# Pairwise Naive Bayes Classifier

Jan-Nikolas Sulzmann[1]

[1] nik.sulzmann@gmx.de
Fachbereich Knowledge Engineering
Technische Universität Darmstadt

# Gliederung

# Kurzfassung

## Ziel

Verbesserung der Performanz des Naive Bayes Klassifizierers durch Methoden der paarweisen Klassenbinarisierung

## Verwendete Methoden

- Round Robin Klassenbinarisierung
- Alternative Methoden
- Ensemble-Methoden: Bagging und Boosting

# Kurzbeschreibung

## Naive Bayes Klassifizierer

- Bayes'sche Lernverfahren
  - d.h. auf dem Satz von Bayes basierend
  - Fundierte wahrscheinlichkeitstheoretische Grundlage
  - Vorhersage der wahrscheinlichsten Klasse
- Effizient
  - Zum Teil bessere Performanz als aktuelle Lernverfahren, z.B. Neuronale Netze, Nearest Neighbour oder Entscheidungsbäume
- Einfach implementierbar
- Flexibel
  - Anwendbar auf strukturierte (z.B. Tabellen) und unstrukturierte Daten (z.B. Texte, Web-Dokumente)

# Satz von Bayes

## Satz von Bayes

Für zwei Zufallsereignisse A und B gilt

$$\Pr(A|B) = \frac{\Pr(B|A)\Pr(A)}{\Pr(B)}$$

## Bedingte Wahrscheinlichkeit

- Für zwei Zufallsereignisse A und B gilt

$$\Pr(A|B) = \frac{\Pr(A \wedge B)}{\Pr(B)}$$

- Die Wahrscheinlichkeit, daß ein Zufallsereignis A auftritt, wenn wir bereits das Zufallsereignis B beobachtet haben

# Naive Bayes Klassifizierer

## Ziel

Bestimmung der wahrscheinlichsten Klasse $c$ unter Beobachtung des Beispieles $D$ anhand der Trainingsdaten

## Gegeben

- Menge von Klassen
- Trainingsbeispiele
- Beispiele durch Attributwerte beschrieben
- $a_i$ Wert des Attributes $A_i$
- Beispiel $D = (a_1, a_2, ..., a_n)$

## Vorläufige Version

$$\begin{aligned} & \arg\max_c \Pr(c|D) \\ = \; & \arg\max_c \frac{\Pr(D|c)\Pr(c)}{\Pr(D)} \\ = \; & \arg\max_c \Pr(D|c)\Pr(c) \end{aligned}$$

# Unnabhängigkeitsannahme

## Ziel

Bestimmung von $\Pr(D|c) = \Pr(a_1, a_2, ..., a_n|c)$ anhand der Trainingsbeispiele

## Problem

Relative Häufigkeit der Beispiele $D = (a_1, a_2, ..., a_n, c)$ bezüglich Klasse c ist eine schlechte Abschätzung, da diese Beispiele selten oder gar nicht vorkommen

## Lösung

- naive Annahme über Unabhängigkeit der Attribute
- $\Pr(a_1, a_2, ..., a_n|c) = \prod_{i=1}^{n} \Pr(a_i|c)$

# Abschätzung der Wahrscheinlichkeiten

## Ziel

Abschätzung der Wahrscheinlichkeiten $\Pr(a|c)$ und $\Pr(c)$

## Abschätzung von $\Pr(a|c)$

- $\Pr(a|c) = \frac{n_{a \wedge c}}{n_c}$
- $n_{a \wedge c}$: absolute Häufigkeit von Beispielen mit Attributwert a und Klasse c
- $n_c$: absolute Häufigkeit von Klasse c
- Problem:

$$n_{a \wedge c} = 0 \Rightarrow \Pr(c|D) = 0$$

## Lösung für $\Pr(a|c)$

- Laplace-Abschätzung: $\Pr(a|c) = \frac{n_{a \wedge c}+1}{n_c+v}$
- v: Anzahl unterschiedlicher Attributwerte

## Abschätzung von $\Pr(c)$

- $\Pr(c) = \frac{n_c}{n}$
- n: Anzahl von Trainingsbeispielen

# Illustratives Beispiel

**$D = (Warm, Regen)$,Golf?**

$$\Pr(+) = 2/5,\ \Pr(-) = 3/5$$

$$\Pr(W|+)\Pr(R|+) = \frac{3}{5} * \frac{1}{5} = \frac{3}{25}$$

$$\Pr(W|-)\Pr(R|-) = \frac{1}{6} * \frac{2}{6} = \frac{1}{18}$$

$$\frac{3}{25} * \frac{2}{5} = \frac{6}{150}, \frac{1}{18} * \frac{3}{5} = \frac{5}{150}$$

$$\Pr(+|W,R) = 6/11$$

$$\Pr(-|W,R) = 5/11$$

Abk.: W = Warm, R = Regen

| Temp. | Aussicht | Golf? |
|---|---|---|
| Warm | Sonnig | ja |
| Warm | Bewölkt | ja |
| Mild | Bewölkt | nein |
| Kalt | Regen | nein |
| Kalt | Bewölkt | nein |

| | ja | nein |
|---|---|---|
| Kalt | 0 | 2 |
| Mild | 0 | 1 |
| Warm | 2 | 0 |
| Bewölkt | 1 | 2 |
| Regen | 0 | 1 |
| Sonnig | 1 | 0 |

# Grundidee

## Problem

- Reale Klassifikationsrobleme sind häufig Multiklassenprobleme
- Aber viele Klassifizierer sind inhärent binär
- d.h sie können nur mit binären Probleme umgehen
- Bekannte Vertreter sind z.B. Perceptrons und SVM

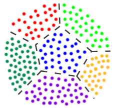
## Lösung

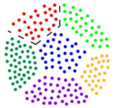
- Zerlegung des Multiklassenproblem in mehrere binäre Probleme
- Decodierung der Vorhersagen der binären Klassifizierer
- Kann auch bei nichtbinären Klassifizierer zu einer Verbesserung führen

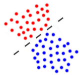
# Typen der Klassenbinarisierung

## Behandlung eines Multiklassenproblem

- Ungeordnete Klassenbinarisierung
  - Transformation in $m$ binäre Probleme
  - Jeweils eine Klasse gegen die anderen
  - Benötigt Decodierung der Vorhersagen
- Geordnete Klassenbinarisierung
  - Wie ung. KB mit sukzessivem Ausschluß
  - Benötigt keine Decodierung
- Round Robin Klassenbinarisierung
  - Transformation in $m(m-1)/2$ binäre Probleme
  - Je Klassenpaar ein Klassifikator
  - Benötigt Decodierung der Vorhersagen







# Voting & Weighted Voting

## Voting

- Jeder Klassifizierer stimmt für eine der Klassen
- Vorhersage der Klasse mit den meisten Stimmen
- z.B. Bagging

## Weighted Voting

- Berechne gewichtete Summe der Stimmen für jeden Klassifizierer
- Gewichte normalerweise abhängig von
  - dem Vertrauen des Klassifizierers auf seine Vorhersage (z.B. geschätzte Wahrscheinlichkeit einer Klasse)
  - der Fehlerabschätzung des Klassifizierers (z.B. Boosting)

# Round Robin Klassenbinarisierung mit Naive Bayes

## Voting

$\arg\max_{c_i} \sum_{j\neq i} [\Pr(c_i|D, c_{ij})]$

## Weighted Voting

$\arg\max_{c_i} \sum_{j\neq i} \Pr(c_i|D, c_{ij})$

## Ergebnisse

- RR mit Voting ist äquivalent zu RR mit Weighted Voting, da

$$[\Pr(c_i|D, c_{ij})] \leq [\Pr(c_j|D, c_{ij})] \Leftrightarrow \Pr(c_i|D, c_{ij}) \leq \Pr(c_j|D, c_{ij})$$

- RR mit Weighted Voting ist äquvialent zu regulärem NB, da

$$\frac{\Pr(c_i|D, c_{ij})}{\Pr(c_j|D, c_{ij}} = \frac{\Pr(c_i|D)}{\Pr(c_j|D)}$$

# Wahrscheinlichkeitstheoretischer Ansatz

## Ansatz

$$\Pr(c_i|D) = \frac{1}{(m-1)} \sum_{j \neq i} \Pr(c_i|D, c_{ij}) \Pr(c_{ij}|D)$$

- $v_{ij} = \Pr(c_i|D, c_{ij}) = \frac{\Pr(c_i|D)}{\Pr(c_i|D)+\Pr(c_j|D)}$
- Für $w_{ij} = w_{ji} = \Pr(c_{ij}|D)$ gibt es zwei Möglichkeiten

## Reguläre Berechnung von $w_{ij}$

- Reguläres „Abzählen“
- Häufigkeitstabelle
  - 1 Zelle je Kombination v. Attributwert und Klasse

## Paarweise Berechnung von $w_{ij}$

- „Abzählen“ für Klassenpaare
- Häufigkeitstabelle
  - 1 Zelle je Kombination v. Attributwert und Klassenpaar

# Resultierende Methoden

## Anwendungsmöglichkeiten

- $v_{ij}$ verwendbar zur (un-)gewichteten Abstimmung
- $w_{ij} = w_{ji}$ verwendbar als Gewicht

## PNB1

$$\arg\max_i \sum_{\substack{j \neq i \\ v_{ij} \geq v_{ji}}} w_{ij}$$

## PNB2

$$\arg\max_i \sum_{j \neq i} v_{ij} * w_{ij}$$

## PNB3

$$\arg\max_i \sum_{\substack{j \neq i \\ \Pr(c_i) \geq \Pr(c_j)}} w_{ij}$$

## PNB4

$$\arg\max_i \sum_{j \neq i} \frac{\Pr(c_i)}{\Pr(c_i) + \Pr(c_j)} w_{ij}$$

# Vergleich mit Naive Bayes: Ergebnisse

Accuracy (Win/Draw/Loss)

| Methode | Regulär | Paarweise |
|---|---|---|
| PNB1 | 83,34 (0/36/0) | 83,27 (4/24/8) |
| PNB2 | 83,34 (0/36/0) | 83,29 (7/19/10) |
| PNB3 | 71,47 (3/3/30) | 70,17 (3/0/33) |
| PNB4 | 73,49 (4/4/28) | 72,45 (4/0/32) |

- Reguläre PNB1 und PNB2 äquivalent zu NB
- Paarweise PNB1 und PNB2 schlechter als NB
- PNB3 und PNB4 signifikant schlechter (99%)

# Decodierungsmethoden

## Idee

- Schätze über Kenntnis der $\Pr(c_i|D, c_{ij})$ alle $\Pr(c_i|D)$ ab
  - : Idee: $\frac{\Pr(c_i|D,c_{ij})}{\Pr(c_j|D,c_{ij})} \overset{!}{\approx} \frac{\Pr(c_i|D)}{\Pr(c_j|D)}$
- Problem: für NB gilt $\frac{\Pr(c_i|D,c_{ij})}{\Pr(c_j|D,c_{ij})} \overset{!}{=} \frac{\Pr(c_i|D)}{\Pr(c_j|D)}$
- $\Rightarrow$ Als Decodierungsmethode für reguläres RR mit NB nicht geeignet, da Resultat äquivalent zu NB

## Methoden

- M. von Price, Kner, Personnaz und Dreyfus
  - $\Pr(c_i|D) = (\sum_{j \neq i} \frac{1}{\Pr(c_i|D,c_{ij})} - (\#Klassen - 2))^{-1}$
- M. von Refregier und Vallet: Lösung eines LGS
- M. von Hastie und Tibshirani: Konvergenter Algorithmus

# Ensemble-Methoden

## Idee

- Lerne anstatt einzelnen Klassifizierer eine Menge von Klassifizierern
- Bilde ein Ensemble unterschiedlicher Klassifizierer anhand derselben Trainingsmenge
- Kombiniere die Vorhersagen dieser Klassifierer (Decodierung)

## Motivation

- Reduziert Varianz: Ergebnisse sind weniger abhängig von den Eigenheiten einer einzelnen Trainingsmenge
- Reduziert Bias: eine Kombination von mehreren Klassifizierer kann ein aussagekräftigeres Konzept darstellen ein einzelner Klassifizierer

# Bagging & Boosting

## Idee

- Veränderung der Trainingsmenge, um unterschiedliche Klassifizierer zu erhalten
- Kombination der Vorhersagen durch (Weighted) Voting zur endgültigen Vorhersage
- Unser Ansatz: Anwendung auf Klassenpaare $c_{ij}$

## Bagging

- Parallele Berechnung
- Resampling
- Rauschen kein Problem
- Voting

## Boosting

- Sequentielle Berechnung
- Reweighting
- Rauschen problematisch
- Weighted Voting

# Vergleich von NB, Bagging und Boosting

| (W/D/L) | Bagging | AdabostM1 | Naive Bayes |
|---|---|---|---|
| Bagging | - | (11/0/25) | (16/0/20) |
| AdaboostM1 | (25/0/11) | - | (24/0/12) |
| Naive Bayes | (20/0/16) | (12/0/24) | - |

- Verwenden für Boosting für AdaboostM1
- Bagging schlechter als Naive Bayes
- AdaboostM1 signifikant besser als Naive Bayes und Bagging (95%)

# AdaBoostM1 & Bagging: Regulär & Alternativ

Vergleich: alternativ decodierte Ensembles mit regulären Ensembles

| (W/D/L) | HT | PKPD | RV | Voting |
|---|---|---|---|---|
| AdaboostM1 | (12/16/5) | (13/16/6) | (1/16/14) | (13/16/6) |
| Bagging | (7/14/11) | (7/19/9) | (6/16/10) | (7/19/9) |

- Vergleich der Decodierungsmethoden
  - AdaboostM1: HT, PKPD und Voting signifikant besser, RV signifikant schlechter
  - Bagging: alle schlechter
- Vergleich mit Naive Bayes
  - AdaboostM1: HT, PKPD und Voting signifikant besser
  - Restliche Methoden (signifikant) schlechter

# Noch zu tun

- Bagging & Boosting der alternativen Methoden
- Vergleich mit dem Bagging/Boosting der Round Robin Klassenbinarisierung
- Beide Fälle noch mit Weighted Voting testen