Bayes Klassifikatoren

METHODEN DES DATA MINING

FABIAN GREUEL

Inhalt

- Grundlagen aus der Wahrscheinlichkeitsrechnung
- Hypothesenwahl
- Optimale Bayes Klassifikator
- Naiver Bayes Klassifikator
- Beispiel Textmining
- Ansatzder Bayes-Netzte

Grundlagen

- P(A) : Wahrscheinlichkeit für ein Ereignis
- P(A,B) : Wahrscheinlichkeit für ein Ereignis A und B
- Bedingte Wahrscheinlichkeit:

$$P(A|B) = \frac{P(AB)}{P(B)}$$

- Satz der totalen Wahrscheinlichkeit:

$$P(A) = \sum_{i \in I} P(AB_i) = \sum_{i \in I} P(B_i)P(A|B_i)$$

Das Theorem von Bayes

- Da P(A,B) = P(A|B)P(B) = P(B|A)P(A) gilt…

$$P(H|B) = \frac{P(H,B)}{P(B)} = \frac{P(B|H)P(H)}{P(B)}$$

A-Posterior-Wahrscheinlichkeit

Likelihood

A-Priori-Wahrscheinlichkeit

Thomas Bayes *1702 †1761
Quelle:
http://www.wikipedia.de

Von Wahrscheinlichkeiten zu Hypothesen

$$\max_H P(H|B) \equiv \frac{P(H,B)}{P(B)} \equiv \frac{P(B|H)P(H)}{P(B)}$$

- Beim Bayesschen Lernen wird eine Hypothese mit maximaler A-Posterior-Wahrscheinlichkeit (kurz eine MAP – Hypothese) gesucht.
- Der Term P(B) kann hierbei vernachlässigt werden
  - Normalisierung

Ein fiktives Beispiel (I)

- 0.6 % der Bevölkerung hat Krebs, notiert durch $P(K)=0.006$ und $P(\neg K)=0.994$
- Ein Test erkennt Krebs mit 98% Genauigkeit, also… $P(+|K)=0.98$ und $P(-|K) = 0.02$
- Ein negatives Ergebnis ist nur in 96% richtig, also… $P(-|\neg K)=0.96$ und $P(+|\neg K) = 0.04$

Ein fiktives Beispiel (II)

- Fragestellung: Welche Hypothese ist bei einem positiven Test wahrscheinlicher?
  - Der Patient hat Krebs:

$$P(+|K)P(K) = 0.98 * 0.006 = \ 0.00588$$

Ein fiktives Beispiel (II)

- Fragestellung: Welche Hypothese ist bei einem positiven Test wahrscheinlicher?
  - Der Patient hat Krebs:

$$P(+|K)P(K) = 0.98 * 0.006 = 0.00588$$

  - Der Patient hat keinen Krebs

$$P(+|\neg K)P(\neg K) = 0.04 * 0.994 = \underline{0.03976}$$

MAP-Hypothese!

Maximum A Posteriori

Ein fiktives Beispiel (III)

- Die Bayes-Umkehrformel

$$P(B_k|A) = \frac{P(A|B_k)P(B_k)}{\sum_{i=1}^{n} P(A|B_i)P(B_i)}$$

- Normalisierung

$$P(K|+) = \frac{0.00588}{(0.00588 + 0.03976)} = 12.88\%$$

$$P(\neg K|+) = \frac{0.03976}{(0.00588 + 0.03976)} = 87.11\%$$

Ein Brute-Force-Algorithmus

- Berechne für jede mögliche Hypothese die A-Posteriori-Wahrscheinlichkeit

$$P(H|B) = \frac{P(B|H)P(H)}{P(B)}$$

- Wähle die Hypothese mit der größten Wahrscheinlichkeit

$$\max_H P(H|B) = \frac{P(B|H)P(H)}{P(B)}$$

Optimaler Bayes Klassifikator (I)

- Fragestellung: Beste Klassifizierung

Optimaler Bayes Klassifikator (I)

- Fragestellung: Beste Klassifizierung
- Idee: Gewichtung der Vorhersagen v_J von Hypothese h_J mit der Posterior-Wahrscheinlichkeit auf Basis der Daten *D*.

Optimaler Bayes Klassifikator (I)

- Fragestellung: Beste Klassifizierung
- Idee: Gewichtung der Vorhersagen v_J von Hypothese h_J mit der Posterior-Wahrscheinlichkeit auf Basis der Daten *D*.
- Beispiel: $P(h_1|D) = 0.4 \quad P(h_2|D) = 0.3 \quad P(h_3|D) = 0.3$

- Die Hypothesen klassifizieren wie folgt:

$$P(+|h_1) = 1 \quad P(-|h_2) = 1 \quad P(-|h_3) = 1$$

MAP-Hypothese

Optimaler Bayes Klassifikator (II)

- Fragestellung: Optimale Klassifizierung
- Idee: Gewichtung der Vorhersagen v_J von Hypothese h_J mit der Posterior-Wahrscheinlichkeit auf Basis der Daten *D*.

$$v_{BOC} = \arg\max_{v_j \in V} P(v_j|D) = \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

- Für das Beispiel:

$$P(+) = 1 * 0.4 = 0.4$$

$$P(-) = 1 * 0.3 + 1 * 0.3 = \boxed{0.6}$$

Optimaler Bayes Klassifikator (III)

- Der BOC liefert im Durchschnitt das optimalste Ergebnis, aber…
- … ist sehr rechenaufwendig
- Variante: Der Gibbs-Algorithmus
  - Wähle zufällig eine Hypothese h, unter Beachtung der Posterior-Wahrscheinlichkeitsverteilung über H
  - Benutze h um eine Klassifizierung eines Exemplars x.
  - $\text{Fehlerrate}_{\text{Gibbs}} <= 2 * \text{Fehlerrate}_{\text{BOC}}$

Naiver Bayes Klassifikator (I)

- Fragestellung: Optimale Klassifizierung
- Attribute: $< a_1, a_2, \dots, a\_n >$
- Klassifizierungen: V

$$v_{MAP} = \arg\max_{v \in V} P(v_j | a_1, a_2, \dots, a_n)$$

$$v_{MAP} = \arg\max_{v \in V} \frac{P(a_1, a_2, \dots, a_n | v_j) P(v_j)}{P(a_1, a_2, \dots, a_n)}$$

$$v_{MAP} = \arg\max_{v \in V} P(a_1, a_2, \dots, a_n | v_j) P(v_j)$$

Naiver Bayes Klassifikator(II)

- Attribute: $< a_1, a_2, \ldots, a\_n >$
- Klassifizierungen: V

$$v_{MAP} = \arg\max_{v \in V} P(a_1, a_2, \ldots, a_n | v_j) P(v_j)$$

- Annahme: Alle Attribute sich stoch. unabhängig:
  - Komplexität

$$P(a_1, a_2, \ldots, a_n | v_j) P(v_j) = P(v_j) \prod_i P(a_i | v_j)$$

- Naiver Bayes Klassifikator:

$$v_{NB} = \arg\max_{v \in V} P(v_j) \prod_i P(a_i | v_j)$$

Beispiel(I)

Tag	Temperatur	Luft-Feuchte	Wind	Aussicht	Tennis spielen?
1	Warm	Hoch	Schwach	Sonnig	Nein
2	Warm	Hoch	Stark	Sonnig	Nein
3	Warm	Hoch	Schwach	Bewölkt	Ja
4	Milde	Hoch	Schwach	Regen	Ja
5	Kalt	Normal	Schwach	Regen	Ja
6	Kalt	Normal	Stark	Regen	Nein
7	Kalt	Normal	Stark	Bewölkt	Ja
8	Milde	Hoch	Schwach	Sonnig	Nein
9	Kalt	Normal	Schwach	Sonnig	Ja
10	Milde	Normal	Schwach	Regen	Ja
11	Milde	Normal	Stark	Sonnig	Ja
12	Milde	Hoch	Stark	Bewölkt	Ja
13	Warm	Normal	Schwach	Bewölkt	Ja
14	Milde	Hoch	Stark	Regen	Nein

Beispiel(II)

- Das neue Exemplar (Kalt, Hoch, Stark, Sonnig) soll klassifiziert werden.

$$v_{NB} = \arg\max_{v \in V} P(v_j) \prod_i P(a_i | v_j)$$

$$v_{NB} = \arg\max_{v \in \{ja, nein\}} P(v_j) P(Aussicht = sonnig)\, P(Temperatur = kalt) * P(Luftfeuchte = hoch) P(Wind = stark)$$

Beispiel(III)

Tag	Tennis spielen?
1	Nein
2	Nein
3	Ja
4	Ja
5	Ja
6	Nein
7	Ja
8	Nein
9	Ja
10	Ja
11	Ja
12	Ja
13	Ja
14	Nein

- P(ja) = 9 / 15
- P(nein) = 6 / 15

Beispiel(IV)

Tag	Temperatur	Luft-Feuchte	Wind	Aussicht	Tennis spielen?
1	Warm	Hoch	Schwach	Sonnig	Nein
2	Warm	Hoch	Stark	Sonnig	Nein
3	Warm	Hoch	Schwach	Bewölkt	Ja
4	Milde	Hoch	Schwach	Regen	Ja
5	Kalt	Normal	Schwach	Regen	Ja
6	Kalt	Normal	Stark	Regen	Nein
7	Kalt	Normal	Stark	Bewölkt	Ja
8	Milde	Hoch	Schwach	Sonnig	Nein
9	Kalt	Normal	Schwach	Sonnig	Ja
10	Milde	Normal	Schwach	Regen	Ja
11	Milde	Normal	Stark	Sonnig	Ja
12	Milde	Hoch	Stark	Bewölkt	Ja
13	Warm	Normal	Schwach	Bewölkt	Ja
14	Milde	Hoch	Stark	Regen	Nein

Beispiel(V)

- Abschätzung bedingter Wahrscheinlichkeiten

$$P(Wind = stark|Tennisspielen = nein) = \frac{3}{5}$$

$$P(Wind = stark|Tennisspielen = ja) = \frac{3}{9}$$

- P(ja)P(sonnig|ja)P(kalt|ja)P(hoch|ja)P(stark|ja) = 0.0053
- P(nein)P(sonnig|nein)P(kalt|nein)P(hoch|nein) * P(stark|nein) = 0.0206

Beispiel(VI)

- P(ja)P(sonnig|ja)P(kalt|ja)P(hoch|ja)P(stark|ja) = 0.0053
- P(nein)P(sonnig|nein)P(kalt|nein)P(hoch|nein) * P(stark|nein) = 0.0206
- Normalisiert:

$$\frac{0.0053}{0.0053 + 0.0206} = 20.5\%$$

$$\frac{0.206}{0.0053 + 0.0206} = 79.5\%$$

Probleme

- Bei einer geringe bedingten Wahrscheinlichkeit…
  - (1) $P(Wind = stark \mid Tennisspielen = nein) = \frac{n_c}{n} = 0.08$
- … und einer geringe Anzahl von Datensätzen
  - z. B. n=5 dann folgt $n_c = 0.4$
- …produziert $\frac{n_c}{n}$ tendenziell einen zu niedrigen Schätzwert.
- (1) Dominiert die anderen Terme bei der Berechnung

m-Schätzer

- Idee: Gewichtung des Schätzwertes mit einer angenommenen Verteilung p

$$\frac{n_{c+mp}}{n+m}$$

  - m lässt sich als die Anzahl zusätzlicher Datensätze interpretieren

m-Schätzer

- Idee: Gewichtung des Schätzwertes mit einer angenommenen Verteilung p

$$\frac{n_{c+mp}}{n+m}$$

  - m lässt sich als die Anzahl zusätzlicher Datensätze interpretieren

- Standard-Methode:
  - p ist eine Gleichverteilung über die Anzahl der möglichen Werte m

Anwendungsgebiet Text-Mining

- Ziel
  - Identifizierung von relevanten Inhalten
  - Klassifikation von Texten (z.B. Web-Seiten)
- Repräsentierung
  - Dokument als Vektor mit einem Attribut je Wortposition a_i
  - Vorgegebenes Vokabular mit Wörtern w_i
- Auch in dieser Anwendung gilt die stochastische Unabhängigkeitsannahme! …
- … und $\forall i \forall m: P(a_i = w_k | v_j) = P(a_j = w_k | v_j)$

Algorithmus - Lernphase

- *Vokabular* besteht aus allen Wörtern, die in Beispieltexten vorkommen.
- Für jeden Klassifikation v_J
  - $P(v_j) = \frac{dok_j}{|Dok|}$
  - $Text_J$ = Konkatenation aller dok_J
  - n = gesamte Anzahl der Worte in $Text_J$
  - für jedes Wort in w_k im *Vokabular*
    - n_k = Anzahl des Auftretens von Wort w_k
    - $P(w_k|v_j) = \frac{n_k + 1}{n + |Vokabular|}$

Algorithmus - Klassifikation

- Ein Dokument wird auf Basis des Vokabulars klassifiziert, andere Wörter werden ignoriert.
- *Pos* sei die Menge aller Wortposition mit einem Wert aus dem *Vokabular*
- Der Algorithmus liefert:

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_{i \in Pos} P(a_i | v_j)$$

Algorithmus - Betrachtungen

- Komplexität
  - |V={comp.graphics, … sci.med}|= 20
  - |Vokabular| ≈ 38500

$$\left|P\left(w_k\middle|v_j\right)\right| = |V| * |Vokabular|$$

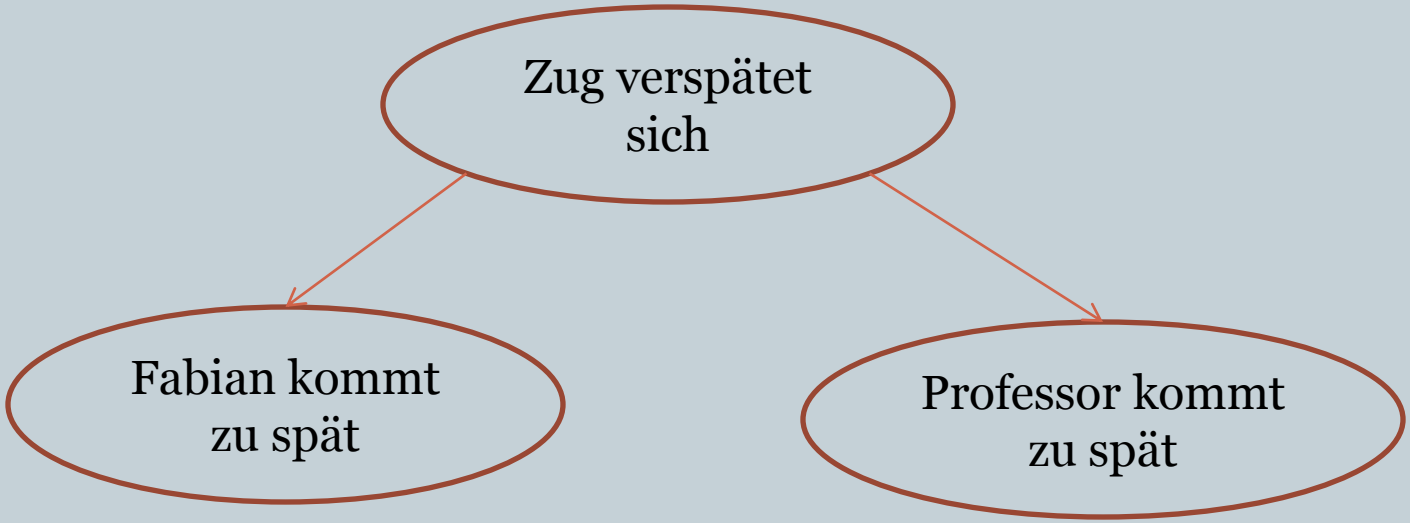
- Experiment
  - Jeweils 1000 Dokumente aus einer Newsgroups
  - 2/3 als Trainingsdaten, 1/3 als zu klassifizierende Daten
  - Die 100 häufigsten Wörter wurden aus dem Vokabular entfernt.
- Resultate
  - Die erreichte Genauigkeit lag bei 89%.
  - Die Performe leidet nicht unter den Unabhängigkeitsannahmen

Bemerkungen zum naiven Bayes Klassifikator

- Beste Resultate bei stoch. Unabhängigkeit der Attribute
  - Diagnosen
  - Spam-Filterung
- Kann problemlos mit fehlenden Attributwerten umgehen
- Einfach zu programmieren.

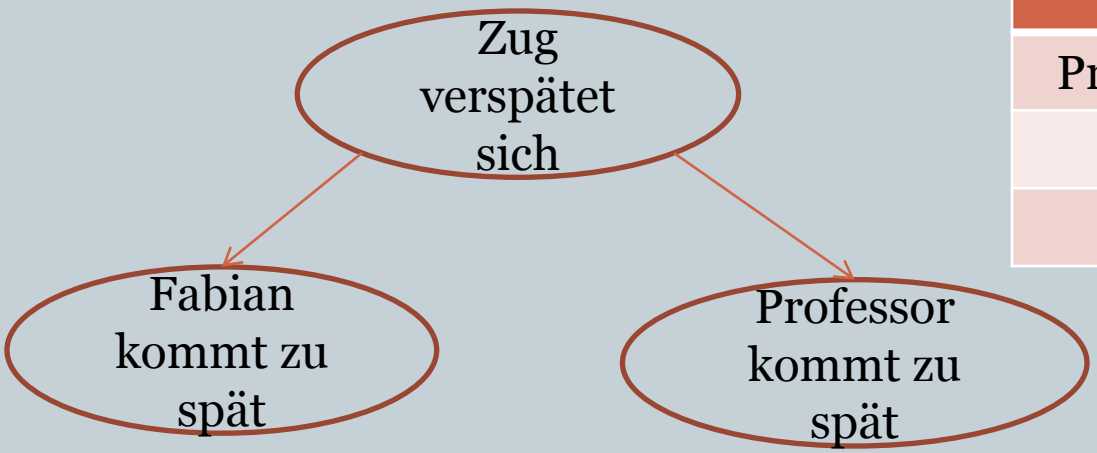
Bayessche Netzwerke

- Repräsentieren eine Verbundwahrscheinlichkeit
- Modelliert Knoten repräsentieren Zufallsvariablen
- Gerichte Kanten modellieren Abhängigkeiten



Bayessche Netzwerke

- Jeder Knoten enthält eine Wahrscheinlichkeitstabelle



	Zug verspätet sich	
Fabian spät	True	False
True	0.8	0.1
False	0.2	0.9

	Zug verspätet sich	
Prof spät	True	False
True	0.8	0.1
False	0.2	0.9

Zug verspätet sich	
True	False
0.1	0.9

Bayessche Netzwerke

- Die Wahrscheinlichkeit, dass Fabian zu spät kommt:

$$P(F\ spät) = P(F\ spät|Zug\ spät)P(Zug\ spät) +$$
$$P(F\ spät|Zug\ nicht\ spät)P(Zug\ nicht\ spät)$$
$$P(F\ spät) = 0.8 * 0.1 + 0.1 * 0.9$$

Bayessche Netzwerke

- Ausbreitung
  - Fakten verändern Wahrscheinlicheiten
  - Beispiel: Fabian kommt zu spät.

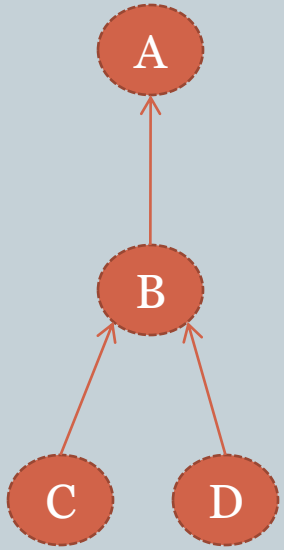
$$P(Zug\ spät|F\ spät) = \frac{P(F\ spät|Zug\ spät)P(Zug\ spät)}{P(F\ spät)}$$

$$P(Zug\ spät|F\ spät) = \frac{0.8 * 0.1}{0.17} = 0.47$$

  - Dieser Wert wird in den Knoten „Zug verspätet sich eingetragen und beinflusst dadurch P(Professor spät|Zug spät)

Bayessche Netzwerke

- Statt
  - $P(A,B,C,D)=P(A|B,C,D)P(B|C,D)P(C|D)P(D)$
- benötigt man nur
  - $P(A,B,C,D)=P(A|B)P(B|C,D)P(C)P(D)$
- Aber …. das Lernen liegt in der Klasse NP
  - Näherungsverfahren
  - Gradientenabstiegsverfahren



Quellenangaben

- Tom Mitchel – Machine Learning (1997)
- Richard O. Duda – Pattern Classification (2001)