus.
Genauso spezialisieren wir
[y= Spiel or Doku] zu [y= Spiel or Doku] AND [x=Video or Super8].,
Das deckt e1, e3 ab und schließt e2, e4 aus.

RG(e1|e4): a) [x= Video or 16mm] AND [y= Spiel or Trick] OR e1, e2, e3
c) [x= Video or Super8] AND [y=Spiel or Doku]
Bei G(e4|e1) bilden x und y keine verschiedenen Spezialisierungen.
RG(e4|e1): b) [x= 16mm or Super8] AND[y= Trick or Doku]

e2, e3, e4

disjunkte Abdeckung (cover)

Die Glieder der spezialisierten *stars* werden zu überschneidungsfreien Abdeckungen aller möglichen Beobachtungen kombiniert.

1) a) [x= Video or 16mm] AND [y= Spiel or Trick] deckt e1, e2 ab
   b) [x= 16mm or Super8] AND [y= Trick or Doku] deckt e2, e3, e4 ab
   Diese Kombination muß modifiziert werden, damit e2 nicht doppelt abgedeckt wird.
   b') [x= Super8] AND [y= Trick or Doku] deckt e3, e4 ab

2) c) [x= Video or Super8] AND [y= Spiel or Doku] deckt e1, e3 ab
   b) [x= 16mm or Super8] AND [y= Trick or Doku] deckt e2, e3, e4 ab
   Diese Kombination muß modifiziert werden, damit e3 nicht doppelt abgedeckt wird.
   b'')[x= 16mm or Super8] AND [y= Trick] deckt e2, e4 ab

Evaluierung (Lexical Evaluation Function)

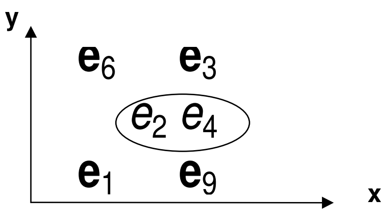
Die LEF kann vom Benutzer definiert werden. Ein ganz einfaches Maß ist:

$$1 - \frac{\text{Anzahl der abgedeckten beobachteten Objekte}}{\text{Anzahl aller abgedeckten Objekte}}$$

1a) 1 - 2/4 = 1/2
1b') 1 - 2/2 = 0 Bewertung von 1') ist die Summe: 1/2
2c) 1 - 2/4 = 1/2
2b'') 1 - 2/2 = 0 Bewertung von 2') ist die Summe: 1/2
In diesem Fall ist die Bewertung gleich. Gewählt wird zufällig 2').

Ergebnis einer Iteration

Es gibt jetzt zwei cluster:
2') c) [x= Video or Super8] AND [y= Spiel or Doku] deckt e1, e3, e6, e9 ab
b'') [x= 16mm or Super8] AND [y= Trick] deckt e2, e4 ab



Es gibt also Trickfilme auf den klassischen Filmträgern. Und:
Es gibt bei Heimfilmen Spiel- oder Dokumentarfilme.

In weiteren Schritten könnten jetzt *cluster* innerhalb der *cluster* gefunden werden.

Alternative