Mathematische Grundlagen III

Ansätze bei Sparse Data

Garance PARIS

16. Juli 2011

Maximum Likelihood Estimation

- In einem Sprachmodell sind wir daran interessiert, die Wahrscheinlichkeit des nächsten Zeichens vorherzusagen
- Am einfachsten werden Wahrscheinlichkeiten an Hand der relativen Wahrscheinlichkeiten in einem Korpus geschätzt ("Maximum Likelihood Estimation"):

$$P(w) = \frac{\text{Häufigkeit}(w)}{\text{Größe des Korpus}}$$

N-Gramme

- N-Gramm: Wahrscheinlichkeit eines Wortes in Abhängigkeit vom vorhergehenden Kontext

  $P(w_n|w_1, \ldots, w_{n-1})$
- Idealerweise müsste man den gesamten Kontext berücksichtigen
- Aber:
  - Es ist unwahrscheinlich, dass wir dem selben Text/Satz schon einmal begegnet sind
  - Die Wahrscheinlichkeit der Folge wäre also gleich 0

N-Gramme

- Der Kontext wird also auf ca. 1–4 Wörter gekürzt:
  - $w_{n-1}w_n$: Bigramm
    (1 Wort vorausgehender Kontext)
  - $w_{n-2}w_{n-1}w_n$: Trigramm
    (2 Wörter vorausgehender Kontext)
  - usw.
- Je mehr Kontext, umso viele Parameter (=Wahrscheinlichkeiten) müssen geschätzt werden

  Zum Beispiel: $\ldots w_{n-3}\ w_{n-2}\ w_{n-1}\ w_n$

  Bei 5 Möglichkeiten für w_n, w_{n-1}, w_{n-2}, w_{n-3} sind es

  25 Parameter in einem 2-Gramm-Modell
  125 Parameter in einem 3-Gramm-Modell
  625 Parameter in einem 4-Gramm-Modell
  5^n Parameter in einem n-Gramm-Modell

Das Sparse Data-*Problem*

- Kleine Datenmengen, Fachjargon, neu entstehende Wörter, unbekannte Strukturen, ...
  führen alle zu **Sparse Data**:
  die geschätzten Parameter enthalten viele Null-Werte
- Außerdem ist MLE unzuverlässig für Zeichen,
  die selten auftreten

Das Gesetz von Zipf

The Principle of Least Effort

"People will act so as to minimise their probable average rate of work." (George Kinsley Zipf, 1949)

Anwendung auf natürliche Sprache

- Sprecher bevorzugen eine kleine Menge häufiger aber ambiger Wörter
- Hörer bevorzugen ein größeres Vokabular mit vielen eindeutigen Wörtern
- Der Kompromiss daraus ist maximal effizient

Das Gesetz von Zipf

Beobachtung

Korpora enthalten

- Eine sehr kleine Anzahl sehr häufiger Wörter
- Eine kleine bis mittelgroßer Anzahl Wörter mittlerer Häufigkeit
- Eine sehr große Anzahl niedrig frequenter Wörter

Rank	Word	Freq.	Rank	Word	Freq.
1	the	3332	200	turned	51
2	and	2972	300	you'll	30
3	a	1775	400	name	21
10	he	877	500	comes	16
20	but	410	600	group	13
30	be	294	700	lead	11
40	there	222	800	friends	10
50	one	172	900	begin	9
60	about	158	1000	family	8
70	more	138	2000	brushed	4
80	never	124	3000	sins	2
90	Oh	116	4000	could	2
100	two	104	8000	applausive	1

Tabelle: Worthäufigkeiten in *Tom Sawyer*, absteigend sortiert

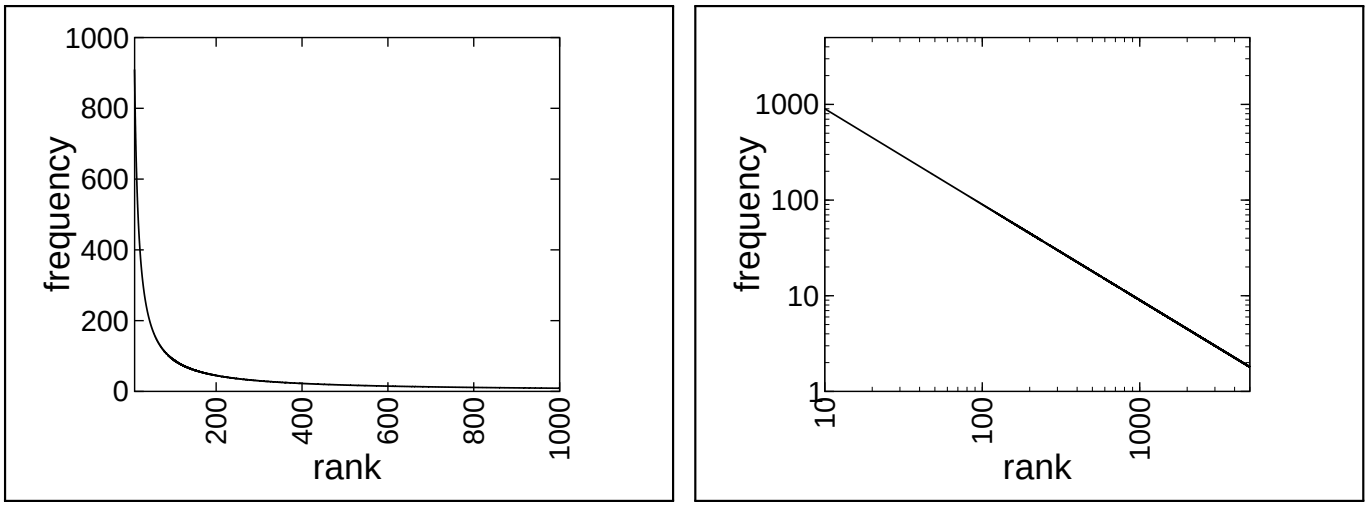

Abbildung: Diagramme der Worthäufigkeiten in *Tom Sawyer*

Für NLP bedeutet das: Sparse Data

- Eine kleine Anzahl Wörter treten häufig auf
- Aber die Mehrheit der Wörter treten in einem Korpus nur selten auf

Smoothing

- Häufigkeiten oder Wahrscheinlichkeiten seltener Ereignisse neu schätzen, so dass sie einen positiven Wert erhalten
- Begründung: Die an Hand des Training Korpus geschätzten Werte sind unzuverlässig bzw. entsprechen nicht der Sprache im Allgemeinen
- Die daraus resultierende Verteilung wird als “glatter” bezeichnet

Smoothing

Adding One

Intuition: Kleine Beträge zu den Null-Werten hinzufügen

Beispiel

$$P'(x) = \frac{f(x)+1/T}{N+1}$$

N: Größe des Korpus,
T: Anzahl Types

	f(x)	P(x)		P'(x)		
x_1	8	$\frac{8}{15}$	0.5333	$\frac{8+1/5}{15+1}$	0.5125	−
x_2	4	$\frac{4}{15}$	0.2667	$\frac{4+1/5}{15+1}$	0.2625	−
x_3	2	$\frac{2}{15}$	0.1333	$\frac{2+1/5}{15+1}$	0.1375	+
x_4	1	$\frac{1}{15}$	0.0667	$\frac{1+1/5}{15+1}$	0.075	+
x_5	0	$\frac{0}{15}$	0	$\frac{0+1/5}{15+1}$	0.0125	+
$\sum$	15		1		1	

Smoothing

Adding One

- Ein Teil der Wahrscheinlichkeitsmasse wird von Ereignissen mit hoher Wahrscheinlichkeit genommen und unter den kleinen Werten umverteilt
- Die Methode ist aber zu einfach und hat viele Nachteile, u. a. dass zuviel Wahrscheinlichkeitsmasse umverteilt wird
- Aber es gibt bessere Algorithmen, die nach dem selben Prinzip funktionnieren und Anwendung finden: Witten-Bell, Good-Turing, usw.

Smoothing

Alternative: Linearer Discounting

- Alle Ereignisse mit Wahrscheinlichkeit größer Null werden durch eine Konstante skaliert, die geringfügig kleiner 1 ist
- Die übrige Wahrscheinlichkeitsmasse wird umverteilt

$$P(w_1, \ldots, w_n) = \begin{cases} \frac{(1-\alpha)\, c(w_1,\ldots,w_n)}{N} & \text{if } c(w_1, \ldots, w_n) > 0 \\ \frac{\alpha}{N_0} & \text{otherwise.} \end{cases}$$

Smoothing

Class-Based Smoothing

Intuition:

- Ereignisse in Klassen gruppieren
- Wenn die Wahrscheinlichkeit des Ereignisses nicht verfügbar ist, stattdessen die Wahrscheinlichkeit der Klasse verwenden

Class-Based Smoothing

Beispiel

"You shall know a word by the company it keeps"
(Zellig Harris)

Die Bedeutung eines Wortes kann an Hand seiner Kookurrenzhäufigkeit mit anderen Wörtern dargestellt werden

..... is sold in liter bottles.
..... is a popular drink in Russia.
..... makes you drunk quickly.

$$w_i =< |w_1|, |w_2|, \ldots, |w_n| >$$

Class-Based Smoothing

Clustering mit Mutual Information

- Um ähnliche Wörter zu Klassen zusammenzufügen, wird ein Ähnlichkeitsmaß gebraucht
- Euklidische Distanz macht einen großen Unterschied zwischen hochfrequenten und niedrigfrequenten Wörtern
- Mutual Information kann verwendet werden:
  z. B.
  - “big” und “large” treten häufig mit denselben Wörtern auf
  - I(big;large) ist hoch

Class-Based Smoothing

Clustering mit Mutual Information

Beispiel für die erstellten Klassen:

- Friday Monday Thursday Wednesday Tuesday Saturday Sunday weekends Sundays Saturdays (days)
- People guys folks fellows CEOs chaps doubters commies unfortunates blokes (humans)
- down backwards ashore sideways southward northward overboard aloft downwards adrift (directional nouns)
- water gas coal liquid acid sand carbon steam shale iron (substances)
- had hadn't hath would've could've should've must've might've
- that tha theat (definite article, including misspellings)
- Head body hands eyes voice arm seat eye hair mouth (body parts)

Class-Based Smoothing

Modellvergleich

Es können mehrere Modelle jetzt verglichen werden

- Ein übliches Bigramm Modell
- Ein Modell, in dem Klassen anstatt Wörter verwendet werden
- Ein Modell, in dem Klassen und Wörter kombiniert werden

Dafür verwenden wir Cross-Entropy und finden heraus, dass das 3. Modell am besten abschneidet:

- Cross-Entropy: 7.93
- Cross-Entropy: 7.88
- Cross-Entropy: 8.08