

Mathe III: Statistische Methoden

Teil 2

Garance PARIS

Sommersemester 2010

Probabilistische kontextfreie Grammatiken

22. Juni 2010

Überblick

- Motivierung
- Probabilistische kontextfreie Grammatiken
- Wahrscheinlichkeit eines Parsebaums
- Wahrscheinlichkeit eines Satzes
- Annahmen
- Verwendete Algorithmen

Literatur: Manning and Schütze

Ambiguität beim Parsing

- Wörter können verschiedene Bedeutungen haben und mehr als einer Wortkategorien angehören
- Verschiedene Wortkategorien führen zu verschiedenen Satzstrukturen
- Es gibt auch andere strukturelle Ambiguitäten, die nicht aus dem Lexikon stammen

Lösung: Statistik hinzuziehen

- Zuerst mit PoS-Tagging für jedes Wort die beste Kategorie bestimmen
- Dann den besten Parsebaum für den Satz bestimmen:

$$\operatorname{argmax}_t P(t|\text{Satz}, \text{Grammatik})$$

Weitere Vorteile

- Kann bei der Effizienz helfen
- Geeignet für die kognitive Modellierung menschlichen Verhaltens

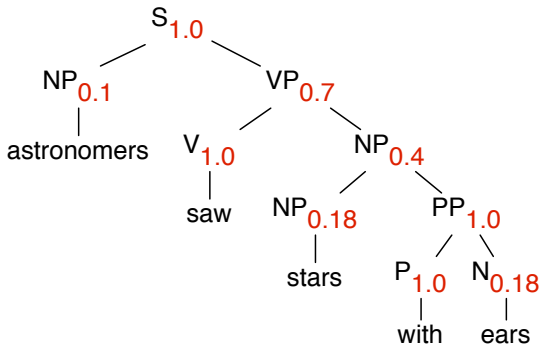
Probabilistische kontextfreie Grammatiken (PCFGs)

- Eine PCFG ist eine kontextfreie Grammatik, in der jede Regel mit einer Wahrscheinlichkeit versehen ist
- Die Summe der Wahrscheinlichkeiten aller Regeln mit dem selben Symbol auf der linken Seite muss 1 betragen
- Beispiel:

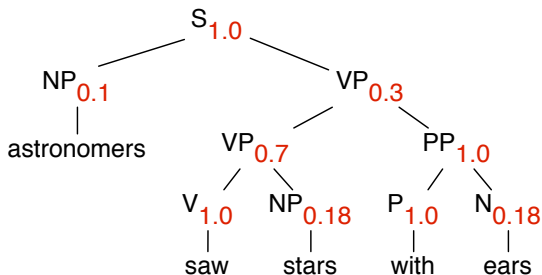
S	→	NP VP	1.0	NP	→	astronomers	0.1
VP	→	V NP	0.7	NP	→	telescopes	0.1
VP	→	VP PP	0.3	NP	→	saw	0.04
NP	→	NP PP	0.4	NP	→	stars	0.18
PP	→	P NP	1.0	NP	→	ears	0.18
				P	→	with	1.0
				V	→	saw	1.0

Wahrscheinlichkeit eines Parsebaums

Die Wahrscheinlichkeit eines Parses ist das Produkt der Wahrscheinlichkeiten der Regeln, die während des Parsens angewandt werden



$$\begin{aligned}
 P(t1) &= 1.0 * 0.1 * 0.7 * 1.0 * 0.4 * 0.18 * 1.0 * 1.0 * 0.18 \\
 &= 0.009072
 \end{aligned}$$



$$\begin{aligned}
 P(t_2) &= 1.0 * 0.1 * 0.3 * 0.7 * 1.0 * 0.18 * 1.0 * 1.0 * 0.18 \\
 &= 0.0006804
 \end{aligned}$$

Wahrscheinlichkeit eines Satzes

Die Wahrscheinlichkeit eines Satzes ist die Summe der Wahrscheinlichkeiten aller Parsebäume:

$$P(\text{Satz}) = P(t_1) + P(t_2) = 0.0015876$$

Formell gesehen

Wahrscheinlichkeit eines Parsebaums:

$$P(t, \text{Satz}) = \prod_{n \in t} P(r(n))$$

Wahrscheinlichkeit eines Satzes:

$$P(\text{Satz}) = \sum_t P(\text{Satz}, t)$$

t : Parsebaum für den Satz

n : Knoten im Baum

r : Regel, die bei der Expansion eines Knotes angewandt wird

Annahmen bei PCFGs

- Positionsunabhängigkeit: Die Wahrscheinlichkeit eines Teilbaums ist unabhängig davon, wo im Satz die entsprechende Wortfolge vorkommt (vgl. Zeitunabhängigkeit bei HMMs)
- Kontextunabhängigkeit: Die Wahrscheinlichkeit eines Teilbaums ist unabhängig von Wörtern, die er nicht dominiert
- Vorfahrenunabhängigkeit: Die Wahrscheinlichkeit eines Teilbaums ist unabhängig von Vorgängerknoten im Baum

Zu beantwortenden Fragen

- Was ist die Wahrscheinlichkeit eines Satzes gegeben eine Grammatik?
- Was ist der wahrscheinlichste Parsebaum für einen Satz?
- Parameterschätzung: Wie ordnen wir den Regeln Wahrscheinlichkeiten zu?

Algorithmen dazu

- Wahrscheinlichkeit eines Satzes:
Inside- und Outside-Algorithmen
- Den wahrscheinlichsten Parsebaum bestimmen:
Viterbi-Algorithmus
- Den Regeln Wahrscheinlichkeiten zuweisen:
Inside-Outside Algorithm

Parallel mit HMMs

- Diese Algorithmen sind ähnlich den Algorithmen, die beim PoS-Tagging eingesetzt werden
- Inside- und Outside-Wahrscheinlichkeiten entsprechen Forward- und Backward-Wahrscheinlichkeiten

Chomsky Normalform

Die Algorithmen gelten im Prinzip nur für Grammatiken in Chomsky-Normalform, d. h. Grammatiken, in denen

- alle Grammatikregeln binär sind,
 $X \rightarrow YZ$
- und nur lexikalische Regeln unär
 $X \rightarrow w$

(X, Y, Z : Nicht-Terminale, w : Wort)

Alle sonstige Grammatiken können aber in eine Grammatik in Chomsky-Normalform umgeformt werden

Zu schätzenden Parameter

Wenn die Grammatik in Chomsky-Normalform ist, sind die zu schätzenden Parameter:

- Eine Matrix mit n^3 Elementen bei n Nicht-Terminalen der Form $X \rightarrow YZ$
- $n \cdot V$ Parameter bei V Terminalen der Form $X \rightarrow w$

Außerdem muss gelten:

$$\sum_{Y,Z} P(X \rightarrow YZ) + \sum_w P(X \rightarrow w) = 1$$

Erwerb der Regelwahrscheinlichkeiten

Zwei Möglichkeiten:

- Automatischer Erwerb:
Es wird versucht, die Parameter zu finden, bei denen ein gewisses Training-Korpus am wahrscheinlichsten ist
 - Erfordert eine gute Grammatik im Voraus, um die Anzahl der Kombinationen einzuschränken
 - Es existieren viele lokale Maxima
 - Funktioniert selten gut
- Externe Wahrscheinlichkeiten verwenden:
z. B. Wahrscheinlichkeiten aus einer Baubank ablesen
 - Setzt Einschränkungen auf die Sprachen und Korpora, die wir zum Trainieren verwenden können
 - Sparse-Data Probleme

Probleme mit PCFGs

- Kontext spielt keine Rolle, aber wir wissen, dass Personalpronomen in Subjektposition häufiger als in Objektposition sind
(“*NP* → *Pronoun*” müsste zwei unterschiedliche, kontextabhängige Wahrscheinlichkeiten haben)
- Lexikalisches Wissen (Subkategorisierung, Selektionseinschränkungen) spielt keine Rolle
- Globale strukturelle Präferenzen haben keine Auswirkung (z.B. in Bezug auf die Anbindung von PPs, Relativsätze, Adverbien, usw.)

Mögliche Lösungen

- Lexikalisierung der Nicht-Terminalen (s.u.)
- Parentisierung:
Ähnliches Prozess, nur hier wird ein bestimmter Vorgängerknoten zur Vorbedingung einer Regel gemacht
- Weitere "Tricks":
Subkategorisierung einführen, Traces, Interpunktion, Clustering, usw.

Eine Grammatik lexikalisieren

Lexikalisierung der erste VP-Regel, $VP \rightarrow V NP$ (0.7):

VP_{saw}	\rightarrow	V_{saw}	$NP_{astronomers}$	0.1
VP_{saw}	\rightarrow	V_{saw}	NP_{ears}	0.15
VP_{saw}	\rightarrow	V_{saw}	NP_{saw}	0.05
VP_{saw}	\rightarrow	V_{saw}	NP_{stars}	0.3
VP_{saw}	\rightarrow	V_{saw}	$NP_{telescopes}$	0.1

- Aus einer Regel werden fünf!
- Zu viele Parameter müssen geschätzt werden (Sparse-Data)
- Typischerweise wird nur den lexikalischen Kopf als Bedingung genommen:

$VP_{saw} \rightarrow V_{saw} NP$ (0.7)_n

Stochastisches Parsing

24. Juni 2010

Motivierung

- Der naive Algorithmus, um die Wahrscheinlichkeit eines Satzes zu berechnen, ist exponential (wie bei HMMs)
- Lösung:
Teilergebnisse (Wahrscheinlichkeit einzelner Wortketten) speichern anstatt sie immer wieder neu zu berechnen

Motivierung

- Der naive Algorithmus, um die Wahrscheinlichkeit eines Satzes zu berechnen, ist exponential (wie bei HMMs)
- Lösung:
Teilergebnisse (Wahrscheinlichkeit einzelner Wortketten) speichern anstatt sie immer wieder neu zu berechnen

Speichervariablen

$\beta_X(p, q)$: Wahrscheinlichkeit, dass die Nicht-Terminals X eine Wortkette mit Anfangspunkt p und Endpunkt q überspannt

(Für die Speichervariablen wählen wir den Buchstaben Beta, weil die Variablen denen im Backward-Algorithmus beim PoS-Tagging ähnlich sind)

Der Inside-Algorithmus

Induktionsanfang

Die Wahrscheinlichkeit des Vor-Terminal-Baums mit Mutter X und Tochter p ist die Wahrscheinlichkeit der entsprechenden Regel:

$$\beta_X(p, p) = P(X \rightarrow p)$$

Algorithmen

Der Inside-Algorithmus *Induktionsanfang*

Die Wahrscheinlichkeit des Vor-Terminal-Baums mit Mutter X und Tochter p ist die Wahrscheinlichkeit der entsprechenden Regel:

$$\beta_X(p, p) = P(X \rightarrow p)$$

Induktionsschritt (Bottom-Up)

Die Wahrscheinlichkeit eines Subbaums ist die Summe über alle möglichen Regelanwendungen des Produktes von Regelwahrscheinlichkeit und Wahrscheinlichkeiten der jeweiligen Unterteile

$$\beta_X(p, r) = \sum_{Y, Z} \sum_q P(X \rightarrow YZ) \beta_Y(p, q) \beta_Z(q+1, r)$$

Beispiel

Eingabesatz:

astronomers₁ saw₂ stars₃ with₄ ears₅

Induktionsanfang

$$\beta_{NP}(1, 1) = P(NP \rightarrow \text{astronomers}) = 0.1$$

$$\beta_V(2, 2) = P(V \rightarrow \text{saw}) = 1.0$$

$$\beta_{NP}(2, 2) = P(NP \rightarrow \text{saw}) = 0.04$$

$$\beta_{NP}(3, 3) = P(NP \rightarrow \text{stars}) = 0.18$$

$$\beta_P(4, 4) = P(P \rightarrow \text{with}) = 1.0$$

$$\beta_{NP}(5, 5) = P(NP \rightarrow \text{ears}) = 0.18$$

Beispiel, Teil 2

Induktionsschritt

$$\beta_{VP}(2,3) = P(VP \rightarrow VNP) \beta_V(2,2) \beta_{NP}(3,3) \\ = 0.126$$

$$\beta_{PP}(4,5) = P(PP \rightarrow PNP) \beta_P(4,4) \beta_{NP}(5,5) \\ = 0.18$$

$$\beta_{NP}(3,5) = P(NP \rightarrow NPPP) \beta_{NP}(3,3) \beta_{PP}(4,5) \\ = 0.01296$$

$$\beta_{VP}(2,5) = P(VP \rightarrow VNP) \beta_V(2,2) \beta_{NP}(3,5) \\ + P(VP \rightarrow VPPP) \beta_{VP}(2,3) \beta_{PP}(4,5) \\ = 0.015876$$

$$\beta_S(1,5) = P(S \rightarrow NPVP) \beta_{NP}(1,1) \beta_{VP}(2,5) \\ = 0.0015876$$

Der Viterbi-Algorithmus

- Wie bei HMMs gibt's einen Inside-ähnlichen Algorithmus, der den wahrscheinlichsten Parsebaum eines Satzes berechnet: Viterbi
- Es werden Speichervariablen eingeführt, die die Wahrscheinlichkeit des besten Unterbaums speichern: $\delta_j(p, q)$
- Außerdem werden Rückverfolgungsvariablen eingesetzt, die ψ s, um sich zu merken, welche Regel zum besten Teilbaum führte:

$\psi_X(p, q) = (Y, Z, r)$ speichert die Anwendung von $X \rightarrow YZ$, wobei die Unterbäume bei r getrennt werden

Der Viterbi-Algorithmus

Induktionsanfang

Wie bei Inside: $\delta_X(p, p) = P(X \rightarrow p)$

Induktionsschritt

$$\delta_X(p, r) = \max_{Y, Z, q} P(X \rightarrow YZ) \delta_Y(p, q) \delta_Z(q+1, r)$$

Falls es kein eindeutigen Maximum gibt, findet eine zufällige Auswahl statt

Verfolgungsvariable speichern:

$$\psi_X(p, r) = \operatorname{argmax}_{Y, Z, q} P(X \rightarrow YZ) \delta_Y(p, q) \delta_Y(q+1, r)$$

Terminierung

Wahrscheinlichkeit des wahrscheinlichsten Baum: $\delta_X(1, n)$
(n : Länge der Wortkette)

Zurückverfolgung des Pfades

Den wahrscheinlichsten Baum kann wie folgt gefunden werden:

- 1 Die Wurzel des wahrscheinlichsten Baums ist $\psi_X(1, n)$
- 2 Sei $\psi_X(p, r) = (Y, Z, q)$ die Verfolgungsvariable von X
- 3 Die linke und rechte Töchter von X sind:

$$\begin{aligned}\text{left}(X) &= Y_{p,q} \\ \text{right}(X) &= Z_{q+1,r}\end{aligned}$$

Beispiel

Induktionsanfang

$$\delta_{NP}(1, 1) = P(NP \rightarrow \text{astronomers}) = 0.1$$

$$\delta_V(2, 2) = P(V \rightarrow \text{saw}) = 1.0$$

$$\delta_{NP}(2, 2) = P(NP \rightarrow \text{saw}) = 0.04$$

$$\delta_{NP}(3, 3) = P(NP \rightarrow \text{stars}) = 0.18$$

$$\delta_P(4, 4) = P(P \rightarrow \text{with}) = 1.0$$

$$\delta_{NP}(5, 5) = P(NP \rightarrow \text{ears}) = 0.18$$

Induktionsschritt

$$\delta_{VP}(2, 3) = P(VP \rightarrow VNP) \delta_V(2, 2) \delta_{NP}(3, 3)$$

$$= 0.126$$

$$\psi_{VP}(2, 3) = (V, NP, 2)$$

Induktionsschritt, Teil 2

$$\begin{aligned} \delta_{PP}(4, 5) &= P(PP \rightarrow PNP) & \delta_P(4, 4) & \delta_{NP}(5, 5) \\ &= 0.18 \end{aligned}$$

$$\psi_{PP}(4, 5) = (P, NP, 4)$$

$$\begin{aligned} \delta_{NP}(3, 5) &= P(NP \rightarrow NP PP) & \delta_{NP}(3, 3) & \delta_{PP}(4, 5) \\ &= 0.01296 \end{aligned}$$

$$\psi_{NP}(3, 5) = (NP, PP, 3)$$

$$\begin{aligned} \delta_{VP}(2, 5) &= \max(P(VP \rightarrow VNP) \delta_V(2, 2) \delta_{NP}(3, 5), \\ & \quad (P(VP \rightarrow VPP) \delta_{VP}(2, 3) \delta_{PP}(4, 5))) \\ &= 0.009072 \end{aligned}$$

$$\psi_{VP}(2, 5) = (V, NP, 2)$$

$$\begin{aligned} \delta_S(1, 5) &= P(S \rightarrow NP VP) & \delta_{NP}(1, 1) & \delta_{VP}(2, 5) \\ &= 0.0015876 \end{aligned}$$

$$\psi_S(1, 5) = (NP, VP, 1)$$

Zurückverfolgung des Pfades

$$\begin{aligned}\psi_S(1, 5) &= (NP, VP, 1) \\ \text{left}(S) &= NP_{1,1} \\ \text{right}(S) &= VP_{1+2,5} = VP_{2,5} \\ \psi_{VP}(2, 5) &= (V, NP, 5) \\ \text{left}(VP) &= V_{2,2} \\ \text{right}(VP) &= NP_{3,5} \\ \psi_{NP}(3, 5) &= (NP, PP, 3) \\ \text{left}(NP) &= NP_{3,3} \\ \text{right}(NP) &= PP_{4,5} \\ \psi_{PP}(4, 5) &= (P, NP, 4) \\ \text{left}(PP) &= P_{4,4} \\ \text{right}(PP) &= NP_{5,5}\end{aligned}$$