Vorlesungsplan

- 17.10. Einleitung
- 24.10. Ein- und Ausgabe
- 31.10. Reformationstag, Einfache Regeln
- 7.11. Naïve Bayes, Entscheidungsbäume
- 14.11. Entscheidungsregeln, Assoziationsregeln
- 21.11. Lineare Modelle, Instanzbasiertes Lernen
- 28.11. Clustering
- 5.12. Evaluation
- 12.12. Evaluation
- 19.12. Lineare Algebra für Data Mining
- 9.1. Statistik für Data Mining
- 16.1. Lineare Modelle, Support Vector Machines (SVM)
- 19.1. **Vorlesung statt Übung: Bayes-Netze**
- 23.1. Clustering
- 26.1. Vorlesung statt Übung: Kombination von Modellen, Lernen von nicht-klassifizierten Beispielen
- 30.1. Finden von häufigen Teilstrukturen
- 2.1. Klausur

Von Naïve Bayes zu Bayesischen Netzwerk-Klassifikatoren

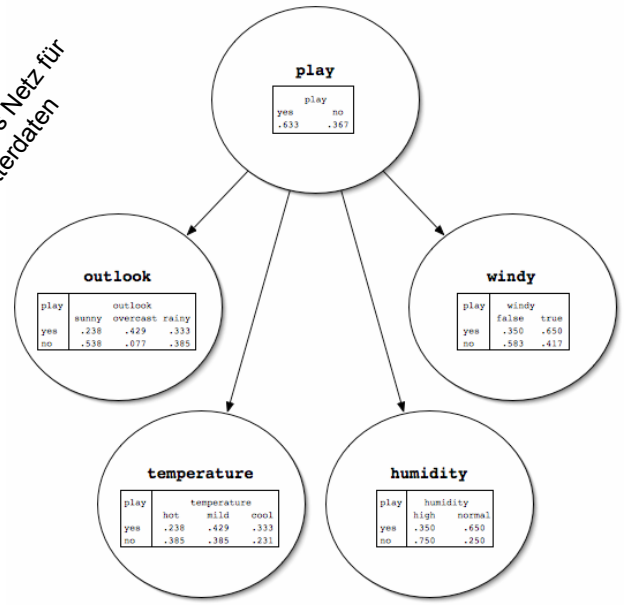
Naïve Bayes

- Annahme: Attribute sind bedingt unabhängig bei gegebener Klasse
- Annahme trift in der Praxis oft nicht zu aber die Methode hat oft eine hohe Genauigkeit
- Jedoch: manchmal ist die Performanz viel schlechter als z.B. bei Entscheidungsbäumen
- Kann die Annahme fallengelassen werden?

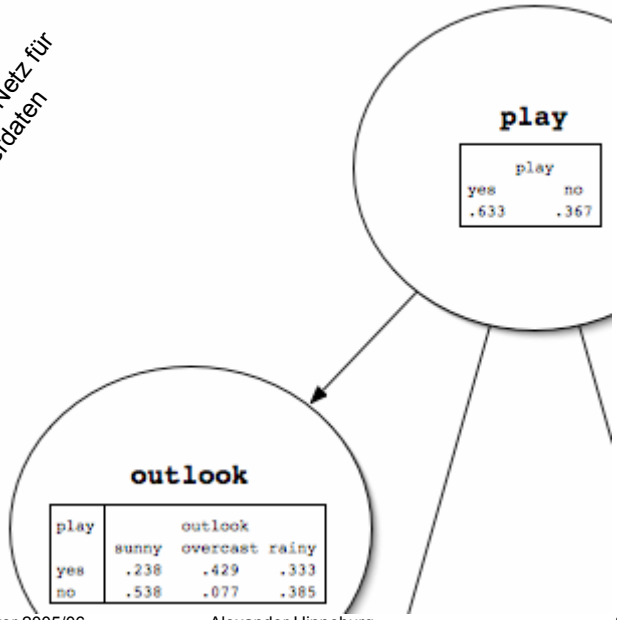
Bayesische Netzwerke

- Graphische Modelle, die jede Wahrscheinlichkeits-verteilung repräsentieren können
- Graphische Repräsentation: gerichteter azyklischer Graph, ein Knoten für jedes Attribut
- Gesamtwahrscheinlichkeitsverteilung wird in Komponentenverteilungen faktorisiert
- Die Knoten des Graphen beschreiben die Komponentenverteilungen (bedingte Verteilungen)

Bayesisches Netz für Wetterdaten



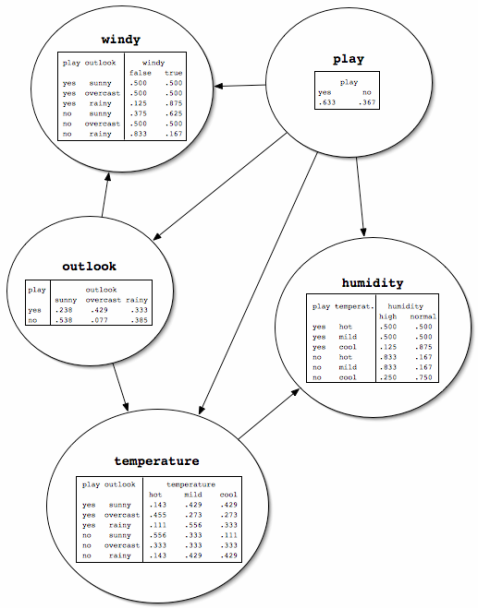
Bayesisches Netz für Wetterdaten



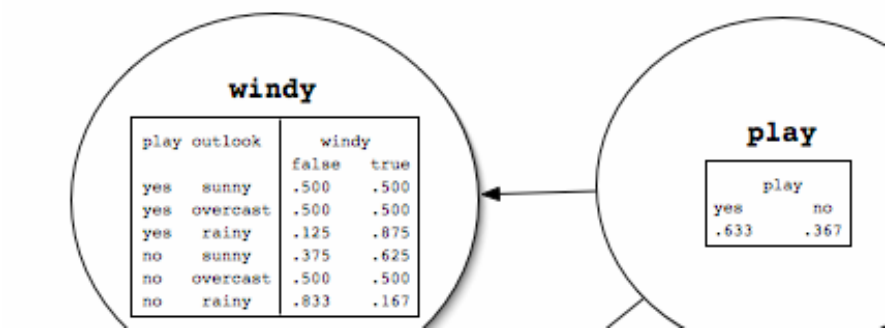
Wetterdaten

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

Bayesisches Netz für Wetterdaten



Bayesisches Netz für Wetterdaten



Wetterdaten

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

Berechnen der Klassenwahrscheinlichkeiten

- Zwei Schritte: (1) Berechnen einer Produktwahrscheinlichkeit für jede Klasse und (2) Normalisierung
- Gegeben eine neue Instanz
  - Für jede Klasse
    - Betrachte alle Attributwerte und und Klassenwerte
    - Schlage die korrespondierenden Einträge in den Tabellen mit den bedingten Whr. nach
    - Multipliziere alle nachgeschlagenen Whr.
  - Dividiere das Produkt für jede Klasse durch die Summe aller Klassenprodukt-Whr. (Normalisierung)

Warum ist das erlaubt? (Teil I)

- Annahme: Werte der Eltern eines Knoten bestimmen ausschließlich die Whr.-Verteilung des Knoten

$$Pr[\text{Knoten}|\text{Vorgänger}] = Pr[\text{Knoten}|\text{Eltern}]$$

- D.h., daß ein Knoten/Attribut bedingt unabhängig von allen anderen Knoten ist, außer von seinen Eltern.

Warum ist das erlaubt? (Teil II)

- Kettenregel der Wahrscheinlichkeits-rechnung:

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

- Wegen der Annahme auf der vorher-gehenden Folie:

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

Lernen von Bayesischen Netzen

- Basis Komponenten eines Algorithmus zum Lernen von Bayesischen Netzen:
  - Methode zum Evaluieren der Güte eines gegebenen Netzwerkes
    - Maß der Whr. der Trainingsdaten gegeben des Netzwerkes (oder der Log. davon)
  - Methode zum Durchsuchen des Raumes der möglichen Netzwerke
    - Entspricht dem Durchsuchen der möglichen Kantenmengen, da Knoten fest sind.

Problem: Overfitting

- Whr. der Trainingsdaten zu maximieren reicht nicht
  - Denn es ist immer besser mehr Kanten hinzuzufügen. Die Trainingsdaten werden so noch besser beschrieben.
- Nutze Kreuz-Validierung oder einen Strafterm für die Komplexität des Netzwerkes
  - AIC Maß: $-LL + K$
  - MDL Maß: $-LL + \frac{K}{2} \log N$
  - *LL:* log-likelihood (Log. der Whr. der Daten), *K*: Anzahl der freien Parameter, *N*: #Instanzen
- Andere Möglichkeit: Bayesischer Ansatz mit einer vorausgesetzen Verteilung über die möglichen Netzwerke

Suchen nach einen guten Struktur

- Aufgabe kann vereinfacht werden: jeder Knoten kann separat optimiert werden
  - da die Whr. einer Instanz das Produkt von einzelnen Knoten-Whr. ist
  - Funktioniert auch für AIC und MDL Kriterien, weil die Strafen sich aufaddieren
- Knoten kann optimiert werden durch Hinzufügen oder Löschen von Kanten zu anderen Knoten
- Darf keine Kreise erzeugen!

Der K2 Algorithmus

- Gegeben ist eine Reihenfolge der Knoten (Attribute)
- In jedem Schritt wird ein Knoten bearbeitet
- Versucht Kanten von vorhergehenden Knoten hinzuzufügen (Greedy Alg.)
- Geht zu nächsten Knoten, wenn der aktuelle Knoten nicht weiter verbessert werden kann
- Ergebnis ist abhängig von der gewählten initialen Reihenfolge

Einige Tricks

- Es kann hilfreich sein die Suche mit einem naïven Bayesischen Netzwerk zu beginnen
- Es kann hilfreich sein zu garantieren, daß jeder Knoten in der Markov Hülle eines Klassenknoten ist
  - Markov Hülle eines Knoten enthält alle Eltern, deren Kinder und die Eltern der Kinder
- Wenn die Whr. für die Knoten aus der Markov Hülle gegeben sind, ist der Knoten bedingt unabhängig von den Knoten außerhalb des Markov Hülle
  - d.h. eine Knoten ist irrelevant für die Klassifikation, wenn er nicht im Markov Hülle eines Klassenknotens ist

Andere Algorithmen

- K2 kann erweitert werden, um gieriges Hinzufügen oder Löschen von Kanten zu jedem Knotenpaar zu erlauben
  - Weitere Möglichkeiten: Invertierung der Richtung von Kanten erlauben
- TAN (Tree Augmented Naïve Bayes):
  - Beginne mit naïve Bayes
  - Erlaube ein zweites Elternteil zu einem Knoten hinzuzufügen (neben dem Klassenknoten)
  - Effizienter Algorithmus existiert

Likelihood vs. Bedingte Likelihood

- Bei Klassifikation soll eigentlich die Whr. einer Klasse bei gegeben Werten anderer Attribute maximiert werden
  - *Nicht* die Whr. der Instanzen
- Aber: es gibt keine geschlossene Lösung für Whr. in den Tabellen, welche dieses Kriterium maximieren würde
- Jedoch: Bedingte Whr. der Daten bei gegebenem Netzwerk kann leicht berechnet werden
- Scheint gut für der Bewerten eines Netzwerkes zu funktionieren

Diskussion

- Annahmen: diskrete Daten, keine fehlenden Werte, keine neuen Knoten
- Andere Methode um Bayesische Netze für Klassifikation zu nutzen: *Bayesische Multinetze*
  - Erzeuge ein Netzwerk für jede Klasse und kombiniere die Vorhersage mittels der Bayesischen Regel
- Andere Klasse von Lernmethoden: teste bedingte Unabhängigkeit