Folien zu “Data Mining”
von
I. H. Witten und E. Frank

übersetzt von N. Fuhr

Von Naivem Bayes zu Bayes'schen Netzwerken

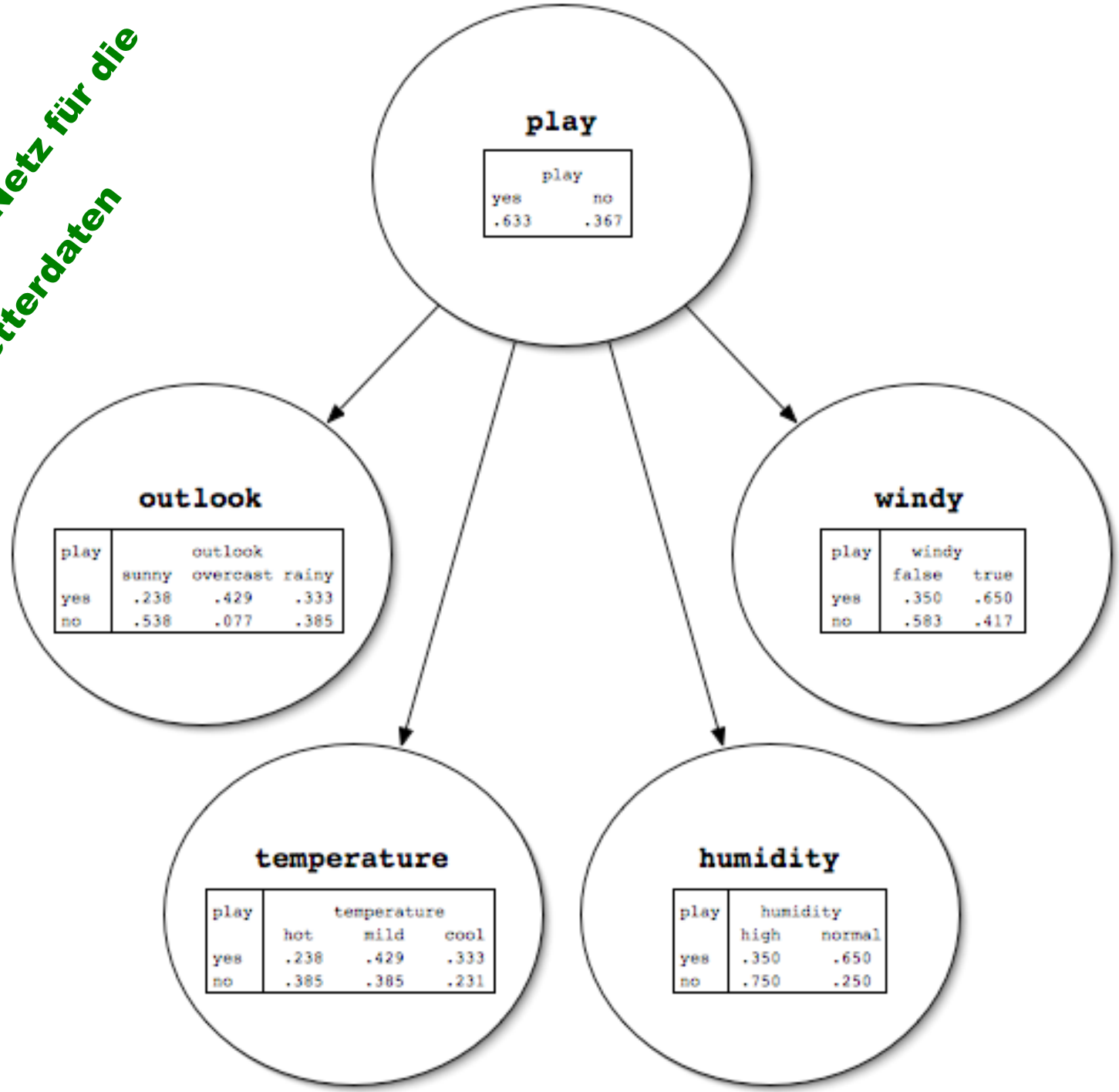
Naiver Bayes

- Annahme: Attribute bedingt unabhängig bei gegebener Klasse
- Stimmt in der Praxis meist nicht, liefert aber überraschend gute Ergebnisse
- Aber: Manchmal Qualität wesentlich schlechter als z.B. beim Entscheidungsbaum
- Kann die Unabhängigkeitsannahme fallengelassen werden?

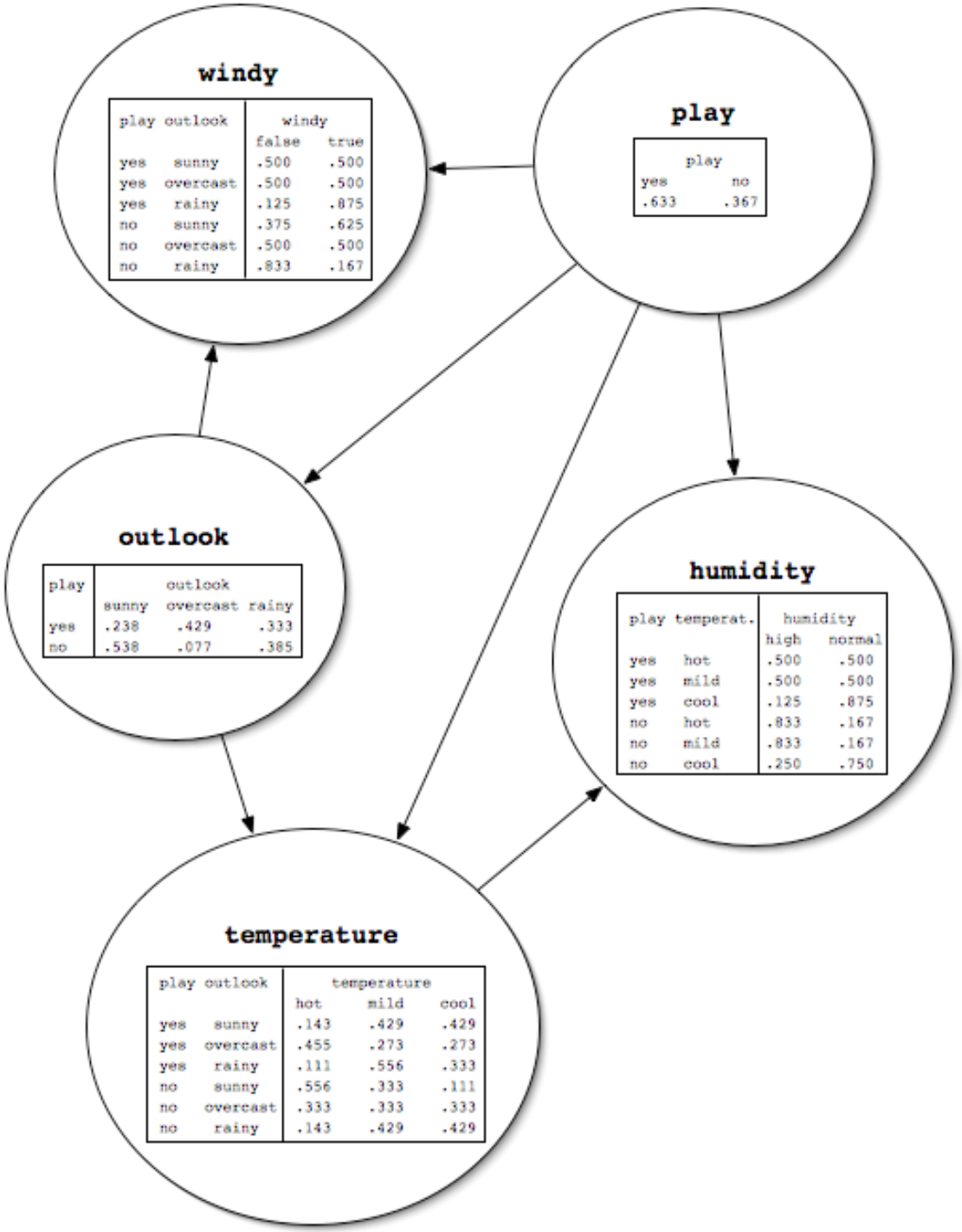
Bayes'sche Netzwerke

- Graphische Modelle, die beliebige Wahrscheinlichkeitsverteilungen modellieren können
- Graphische Repräsentation: gerichteter azyklischer Graph, ein Knoten pro Attribut
- Gesamt-Wahrscheinlichkeitsverteilung zerlegt in einzelne Komponenten
- Knoten des Graphen beinhalten die Komponenten-Verteilungen (bedingte Verteilungen)

Bayes'sches Netz für die Wetterdaten



Bayes'sches Netz für die Wetterdaten



Netzwerk vs. Naiver Bayes

Bayessches Netzwerk:

P(windy=true,outlook=sunny, temperature=cool, humidity=high| play=yes)=

P(windy=true|outlook=sunny,play=yes) *
P(outlook=sunny|play=yes) *
P(temperature=cool|outlook=sunny,play=yes) *
P(humidity=high|temperature=cool,play=yes)

Naiver Bayes:

P(windy=true,outlook=sunny, temperature=cool, humidity=high| play=yes)=

P(windy=true|play=yes)*P(outlook=sunny|play=yes) *
P(temperature=cool|play=yes) * P(humidity=high|play=yes)

Berechnung der Klassenwahrscheinlichkeiten

- Zwei Schritte: Berechnung eines Produktes von Wahrscheinlichkeiten pro Klasse, anschließend Normalisierung
  - Für jeden Klassenwert
    - Nehme alle Attributwerte und den Klassenwert
    - Suche den zugehörigen Wert der bedingten Wahrscheinlichkeit in der Tabelle der Wahrscheinlichkeitsverteilung
    - Berechne das Produkt aller Wahrscheinlichkeiten
  - Dividiere das Produkt für jede Klasse durch die Summe der Produkte (Normalisierung))

Begründung der Vorgehensweise I

- Einzige Annahme: Werte der Elternknoten bestimmen vollständig die Verteilung des jeweiligen Knotens

$$Pr[node|ancestors]=Pr[node|parents]$$

- D.h., der Knoten/das Attribut ist bedingt unabhängig von anderen Knoten bei gegebenen Werten der Elternknoten

Begründung der Vorgehensweise II

- Kettenregel der Wahrscheinlichkeitsrechnung:

$$Pr[a_1, a_2, .. a_n] = \prod_{i=1}^{n} Pr[a_i | a_{i-1}, \ldots, a_1]$$

- Aufgrund unserer Annahmen (vorige Folie):

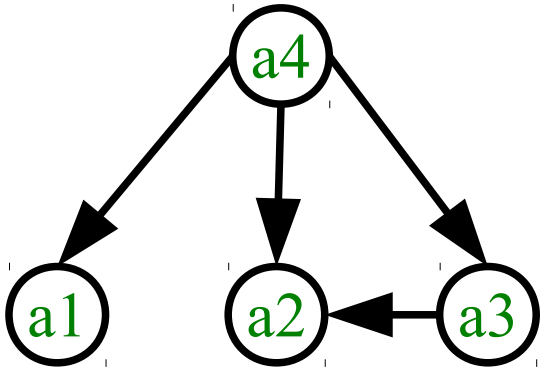
$$Pr[a_1, a_2, .. a_n] = \prod_{i=1}^{n} Pr[a_i | a_{i-1}, \ldots, a_1] = \prod_{i=1}^{n} Pr[a_i | a_i's\ parents]$$

Kettenregel

$P(a1,a2,a3,a4)=P(a1|a2,a3,a4) * P(a2,a3,a4)$
$P(a2,a3,a4)= P(a2|a3,a4) * P(a3,a4)$
$P(a3,a4)=P(a3|a4)*P(a4)$

$$P(a1,a2,a3,a4)=P(a1|a2,a3,a4) * P(a2|a3,a4) * P(a3|a4) * P(a4)$$

$P(a1,a2,a3,a4)=$
$P(a1|a4)*P(a2|a3,a4)*P(a3|a4)*P(a4)$



Lernen von Bayes'schen Netzen

- Grundlegende Komponenten von Algorithmen zum Lernen Bayes'scher Netze:
  - Methode zur Bestimmung der Güte eines gegebenen Netzwerks
    - Maß basiert auf der Wahrscheinlichkeit der Trainingsdaten bei gegebenem Netzwerk (oder dem Logarithmus hiervon)
  - Methode zum Durchsuchen des Lösungsraums der möglichen Netzwerke
    - Nur die Kantenmengen müssen betrachtet werden, da die Knoten feststehen

Problem: Überadaption

- Man kann nicht einfach die Wahrscheinlichkeit der Trainingsdaten maximieren
  - Dann ist es nämlich immer besser, mehr Kanten zu nehmen (um die Daten besser zu approximieren)
- Man muss Kreuzvalidierung benutzen oder eine Art von Strafe für die Komplexität des Netzwerks
  - AIC-Maß $-LL+K$
  - MDL-Maß: $-LL+\frac{K}{2}\log(N)$

  LL: Log-Likelihood (Logarithmus der Wahrscheinlichkeit der Daten),

  K: Anzahl der freien Parameter, *N* : #Instanzen
- Andere Möglichkeit: Bayes'scher Ansatz mit a-priori-Verteilung über die Netzwerke

Suche nach einer guten Struktur

- Aufgabe kann vereinfacht werden: Man kann jeden Knoten einzeln optimieren
  - Denn die Wahrscheinlichkeit einer Instanz ist das Produkt der Wahrscheinlichkeiten der einzelnen Knoten
  - Funktioniert auch für die Kriterien AIC und MDL, da die Kosten sich addieren
- Man kann einen Knoten optimieren durch Hinzufügen oder Löschen von Kanten zu anderen Knoten
- Dabei dürfen keine Zyklen entstehen!!

Der K2-Algorithmus

- Beginnt mit einer gegebenen Reihenfolge der Knoten (Attribute)
- Prozessiert die Knoten nacheinander
- Versucht gierig, Kanten von vorherigen Knoten zum jetzigen Knoten hinzuzufügen
- Geht zum nächsten Knoten, wenn der aktuelle Knoten nicht weiter verbessert werden kann
- Ergebnis hängt von der vorgegebenen Reihenfolge ab

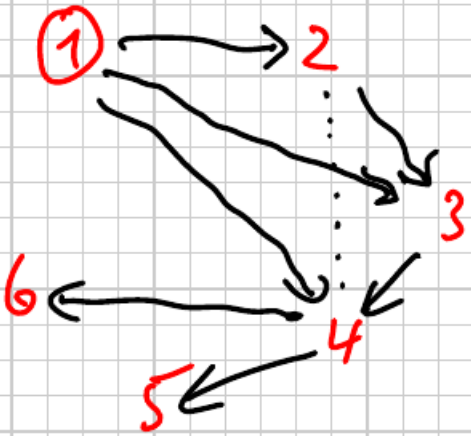
Einige Tricks

- Manchmal hilft es, die Suche mit einem naiven Bayes'schen Netzwerk zu beginnen
- Es kann ebenfalls helfen, sicherzustellen dass jeder Knoten eine Markov-Überdeckung des Klassenknotens ist
  - Die Markov-Überdeckung eines Knotens beinhaltet alle Eltern, Kinder und Eltern der Kinder des Knotens
- Bei gegebener Markov-Überdeckung ist ein Knoten bedingt unabhängig von den Knoten außerhalb der Überdeckung
  - D.h. der Knoten ist irrelevant für die Klassifikation, wenn er nicht zur Markov-Überdeckung des Klassenknotens gehört

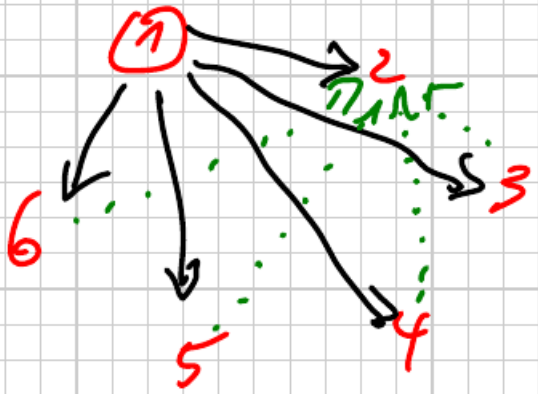
Andere Algorithmen

- Erweitere K2, um gierig Knoten zwischen beliebigen Paaren von Knoten hinzuzufügen bzw. zu löschen
  - Weiterer Schritt: Auch Richtungsumkehr der Kanten in Betracht ziehen
- TAN (Tree Augmented Naïve Bayes):
  - Beginnt mit naivem Bayes
  - Betrachtet die Möglichkeit, zweiten Elternknoten zu jedem Knoten hinzuzufügen (abgesehen vom Klassenknoten)
  - Es gibt einen effizienten Algorithmus hierfür

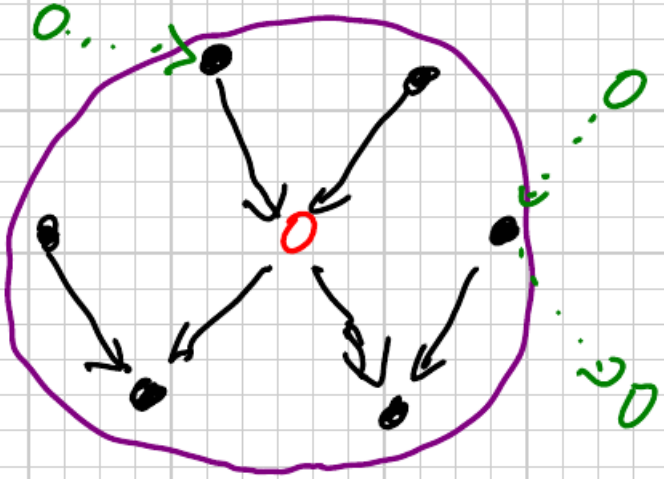
k2-Algorithmus



TAN



Markov-Überdeckung



Markov-Überdeckung des Klassenknotens: nur diese Knoten relevant für Klassenentscheidung

Wahrscheinlichkeit vs. bedingte Wahrscheinlichkeit

- Das eigentliche Ziel bei der Klassifikation ist die Bestimmung der Klasse mit der maximalen Wahrscheinlichkeit bei gegebenen Attributwerten
  - *Nicht* die Wahrscheinlichkeit der Instanzen
- Aber: es gibt keine geschlossenen Formel für die Wahrscheinlichkeiten in den Knotentabellen, die diese maximieren würden
- Aber: man kann einfach die bedingte Wahrscheinlichkeit der Daten bei gegebenem Netzwerk berechnen
- Das scheint für den Vergleich von Klassenwahrscheinlichkeiten gut zu funktionieren

Diskussion

- Betrachtet wurde hier der Fall: Diskrete Werte, keine fehlenden Werte, keine neuen Knoten
- Andere Methode zur Benutzung Bayesscher Netze für die Klassifikation: *Bayes'sche Multinetze*
  - D.h. konstruiere ein Netzwerk pro Klasse und mache die Vorhersage unter Anwendung der Bayes'schen Regel
- Andere Klasse von Lernverfahren: Testen der Annahmen bedingter Unabhängigkeit