Mathematische Grundlagen III

Maschinelles Lernen I: Klassifikation mit Naive Bayes

Vera Demberg

Universität des Saarlandes

10. Juli 2012

Einleitung

Im vorigen Teil der Vorlesung haben wir Informationstheorie und Anwendungen aus der Computerlinguistik behandelt.
Im nächsten Teil geht es weiter mit Ansätzen zum Maschinellen Lernen.

- Naive Bayes Klassifikation
- Entscheidungsbäume
- Clustering

Inhaltsverzeichnis

Inhaltsverzeichnis

Definition of Learning

Aus Mitchell (1997: 2):
A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

Aus Witten und Frank (2000: 6):
things learn when they change their behavior in a way that makes them perform better in the future.

Maschinelles Lernen

- Künstliche Generierung von Wissen aus Erfahrung
- Erkennung komplexer Muster und Regelmäßigkeiten in vorhandenen Daten
- Ziel: Verallgemeinerung (Generalisierung)
  - Über das Nachschlagen bereits gesehener Beispiele hinausgehen
  - Beurteilung unbekannter Daten
- Beispiele:
  - Die Gleichung einer Geraden anhand zweier Punkten bestimmen.
  - Die Bedeutung eines Wortes in einem neuen Text basierend auf der Bedeutung des Wortes in anderen Texten bestimmen.

Beispiel

Der Manager eines Golf-Clubs möchte wissen, wann er viele Kunden zu erwarten hat, damit er Studenten als Aushilfe einstellen kann, und wann keiner spielen will, damit er seinen Angestellten freigeben kann.

Zwei Wochen lang führt er Buch darüber, wie das Wetter ist und ob er viele oder wenige Kunden an dem Tag hat.

Er schreibt sich auf:

- ob das Wetter heiter, bewölkt oder regnerisch ist,
- wie warm es ist,
- wieviel Luftfeuchtigkeit es gibt,
- ob der Wind stark weht oder nicht,
- ob er viele Kunden er an dem Tag hat

Beispieldatensatz

Wir wollen lernen, bei welchen Wetterkonditionen gespielt wird.

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

Beispiel: Ein numerischer Datensatz

Outlook	Temp.	Humidity	Windy	# Cust.
sunny	85	85	false	12
sunny	80	90	true	10
overcast	83	86	false	30
rainy	70	96	false	15
rainy	68	80	false	16
rainy	65	70	true	5
overcast	64	65	true	24
sunny	72	95	false	10
sunny	69	70	false	18
rainy	75	80	false	20
sunny	75	70	true	25
overcast	72	90	true	18
overcast	81	75	false	32
rainy	71	91	true	8

Terminologie

- **Instanz**: Ein einzelnes Beispiel aus dem Datensatz. Beispiel: eine Zeile aus der Tabelle von der letzten Folie.
- **Attribut / Feature**: Eine Eigenschaft einer Instanz. Beispiel: outlook, temperature, humidity, windy.
- **Wert**: Wert eines Attributs, Beispiel: sunny, overcast, rainy für Attribut outlook.
- **Konzept**: das was wir lernen wollen, Beispiel: eine Klassifikation von Instanzen in spielen und nicht spielen.

Regeln lernen

Beispiel für Regeln, die man aus dem Beispieldatensatz lernen könnte:

if	outlook = sunny	and	humidity = high	then	play = no
if	outlook = rainy	and	windy = true	then	play = no
if	outlook = overcast			then	play = yes
if	humidity = normal			then	play = yes
if	none of the above			then	play = yes

(Dies ist eine Entscheidungsliste: von oben nach unten werden die Regeln durchgegangen, bis eine anwendbare Regel gefunden wird, die dann ausgeführt wird.)

Diese Regeln sind **Klassifikationsregeln**
(eine neue Instanz, wie z.B. "outlook=sunny, temp.=hot, humidity=low, windy=true" könnte damit in *spielen* oder *nicht spielen* klassifiziert werden).

Was wollen wir lernen?

Klassifikation ist nur eine Art von maschinellem Lernen.

Maschinelles Lernen kann in folgende Kategorien von Lernzielen unterteilt werden:

- **Klassifikation**: Instanzen einer vordefinierten Klasse zuordnen. (siehe erster Datensatz)
- **Clustering**: Klassen von Instanzen, die zusammengehören, entdecken.
- **Assoziation**: Relationen zwischen Attributen lernen
- **Numerische Vorhersage (Regression)**: Eine numerische Größe (anstelle einer Klasse) für eine Instanz vorhersagen. (siehe zweiter Datensatz)

Inhaltsverzeichnis

Wiederholung Satz von Bayes

In der Statistik gibt es zwei Hauptansätze:

Frequenzen (Maximum Likelihood Estimate)

$$\underset{m}{\text{argmax}}\, P(d|\mu_m) = \frac{C(Kopf)}{N} = 0.8$$

Bayes'scher Ansatz

$$\underset{m}{\text{argmax}}\, P(\mu_m|d) = \frac{P(d|\mu_m) * P(\mu_m)}{P(d)} < 0.8$$

$P(\mu_m)$: Prior – hier können wir unsere Einschätzung, dass die Münze normal aussieht einfließen lassen, und wie sicher wir uns unserer Einschätzung sind.

Beispiel: Münzwurf

- 10 Würfe
- 8 mal Kopf
- die Münze sieht normal aus

Notation

$\mu = model$

$d = data$

C() = count()

Bayes

Bayes'scher Satz:

$$P(\mu|d) = \frac{P(d|\mu) * P(\mu)}{P(d)}$$

- $P(\mu)$ = a-priori Wahrscheinlichkeit
- $P(d|\mu)$ = Likelihood
- $P(d)$ = Wahrscheinlichkeit der Daten
- $P(\mu|d)$ = a-posteriori Wahrscheinlichkeit

Bayes'sche Entscheidungsregel:

Entscheide für Model μ' falls $P(\mu'|d) > P(\mu_i|d)$ für alle $\mu_i \neq \mu'$

Maximum-a-posteriori (Map)

Maximum-a-posteriori (Map) Wir wollen für das beste Model μ gegeben der Daten d finden:

$$\mu_{map} = \underset{m}{\text{argmax}}\ P(\mu_m|d) \qquad \text{Bayes'scher Satz}$$

$$= \underset{m}{\text{argmax}}\ \frac{P(\mu_m)P(d|\mu_m)}{P(d)} \qquad P(d) \text{ fällt weg, weil es konstant und unabhängig von der Hypothese ist}$$

$$= \underset{m}{\text{argmax}}\ P(\mu_m)P(d|\mu)$$

Maximum Likelihood

Jetzt nehmen wir an, dass wir kein Vorwissen haben. Dann brauchen wir einen einförmigen Prior $P(\mu_i) = P(\mu_j)$ für alle $\mu_i, \mu_j \in \mu$.

Damit vereinfacht sich die Berechnung der a-posteriori-Wahrscheinlichkeit deutlich:

$$\mu_{ML} = \underset{m}{\text{argmax}}\, P(d|\mu_m)$$

Dieses Model nennt man die Maximum Likelihood Hypothese.

Inhaltsverzeichnis

Klassifikation mit Naive Bayes

- Bei einer Klassifikationsaufgabe müssen Instanzen einer Klasse c zugeordnet werden.
- Wir brauchen also vordefinierte Klassen c
- Wir brauchen ein Trainingsset von Instanzen (bestehend aus Attributen a), die den verschiedenen Klassen zugeordnet sind.
- Zur Klassifikation können wir die **Naive Bayes** Methode verwenden.

Naive Bayes Annahme

Naive Bayes Annahme:

- Attribute a sind alle voneinander unabhängig!

$$
\begin{aligned}
c_{map} &= \text{argmax}_{c \in C} P(c|a_1, a_2, \cdots, a_n) \\
&= \text{argmax}_{c \in C} \frac{P(a_1, a_2, \cdots, a_n|c)P(c)}{P(a_1, a_2, \cdots, a_n)} && \text{Bayes'scher Satz} \\
&= \text{argmax}_{c \in C} P(a_1, a_2, \cdots, a_n|c)P(c) && \text{Normalisierung entfällt} \\
&= \text{argmax}_{c \in C} \prod_{i=1}^{n} P(a_i|c)P(c) && \text{Unabhängigkeit}
\end{aligned}
$$

- $P(a|c)$: Relative Häufigkeit des Attributs a unter den Instanzen, die Kategorie c angehören
- In NLP sind die Attribute häufig Wörter – die Unabhängigkeits-annahme ist also tatsächlich sehr naiv.
- Allerdings brauchen wir diese Annahme, da wir sonst nicht genug Beobachtungen haben um Wahrscheinlichkeiten sinnvoll abschätzen zu können.

Naive Bayes Klassifizierer

Beispiel

- Klassifizierung von Instanz $outlook = sunny$, $temperature = cool$, $humidity = high$, $windy = true$:

$$\begin{aligned} c_{map} &= \text{argmax}_{c \in yes,no} \; P(c) \cdot \textstyle\prod_i P(a_i|c) \\ &= \text{argmax}_{c \in yes,no} \; P(c) \cdot P(outlook = sunny|c) \\ &\quad \cdot P(temperature = cool|c) \\ &\quad \cdot P(humidity = high|c) \\ &\quad \cdot P(windy = true|c) \end{aligned}$$

- Berechnung der relativen Häufigkeit von $P(sunny|yes)$:

  Insgesamt gibt es 9 Instanzen mit $play = yes$,
  davon 2 bei denen $outlook = sunny$,
  also ist $P(sunny|yes) = \frac{2}{9}$

Naive Bayes Klassifizierer

Vorgehen

- Zuerst die Wahrscheinlichkeit jeder Kategorie in den Trainingsdaten berechnen

  Gesamthäufigkeiten für "play": $yes = 9$, $no = 5$

  Wahrscheinlichkeiten: $yes = \frac{9}{14}$, $no = \frac{5}{14}$

- Für jede Kategorie die Wahrscheinlichkeit der einzelnen Attribute berechnen

		freq.		P	
		yes	*no*	*yes*	*no*
Outlook	sunny	2	3	$\frac{2}{9}$	$\frac{3}{5}$
	overcast	4	0	$\frac{4}{9}$	0
	rainy	3	2	$\frac{3}{9}$	$\frac{2}{5}$
	$\sum$	9	5	1	1
Temperature	...				
Humidity	...				

Naive Bayes Klassifizierer

Vorgehen, Teil 2

- Wahrscheinlichkeit jeder Kategorie für die vorliegende Instanz berechnen

$$P(yes) \cdot P(sunny|yes) \cdot P(cool|yes) \cdot P(high|yes) \cdot P(true|yes)$$
$$= \frac{9}{14} * \frac{2}{9} * \frac{3}{9} * \frac{3}{9} * \frac{3}{9} = 0.0053$$

$$P(no) \cdot P(sunny|no) \cdot P(cool|no) \cdot P(high|no) \cdot P(true|no)$$
$$= \frac{5}{14} * \frac{3}{5} * \frac{1}{5} * \frac{4}{5} * \frac{3}{5} = 0.0206$$

- Kategorie zuweisen

$$\begin{aligned} c_{map} &= \underset{c \in yes,no}{\text{argmax}} \; P(c) \cdot \prod_i P(a_i|c) \\ &= no \end{aligned}$$

Naive Bayes Klassifizierer

Textklassifikation

Bei der Klassifikation von Dokumenten werden die Wörter des Textes als Attribute eingesetzt:

$$\begin{aligned} c_{map} &= \underset{c}{\text{argmax}}\ P(c)P(d|c) \\ &= \underset{c}{\text{argmax}}\ P(c)\prod_i P(w_i|c) \end{aligned}$$

$P(w_i|c)$: Wahrscheinlichkeit, dass Wort w_i in einem Dokument mit Klasse c vorkommt

Klassifikation nach Bayes

Anwendung: Spam herausfiltern

- Datensatz: Ein Korpus von E-Mail-Nachrichten, annotiert als "Spam" oder "Ham"
- Aufgabe: Eine neue Nachricht als "Spam" oder "Ham" klassifizieren
- Attribute: Vokabular der E-Mail-Nachrichten im Trainingskorpus

Bsp.: Eine E-Mail mit dem Inhalt "Get rich fast!!!"

$$
\begin{aligned}
c_{map} &= \underset{c \in spam, ham}{\operatorname{argmax}} \; P(c) \cdot \prod_i P(a_i|c) \\
&= \underset{c \in spam, ham}{\operatorname{argmax}} \; P(c) \cdot P(get|c) \cdot P(rich|c) \cdot P(fast|c) \cdot P(!!!|c)
\end{aligned}
$$

Anwendung: Spam Herausfiltern

- Problem: was passiert, wenn wir "rich" und "!!!" *nie* in Ham-E-Mails gesehen haben?
- Null-Werte führen dazu, dass Kategorien ununterscheidbar werden
  $P(ham) \cdot P(get|ham) \cdot P(rich|ham) \cdot P(fast|ham) \cdot P(!!!|ham)$
  $= P(ham) \cdot P(get|ham) \cdot 0 \cdot P(fast|ham) \cdot 0 = \boxed{0}$!
- Mögliche Lösung: Kleine Werte zu Zähler und Nenner hinzufügen, damit beide ungleich null werden (smoothing)

$$P(a_x|c) = \frac{n_x + \frac{1}{k}}{n + 1}$$

n : Anzahl Instanzen in Kategorie c
n_x : Anzahl Instanzen in Kategorie c, bei denen $a = x$
k : Anzahl Werte, die Attribut a annehmen kann

Eigenschaften von Naive Bayes Klassifizierern

- Inkrementalität: bei jedem neuen Trainingsbeispiel können die a-priori-Wahrscheinlichkeit und die likelihood neu geschätzt werden: flexibel und robust.
- Kombiniert Vorwissen mit beobachteten Daten: A-priori-Wahrscheinlichkeit wird multipliziert mit der Wahrscheinlichkeit des Models gegeben den Trainingsdaten.
- Probabilistische Hypothesen: generiert nicht nur Klassifikation sondern auch Wahrscheinlichkeitsverteilung über die verschiedenen möglichen Klassen.
- Meta-Klassifikation: die Ausgaben mehrerer Klassifizierer können kombiniert werden (z.B. durch Multiplikation der Wahrscheinlichkeiten die die verschiedenen Klassifizierer für eine bestimmte Klasse vorhersagen).

Zusammenfassung Naive Bayes Klassifikation

- Der Naive Bayes Klassifizierer kombiniert Vorwissen mit beobachteten Daten.
- Berechnet die maximum a posterior (MAP) Hypothese oder die maximum likelihood (ML) hypothesis (wie MAP aber mit uniformem Prior).
- Der Naive Bayes Klassifizierer nimmt Unabhängigkeit zwischen den Attributen an, und weist einer neuen Instanz die Klasse mit MAP Wahrscheinlichkeit zu.
- Wahrscheinlichkeiten können von Frequenzen geschätzt werden. Problem: seltene / ungesehene Ereignisse $\rightarrow$ smoothing.

Inhaltsverzeichnis

Wortbedeutungsdisambiguierung

- Viele Worte sind mehrdeutig
- Für viele NLP Anwendungen bekommen wir deutlich bessere Ergebnisse, wenn wir die Mehrdeutigkeit auflösen.
  - Maschinelle Übersetzung
  - Fragebeantwortung
  - Suche
  - Automatische Zusammenfassung
  - und viele andere

Beispiele

Bank → Geldinstitut oder Parkbank?
Decke → Bettdecke oder Zimmerdecke?
Angeln → Anglerwerkzeug oder Aktivität Fische zu fangen oder Türangeln?

Mehrdeutige Wörter

- mehrdeutige Wörter mit total unterschiedlichen Bedeutungen wie "Bank" oder "Decke" sind eher untypisch
- oft haben Wörter mehrere ähnliche Bedeutungen

Beispiel zum Mitmachen

Welche Bedeutungen von "Titel" fallen Ihnen ein?

- viele Bedeutungen sind schwer zu unterscheiden
- Wörterbücher sind weder vollständig noch konsistent
- manachmal kann ein Wort auch verschiedene Wortarten haben (z.B. "vor" – Präposition oder Verbpartikel?)
  - wird über POS-tagging gelöst
  - Probleme recht unterschiedlich
  - POS tagging: lokaler Kontext umgebender Wortarten hilft weiter.
  - Bedeutung von anderen Wörtern im Textabschnitt hilft weiter.

Mehrdeutige Wörter

- mehrdeutige Wörter mit total unterschiedlichen Bedeutungen wie "Bank" oder "Decke" sind eher untypisch
- oft haben Wörter mehrere ähnliche Bedeutungen

Beispiel zum Mitmachen

Welche Bedeutungen von "Titel" fallen Ihnen ein?

- viele Bedeutungen sind schwer zu unterscheiden
- Wörterbücher sind weder vollständig noch konsistent
- manachmal kann ein Wort auch verschiedene Wortarten haben (z.B. "vor" – Präposition oder Verbpartikel?)
  - wird über POS-tagging gelöst
  - Probleme recht unterschiedlich
  - POS tagging: lokaler Kontext umgebender Wortarten hilft weiter.
  - Bedeutung von anderen Wörtern im Textabschnitt hilft weiter.

Training

- Wir betrachten heute vorallem überwachte Lernverfahren.
- Für überwachtes Lernen braucht man Trainingsdaten.
- Ausreichend große Textmenge mit Wortbedeutungen zu annotieren ist teuer.
- Pseudoworte als Training (Gale et al. 1992, Schütze 1992)
  - Ersetze zwei unterschiedliche Wörter durch ein Pseudowort
  - z.B. alle Vorkommen von *Banane* und *Tür* ersetzen durch Banane-Tür.
  - so können wir leicht beliebig grosse Test- und Trainingssets erstellen.

Naive Bayes für Wortbedeutungsdisambiguierung

Naive Bayes Klassifikation

Bayes'scher Satz: $P(s_k|c) = \frac{P(c|s_k)P(s_k)}{P(c)}$

Wir wählen die wahrscheinlichste Bedeutung s':

$$\begin{aligned} s' &= \text{argmax}_{s_k} P(s_k|c) \\ &= \text{argmax}_{s_k} P(c|s_k)P(s_k) \\ &= \text{argmax}_{s_k} [\log P(c|s_k) + \log P(s_k)] \end{aligned}$$

Notation

- w ein mehrdeutiges Wort
- s_k Bedeutungen von w
- c_i Kontexte von w im Korpus
- v_j Wörter aus dem Kontext die als Attribute für die Disambiguierung benutzt werden.

Naive Bayes für Wortbedeutungsdisambiguierung

Naive Bayes Klassifikation

Bayes'scher Satz: $P(s_k|c) = \frac{P(c|s_k)P(s_k)}{P(c)}$

Wir wählen die wahrscheinlichste Bedeutung s':

$$\begin{aligned} s' &= \text{argmax}_{s_k} P(s_k|c) \\ &= \text{argmax}_{s_k} P(c|s_k)P(s_k) \\ &= \text{argmax}_{s_k} [\log P(c|s_k) + \log P(s_k)] \end{aligned}$$

mit Naive Bayes: $P(c|s_k) = \prod_{v_j \in c} P(v_j|s_k)$

Notation

- w ein mehrdeutiges Wort
- s_k Bedeutungen von w
- c_i Kontexte von w im Korpus
- v_j Wörter aus dem Kontext die als Attribute für die Disambiguierung benutzt werden.

Naive Bayes für Wortbedeutungsdisambiguierung

Naive Bayes Klassifikation

Bayes'scher Satz: $P(s_k|c) = \frac{P(c|s_k)P(s_k)}{P(c)}$

Wir wählen die wahrscheinlichste Bedeutung s':

$$\begin{aligned} s' &= \text{argmax}_{s_k} P(s_k|c) \\ &= \text{argmax}_{s_k} P(c|s_k)P(s_k) \\ &= \text{argmax}_{s_k} [\log P(c|s_k) + \log P(s_k)] \end{aligned}$$

mit Naive Bayes: $P(c|s_k) = \prod_{v_j \in c} P(v_j|s_k)$

dann schätzen wir mit MLE:

$$P(v_j|s_k) = \frac{C(v_j, s_k)}{\sum_t C(v_t, s_k)} \text{ und } P(s_k) = \frac{C(s_k)}{C(w)}$$

Notation

- w ein mehrdeutiges Wort
- s_k Bedeutungen von w
- c_i Kontexte von w im Korpus
- v_j Wörter aus dem Kontext die als Attribute für die Disambiguierung benutzt werden.

Resultate für Naive Bayes Klassifikator

- Wir können nach dem Training sehen, welches die aussagekräftigsten Attribute für eine Entscheidung zwischen zwei Klassen sind.
- a_i ist ein guter Prediktor für Bedeutung s_k im Gegensatz zu Bedeutung s_l, wenn $P(a_i|s_k) > P(a_i|s_l)$.
- Beispiel: Disambiguierung von *drug* als entweder *'illicit substance'* oder *medication*.

$$P(\textit{prescription}|\text{`}\textit{illicit substance}\text{'}) < P(\textit{prescription}|\text{`}\textit{medication}\text{'})$$

Beispiel: Disambiguierung von "drug"

Bedeutung	aussagekräftigste Attribute
medication	prices, prescription, patent, increase, consumer, pharmaceutical
illegal substance	abuse, paraphernalia, illicit, alcohol, cocaine, traffickers

Frage:

Bislang sind wir davon ausgegangen, dass wir Trainingsdaten haben, in denen die Wortbedeutungen annotiert sind.
Wie können wir verfahren, wenn wir einfach nur eine große Menge unannotierten Text haben?

EM-Algorithmus

Naive Bayes Klassifikation

Wir wählen die wahrscheinlichste Bedeutung s':

$$s' = \text{argmax}_{s_k}[\log P(c|s_k) + \log P(s_k)]$$

mit Naive Bayes: $P(c|s_k) = \prod_{v_j \in c} P(v_j|s_k)$

Supervised:

$$P(v_j|s_k) = \frac{C(v_j, s_k)}{\sum_t C(v_t, s_k)}$$

$$P(s_k) = \frac{C(s_k)}{C(w)}$$

Notation

- w ein mehrdeutiges Wort
- s_k Bedeutungen von w
- c_i Kontexte von w im Korpus
- v_j Wörter aus dem Kontext die als Attribute für die Disambiguierung benutzt werden.

EM-Algorithmus

- Wenn wir nicht wissen, welche Bedeutungen ein Wort hat, dann wissen wir vielleicht auch nicht wie viele Bedeutungen es hat (oder wie viele seiner Bedeutungen in unserem Korpus vorkommen).
- je mehr Bedeutungen wir annehmen, desto besser wird das Model.
- Wie beschränken wir die Anzahl von Bedeutungen?

Bedeutungsdisambiguierung mit Mutual Information

- Naive Bayes Klassifizierer versucht, Information von allen Wörtern aus dem Kontext von Wort w zu verwenden. (Dafür macht Naive Bayes diese etwas unrealistische Annahme, dass alle Wörter unabhängig voneinander sind.)
- Der Informationstheoretische Ansatz mit KL Divergenz nimmt einen anderen Ansatz:
- Identifiziere ein ganz besonders aussagekräftiges und verlässliches Wort im Kontext und disambiguiere bezogen auf dieses Wort.
- Bei der Klassifizierung eines neuen Worts muss dieses eine Wort dann aber auch im Kontext vorhanden sein. Daher besser geeignet für bestimmte Disambiguierungsprobleme, wie z.B. Übersetzung von leichten Verben
  prendre + mesure $\rightarrow$ to take (a measure)
  prendre + décision $\rightarrow$ to make (a decision)

Klassifikation mit Mutual Information

Beispiel

Bei der Übersetzung von "prendre" wollen wir lernen, dass "mesure" ein guter Hinweis auf die richtige Übersetzung als "take" ist, und "décision" ein guter Hinweis auf die Übersetzung als "make" ist.

- wähle ein Wort aus dem Kontext von "prendre", das die mutual information zwischen dem Wort und der richtigen Übersetzung von "prendre" maximiert.
- Eine effiziente Implementierung is der Flip-Flop Algorithmus (Breiman et al. 1984)

Mutual Information

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} P(x,y) \log \frac{P(x,y)}{P(x)P(y)}$$

Flip-Flop Algorithmus

Daten

prendre mesure → take measure
prendre note → take notes
prendre exemple → take an example
prendre décision → make a decision
prendre parole → speak

- Ziel: Disambiguierung von *prendre*

Flip-Flop Algorithmus

find random partition $Y = \{Y_1, Y2\}$ of $\{y_1..y_m\}$
while (improving) do
find partition $X = \{X_1, X_2\}$ of $\{x_1..x_n\}$ that maximizes $I(X; Y)$
find partition $Y = \{Y_1, Y_2\}$ of $\{y_1..y_m\}$ that maximizes $I(X; Y)$
end

$\{y_1..y_m\} = \{\text{take, make, speak}\}$
$\{x_1..x_n\} = \{\text{mesure, note, exemple, décision, parole}\}$

Zusammenfassung

- Maschinelles Lernen in der Praxis bedeutet automatisch Regeln und Muster in Beispielen (Daten) zu finden.
- Ein typisches Lernproblem ist die Klassifizierung: wir wollen eine Instanz einer bestimmten Klasse zuordnen.
- Um das zu tun, gibt es verschiedene Lernmethoden. Heute haben wir uns den Naive Bayes Klassifizierer angeschaut.
- Bayes'sche Statistik kombiniert Vorwissen mit Beobachtungen
- Naive Bayes macht die Annahme, dass die Attribute einer Instanz voneinander unabhängig sind.
- Fallbeispiele:
  - Textklassifizierung von Emails in Ham und Spam
  - Bedeutungsdisambiguierung von Wörtern

Diskussion

Wir haben gesehen, dass man Trainingsdaten automatisch erzeugen kann durch die Einführung von Pseudoambiguitäten: “banana-door”

Viele natürliche Wörter haben aber unterschiedliche Bedeutungen, die einander ähnlich sind (*Dieses Bild hat keinen Titel*).

Inwiefern modellieren Pseudoworte die Ambiguität natürlicher Sprache?