Mathematische Grundlagen der Computerlinguistik III: Statistische Methoden

Probeklausur

Crocker/Demberg/Staudte

Sommersemester 2014
17.07.2014

1. Sie haben 90 Minuten Zeit zur Bearbeitung der Aufgaben.
2. Erlaubte Hilfsmittel sind 1 Blatt selbsterstellte (handgeschriebene) Zusammenfassung des Stoffes und ein Taschenrechner (kein Handy!).
3. Die einzelnen Lösungsschritte sollen soweit begründet werden, dass der Lösungsansatz nachvollziehbar ist.

Viel Erfolg!!!

1. **Stochastisches Parsing**

   a) Vergleiche mit eigenen Worten die *"Inside"-Prozedur* und den *"Viterbi"-Algorithmus*: Was berechnen sie, was haben sie gemeinsam, was unterscheidet sie?

   Der Inside-Algorithmus berechnet die Wahrscheinlichkeit eines Satzes, welche sich aus der Summe der Wahrscheinlichkeiten aller Parsebaume fr diesen Satz ergibt.
   Der Viterbi berechnet den besten Parsebaum fr einen Satz.
   Beide speichern Unterbaum-Wahrscheinlichkeiten.
   Der Viterbi, im Gegensatz zum Inside-Algorithmus, speichert außerdem die angewendeten Regeln und somit den Pfad zur Rückverfolgung und zum Bau des Parsebaums.

   b) Welchen Vorteil bieten diese Algorithmen gegenüber dem naiven Ansatz?

   Sie sind vor allem effizienter dadurch, dass sie die Unterbäume speichern und diese nicht neu berechnet werden müssen.

   c) Welche Rolle spielen die Variablen β, δ und ψ?

   β und δ und ψ sind Speichervariablen für die Unterbaum-Wahrscheinlichkeiten und -Regeln:
   β speichert die Wahrscheinlichkeit für einen Unterbaum im Inside-Alogirthmus;
   δ speichert die Wahrscheinlichkeit für den besten Parse-Unterbaum,
   und ψ enthält die angewendeten Regeln fr den jeweiligen (besten) Parsebaum im Viterbi.

2. **Unix-Tools**

   a) Was machen die unteren zwei Befehlsketten, wenn man sie auf einem deutschen Korpus laufen lässt? Beschreibe das Ergebnis mit geeigneten linguistischen Begriffen, und erkläre, wie es technisch zu Stande kommt.

      i. `tr -sc 'a-zA-Z' '\n' < KORPUS | tr 'A-Z' 'a-z' | grep 'heit$' | sort | uniq -c`

      Der erste "truncate"-Befehl tokenisiert den Korpus, der zweite wandelt alle Gro- in Kleinbuchstaben um. Mit "grep" werden dann alle Vorkommnisse von *heit* (Nomen, die auf *-heit* enden) erfasst. Anschließend wird mit "sort" und "uniq" gezählt, wieviele verschiedene solcher Nomen es gibt.

      ii. `tr -sc 'a-zA-Z' '\n' < KORPUS | tr 'A-Z' 'a-z' | grep 'heit$' | wc -l`

      Der Befehl "wc -l" zhalt, im Gegensatz zu "sort+uniq", alle Vorkommnisse oder Instanzen von der spezifizierten Nomensorte, und nicht wieviele verschiedene es gibt.

   b) Gib die Ausgabe des untenstehenden Befehls bei folgender Eingabe an.

      Eingabe:
      `abcd 1234 efgh`
      `this 8844 word`

      Befehlskette: `perl -pe 's/(a|o|u|i|e)(\w+)\s(\d+)\s/$2_$3_/g' Eingabe`

      bcd_1234_efgh
      ths_8844_word

3. **Naive-Bayes**

Wir betrachten Sätze mit einer NP-V-NP-PP-Struktur wie *"Peter baked the cake with friends"* oder *"Peter baked the cake with almonds"*.

In solchen Sätzen kann die Präpositionalphrase strukturell entweder an das vorausgehende Verb *(high attachment*, 1. Beispiel) oder an den zweiten Substantiv *(low attachment*, 2. Beispiel) angehängt werden.

In einem kleinen Korpus wurde jede Instanz einer solchen Ambiguität als *low-* oder *high-attachment* annotiert, sowie einige weitere Merkmale des Satzes festgestellt: das Verb, den 1. Substantiv des Satzes, die Präposition und den 2. Substantiv.

Instanz	Verb	N1	Präposition	N2	Attachment
1	baked	cake	with	almonds	low
2	saw	cake	with	almonds	low
3	saw	movie	with	friends	high
4	saw	movie	on	Tuesday	high

a) Gib in die A-Priori und die A-Posteriori-Wahrscheinlichkeiten an, die ein Naiv-Bayes-Klassifikator verwenden würde, um vorauszusagen, ob die PP an das Verb oder an die NP angehängt werden soll.

Input	Low Attachment	High Attachment
	a priori: low=2/4	high=2/4
Verb		
baked:	1/2	0/2
saw:	1/2	2/2
N1		
cake:	2/2	0/2
movie:	0/2	2/2
Präp		
with:	2/2	1/2
on:	0/2	1/2
N2		
almonds:	2/2	0/2
Tuesday:	0/2	1/2
friends:	0/2	1/2

b) Was sagt der Klassifikator für den Satz *"Peter baked the cake on Tuesday"* voraus?

$$
\begin{aligned}
c_{map} &= \text{argmax}_{c \in low, high}\ P(c) \cdot \prod_i P(a_i|c) \\
&= \text{argmax}\ (P(low) \cdot P(baked|low) \cdot P(cake|low) \cdot P(on|low) \\
&\qquad \cdot P(Tuesday|low), \\
&\qquad P(high) \cdot P(baked|high) \cdot P(cake|high) \cdot P(on|high) \\
&\qquad \cdot P(Tuesday|high)) \\
&= \text{argmax}\ (1/2 \cdot 1/2 \cdot 1 \cdot 0 \cdot 0 = 0, \\
&\qquad 1/2 \cdot 0 \cdot 0 \cdot 01/2 \cdot 1/2 = 0)
\end{aligned}
$$

Beide Attachment-Klassifikationen sind gleich (un-)wahrscheinlich.

c) Auf welches Problem stößt Du beim beantworten der vorigen Aufgabe und wie kann man es beheben?

Das Problem liegt darin, dass der Satz so noch nie gesehen worden ist und daher zunchst keine vernünftige Vorhersage mglich ist (da ungesehene Instanzen die relative Häufigkeit 0 haben). Durch Smoothing (zB durch das addieren kleiner Werte zu Zähler und Nenner kann man das Multiplizieren mit Null vermeiden.

4. **Informationstheorie**
Wir betrachten eine Sprache, die aus nur fünf Wörtern (Types) besteht. In einem kleinen Korpus (450 Tokens, 150 Sätze) werden die folgenden Häufigkeiten beobachtet:

Bill	Sue	kisses	pizza	likes
120	130	100	50	50

a) Bestimme die Entropie pro Wort für diese Sprache auf Basis der oberen Werten.

$$H(x) = -\sum_x p(x) \log_2 p(x) = 2.212$$

b) Aufgrund einer Korpusuntersuchung werde aber jetzt festgestellt, dass in der Sprache nur vier Sätze möglich sind:

"Bill kisses Sue" tritt 80 Mal auf.
"Sue kisses Bill" tritt 20 Mal auf.
"Bill likes pizza" tritt 20 Mal auf.
"Sue likes pizza" tritt 30 Mal auf.

Berechne erneut mit Hilfe dieser Angaben die durchschnittliche Entropie pro Wort (Hinweis: Betrachte jeden Satz als eine unteilbare Einheit). Erkläre den Unterschied zu dem Ergebnis aus 4a.

Sozusagen die "pro-Satz-Entropie" behandelt jeden Satz als eine Einheit und berechnet deren pro-Wort-Entropie mithilfe der angegebenen Satz-Häufgleiten: 1.723.
Um zur durchschnittlichen pro-Wort-Entropie innerhalb eines solchen Satzes zu gelangen, teilen wir den Wert durch 3. Das Ergebnis lautet: 0.574

c) Gib die pointwise mutual information für das Wortpaar $(kisses, Sue)$ an.

$$I(x,y) \quad = \log_2 \frac{P(x\ y)}{P(x)P(y)} = \log_2 \frac{80/150}{100/450 * 130/450} = 3.1519$$

5. **Noisy Channel Model**
Bestimme die maximale Kapazität eines binären symmetrischen Übertragungskanals, in dem jedes Symbol (0 und 1) mit einer Wahrscheinlichkeit von 0.15 vertauscht wird.

$$\begin{aligned} C &= 1 - H(p) \\ &= 1 - (0.15 \cdot \log_2 \tfrac{1}{0.15} + 0.85 \cdot \log_2 \tfrac{1}{0.85}) \\ &= 1 - (0.4105 + 0.1992) \\ &= 0.3903 \end{aligned}$$