AODE and Comparison

Naive Bayes: Weakening the Independence Assumption

Seminar Maschinelles Lernen

Steffen Meyer

11.01.2006

Agenda

- Motivation
- Vergleichsalgorithmen
  - Vorstellung von NB, LBR, TAN und deren Komplexitäten
  - Vergleich TAN & LBR
- AODE: Averaged One-Dependence Estimators
  - Intuition
  - Herleitung des Klassifizierers
  - Komplexitäten
  - Vorteile, Erwartungen, Evaluation
  - Ergebnisse
- Fazit

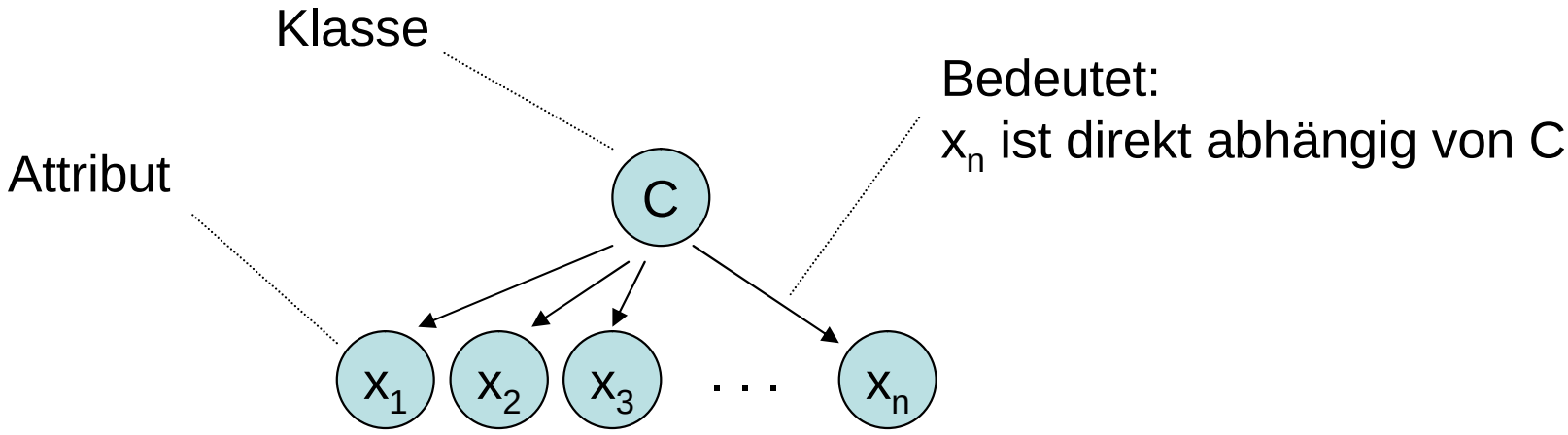
Motivation

- Naive Bayes
  - einfacher, effizienter, weit verbreiteter Klassifizierer
  - Annahme: Attribute sind unabhängig
  - geringe Verletzung à keine Auswirkung
  - Hohe Verletzung, d.h. Vernachlässigung vieler Abhängigkeiten zwischen den Attributen à hohe Fehlerquote
- Mehrere Ansätze existieren, um Unabhängigkeitsannahme abzuschwächen

- Gibt es Unterschiede zwischen diesen? Z. B. zwischen LBR und TAN?
- Lässt sich evtl. bei vergleichbaren Ergebnissen die Laufzeit verringern? Was bedeutet eigentlich AODE?

Vergleichsalgorithmen

- Besonders niedrige Fehlerquoten bei LBR & TAN
- Darstellung der Abhängigkeiten über Bayes‘sche Netzwerke



Bayes‘sches Netzwerk für Naive Bayes

Vergleichsalgorithmen

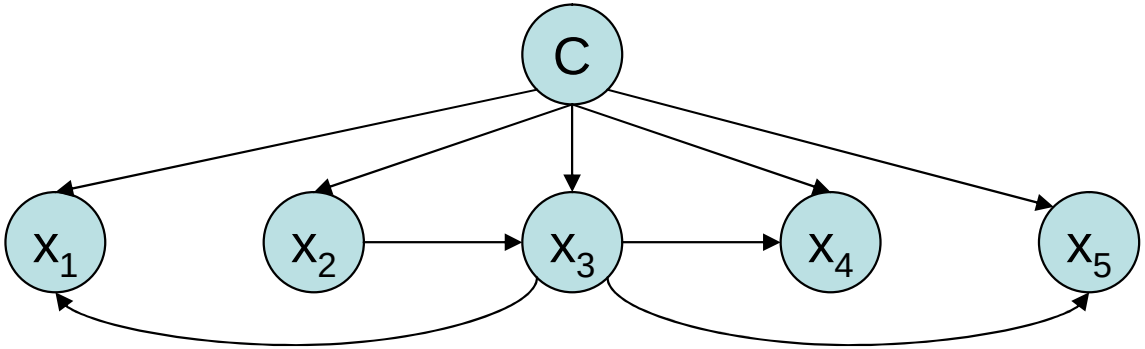
- Naive Bayes
  - Klassifiziert mit $\arg\max_{y}(\hat{P}(y)\prod_{i=1}^{n}\hat{P}(x_i \mid y))$
  - Berechnungskomplexität im Training:
    - Tabelle mit Klassenwahrscheinlichkeiten (eindimensional mit Klasse als Index) zur Berechnung des Schätzers *^P(y)*
    - Tabelle zur Berechnung der bedingten Wahrscheinlichkeiten der Attributwerte (zweidimensional mit Klasse und Attributwert als Index)
    - Datenkomplexität *O(knv); k=Klassen, n=Attribute, v=durchschnittliche Anzahl an Werten pro Attribut*
    - Zeitkomplexität: *O(tn); t=Anzahl der Trainingsbeispiele*
  - Berechnungskomplexität bei Klassifizierung:
    - Datenkomplexität: Verwendung der Trainingstabellen *O(knv)*
    - Zeitkomplexität: *O(kn)*

Vergleichsalgorithmen

- TAN (Tree Augmented Naive Bayes) & Super Parent-TAN
  - Jedes Attribut darf von der Klasse und höchstens einem anderen Attribut abhängig sein, seinem *parent*
  - Klassifizierung durch: $\arg\max_y \left(\hat{P}(y) \prod_{i=1}^{n} \hat{P}(x_i \mid y, parent(x_i) \right)$
  - Berechnungskomplexität im Training:
    - Datenkomplexität: Berechnung einer 3D Wahrscheinlichkeitstabelle für jeden Attributwert, unter der Bedingung jedes anderen Attributwertes und jeder Klasse; $O(k(nv)^2)$; SP-TAN muss zusätzlich noch die Trainingsdaten speichern mit $O(tn)$
    - Zeitkomplexität: Berechung der Wahrscheinlichkeitstabelle $O(tn^2)$; Berechnung der parent Funktion $O(kn^2v^2+n^2 log\ n)$
  - Berechnungskomplexität bei Klassifizierung:
    - Daten: Komprimierte Wahrscheinlichkeitstabelle; $O(knv^2)$
    - Zeit: Klassifizierung; $O(kn)$

Vergleichsalgorithmen

- Bayes‘sches Netzwerk für TAN



- Gründe für die Restriktionen
  - Reduzierung der Anzahl von möglichen Klassifizierern
  - Werden zusätzliche *parents* erlaubt, so werden die Wahrscheinlichkeitsschätzungen unzuverlässiger, da die Tabellen mit bedingten Wahrscheinlichkeiten exponentiell mit der Anzahl der Eltern wachsen

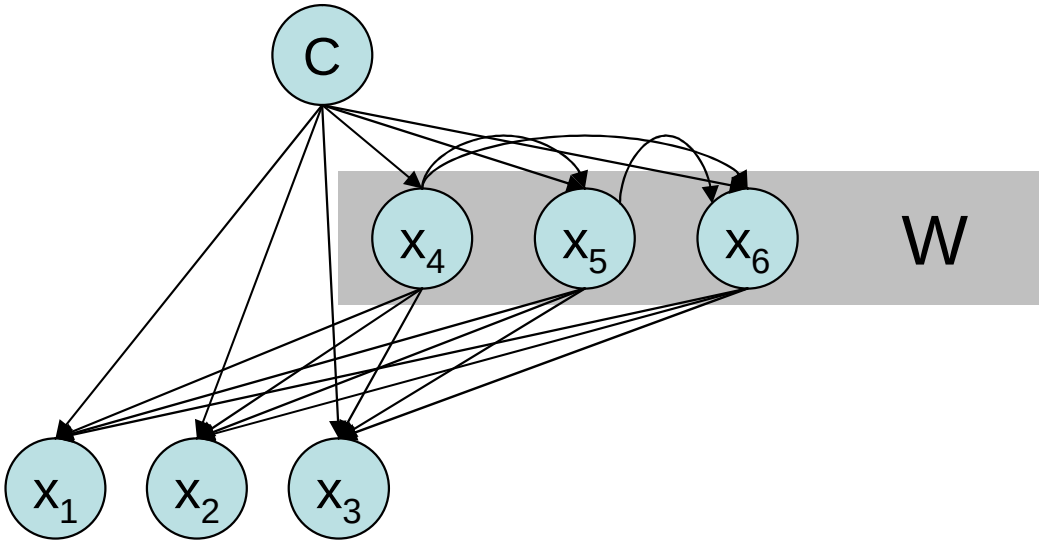
Vergleichsalgorithmen

- Drei Hauptmerkmale von Lazy Learning Algorithmen:
  - Verarbeiten der Eingaben erst bei Informationsanforderung
  - Ausgaben ergeben sich ausschließlich aus gespeicherten Daten
  - Zwischenergebnisse und berechnete Antworten gehen verloren

- LBR (Lazy Bayesian Rules)
  - Für jedes zu klassifizierende $\boldsymbol{x} = <x_i, \ldots, x_n>$, wird eine Menge W an beliebig abhängigen Attributen ausgewählt. Alle anderen Attribute sind voneinander unabhängig
  - Dadurch hängt jedes Attribut von der Klasse und den Attributen in W ab.
  - Anwendung speziell für Klassifizierung weniger Beispiele pro Trainingsset
  - Komplette Neugenerierung eines Bayes‘schen Netzwerkes bei Klassifizierung

Vergleichsalgorithmen

- LBR (cnt'd)
  - Klassifizierung durch: $\arg\max_y \left(\hat{P}(y \mid W) \prod_{i=1}^{n} \hat{P}(x_i \mid y, W) \right)$
  - Berechnungskomplexität im Training: nur Speicherung der Trainingsdaten; Daten-, Zeitkomplexität jeweils $O(tn)$
  - Berechnungskomplexität bei Klassifikation: Auswählen der Attribute für W; $O(tkn^2)$; Datenkomplexität $O(tn)$
- Bayes'sches Netzwerk für LBR



Vergleich TAN & LBR

- Unterschiede
  - In der Art und Anzahl von Abhängigkeiten, die erlaubt werden
    - TAN erlaubt verschiedene Parents der Attribute, allerdings maximal ein parent pro Attribut
    - Bei LBR sind beliebig viele Abhängigkeiten für Attribute erlaubt, allerdings nur von den gleichen Parents
  - In der Art des Lernens
    - Bei TAN bei Eingabe der Trainingsdaten, dadurch Verwendung des selben Netzwerkes
    - Bei LBR auf Anfrage, dadurch Erzeugung eines neuen Netzwerkes pro Test
  - LazyTAN zur Überprüfung des zweiten Unterschiedes
- Kurzevaluation
  - LBR, TAN und LazyTAN haben vergleichbare Fehlerraten
  - Weder LBR noch TAN sind prinzipiell immer besser
  - Der Vorteil gegenüber NB hängt von den Datensets ab

Averaged One-Dependence Estimators

- Warum AODE?
  - LBR und SP-TAN haben liefern ähnliche Fehlerraten wie Boosting Decision Trees
  - Dafür fallen aber hohe Berechnungskosten an. (Ausnahme: LBR für Klassifizierung weniger Beispiele)
- 2 Ursachen für Berechnungskomplexität:
  - Wahrscheinlichkeitsschätzung: „on the fly“ bei LBR und 3D Tabelle mit bedingten Wahrscheinlichkeiten bei SP-TAN
  - Modellauswahl: W bei LBR und $parent()$ bei SP-TAN
- AODE versucht nun
  - die Unabhängigkeitsannahme abzuschwächen, dabei
  - konkurrenzfähige Fehlerraten zu LBR und SP-TAN zu erzielen,
  - allerdings ohne deren Berechnungsaufwand.

Averaged One-Dependence Estimators

1. Ursache: Wahrscheinlichkeitsschätzung

- x-dependence Estimator
  - Wahrscheinlichkeit für jeden Attributwert ist bedingt durch die Klasse und maximal x andere Attribute
  - Allgemein können die benötigten Attribute in einer (x+2)-dimensionalen Tabellen gespeichert werden, indexiert durch
    - den Zielattributwert,
    - die Klasse,
    - die x anderen Attribute von denen der Zielwert abhängig ist.

- Aus Effizienzgründen werden hier 1-dependence Klassifizierer gewählt

Averaged One-Dependence Estimators

2. Ursache: Modellauswahl

- Einfachste Variante: Kein Modell auswählen (siehe NB)
  - Berechnungsaufwand minimiert
  - Außerdem kann sich bei Modellauswahl die Varianz erhöhen
  - Vermeidung einer Modellauswahl kann somit den, durch Varianz entstehenden, Fehleranteil eines Klassifizierers vermindern
- Allerdings muss bei 1-dependence Klassifizierern das eine abhängige Attribut ausgewählt werden.

- Lösung:
  - Auswahl einer begrenzten Zahl an 1-dependence Klassifizierern
  - Bilden eines Durchschnittes der entstehenden Voraussagen

Averaged One-Dependence Estimators

- Intuition
  - Wähle alle 1-dependence Klassifizierer, bei denen ein einzelnes Attribut *parent* von allen anderen ist
  - Beziehe aber nur solche Modelle für die Wahrscheinlichkeits-berechnung mit ein, bei denen der Wert für ein Attribut x_i aus dem zu klassifizierenden $\mathbf{x}$ öfter als oder gleich m –mal in den Trainingsdaten vorkommt.

- Herleitung des Klassifizierers
  - Für jedes x_i gilt wegen der Produktregel:

$$P(y, \mathrm{x}) = P(y, x_i) P(\mathrm{x} \mid y, x_i)$$

Averaged One-Dependence Estimators

- Herleitung des Klassifizierers (cnt'd)
  - Daraus der Durchschnitt über alle gewählten x_i:

$$P(y, \mathrm{x}) = \frac{\sum_{i:1 \le i \le n \wedge F(x_i) \ge m} P(y, x_i) P(\mathrm{x} \mid y, x_i)}{\left| \{ i : 1 \le i \le n \wedge F(x_i) \ge m \} \right|}$$

  - $F(x_i)$ ist hierbei die Anzahl an Trainingsbeispielen, die den Attributwert x_i besitzen
- Klassifizierung durch:

$$\arg\max_{y} \left(\sum_{i:1 \le i \le n \wedge F(x_i) \ge m} \hat{P}(y, x_i) \prod_{j=1}^{n} \hat{P}(x_j \mid y, x_i) \right)$$

  - Ansonsten NB

Averaged One-Dependence Estimators

- Berechnungskomplexität im Training
  - Datenkomplexität: Erzeugen der Tabellen zur Berechnung der Schätzer; $O(k(nv)^2)$
  - Analyse der Häufigkeiten in den Trainingsdaten um die Tabellen zu füllen; $O(tn^2)$

- Berechungskomplexität bei Klassifizierung
  - Es findet keine Modellauswahl statt, da der Durchschnitt über alle x_i gebildet wird
  - Datenkomplexität: Es werden die Wahrscheinlichkeitstabellen aus dem Training verwendet; $O(k(nv)^2)$
  - Zeitkomplexität: Berechnung der Klassifizierungsfunktion; $O(kn^2)$

Komplexität (Übersicht)

	Training		Klassifikation	
Algorithmus	Zeit	Daten	Zeit	Daten
NB	$O(nt)$	$O(knv)$	$O(kn)$	$O(knv)$
TAN	$\mathbf{O(tn^2+kn^2v^2+n^2 \log n)}$	$O(k(nv)^2)$	$O(kn)$	$O(knv^2)$
SP-TAN	$\mathbf{O(tkn^3)}$	$O(tn+k(nv)^2)$	$O(kn)$	$O(knv^2)$
LBR	$O(tn)$	$O(tn)$	$\mathbf{O(tkn^2)}$	$O(tn)$
AODE	$\mathbf{O(tn^2)}$	$O(k(nv)^2)$	$\mathbf{O(kn^2)}$	$O(k(nv)^2)$

k – Anzahl der Klassen

n – Anzahl der Attribute

v – Durchschnittliche Anzahl an Werten pro Attribut

t – Anzahl der Trainingsbeispiele

Averaged One-Dependence Estimators

- Vorteile
  - Inkrementelles Lernen, indem einfach die relevanten Wahrscheinlichkeiten in den Tabellen geändert werden
  - Schneller als SP-TAN & TAN während Training
  - Schneller als LBR während Klassifizierung
- Erwartung: Zusätzlich niedrigere Fehlerrate als NB, weil
  - Schwächere Unabhängigkeitsannahme, da Abhängigkeit von einem Attribut erlaubt wird
  - Allerdings kann $P(y, x_i)$ und $P(x_j \mid y, x_i)$ ungenauer sein als $P(y)$ und $P(x_i, y)$ bei NB, wenn x_i nicht häufig genug in den Trainingsdaten vorkommt
  - Fehlerrate wird deshalb mit steigender Anzahl an Trainingsdaten zusätzlich sinken
  - Hinweise vorhanden, dass eine Aggregation mehrerer vertrauenswürdiger Modelle die Vorhersagegenauigkeit erhöht

Evaluation - Beobachtung

- Fehler

Data	AODE	NB	ODE	LBR	TAN	SP-TAN	J48	Boost J48
Mittel	0,204	0,219	0,216	0,209	0,216	0,211	0,230	0,206
Geo. Mittel		1,124	1,115	1,049	1,102	1,056	1,225	1,026

- Bias

Data	AODE	NB	ODE	LBR	TAN	SP-TAN	J48	Boost J48
Mittel	0,140	0,155	0,147	0,139	0,140	0,142	0,126	0,111
Geo. Mittel		1,160	1,084	1,003	0,998	1,026	0,963	0,773

- Varianz

Data	AODE	NB	ODE	LBR	TAN	SP-TAN	J48	Boost J48
Mittel	0,063	0,062	0,068	0,069	0,075	0,068	0,102	0,093
Geo. Mittel		0,980	1,178	1,131	1,375	1,119	1,735	1,717

Evaluation – Beobachtung

- Win/Draw/Loss, AODE vs Alternativen

	NB		ODE		LBR		TAN	
	W/D/L	p	W/D/L	p	W/D/L	p	W/D/L	p
Fehler	22/7/8	0,008	23/10/4	<0,001	19/6/12	0,281	27/2/8	0,002
Bias	24/9/4	<0,001	19/8/10	0,136	18/4/15	0,728	13/2/22	0,175
Varianz	6/15/16	0,026	23/10/4	<0,001	19/8/10	0,136	31/4/2	<0,001

	SP-TAN		J48		Boosted J48	
	W/D/L	p	W/D/L	p	W/D/L	p
Fehler	23/3/11	0,058	25/0/12	0,047	21/0/16	0,511
Bias	18/3/16	0,864	15/0/22	0,324	10/0/27	0,008
Varianz	25/5/7	0,002	33/0/4	<0,001	32/1/4	<0,001

Evaluation - Ergebnisse

- Fehler
  - Im Durchschnitt geringer als bei allen anderen
  - Allerdings bei W/D/L nur gegenüber NB, ODE, TAN, J48 signifikant geringer
- Bias
  - Im Durchschnitt geringer als bei NB, LBR, TAN, SP-TAN
  - Bei W/D/L signifikant geringer als bei NB, aber nicht signifikant bei LBR, TAN, SP-TAN, J48
  - Höher als bei J48 und Boosted J48, bei W/D/L bei J48 aber nicht signifikant
- Varianz
  - Im Durchschnitt geringer als bei ODE, LBR, TAN, SP-TAN, J48, Boosted J48; allerdings bei W/D/L nicht signifikant bei LBR
  - Im Durchschnitt höher als bei NB, auch bestätigt bei W/D/L, allerdings wäre ein zweiseitiger Test mit 0,052 nur marginal signifikant

Fazit

- Starke Senkung des Bias auf Kosten einer sehr kleinen Erhöhung der Varianz im Vergleich zu NB
- Scheinbar niedrigere Varianz aber höherer Bias als LBR, TAN, SP-TAN, decision trees, allerdings ohne die hohen Trainingskosten von SP-TAN und ohne Klassifizierungskosten von LBR
- Eignung für inkrementelles Lernen

- Die Idee begrenzte Modelle zu aggregieren und damit die Modellauswahl zu vermeiden scheint aufzugehen

Noch Fragen?

Vielen Dank für eure Aufmerksamkeit!