3: Statistical Properties of Language
Machine Learning and Real-world Data (MLRD)

Simone Teufel

Last session: You implemented a Naive Bayes classifier

- Smoothed vs Unsmoothed
- The accuracy of the un-smoothed classifier was seriously affected by unseen words.
- We implemented add-one (Laplace) smoothing:

$$\hat{P}(w_i|c) = \frac{count(w_i, c) + 1}{\sum_{w \in V}(count(w, c) + 1)} = \frac{count(w_i, c) + 1}{(\sum_{w \in V} count(w, c)) + |V|}$$

- Smoothing helped!

Today: frequency distributions in language

Questions:

- Why did smoothing help? (or in other words:)
- What is it about the distribution of words in a language that affected the performance of the un-smoothed classifier?
- Two Laws: Zipf's Law and Heap's Law

Zipf's Law: Word frequency distributions obey a power law

- There are a small number of very high-frequency words
- There are a large number of low-frequency words
- Zipf's law: the nth most frequent word has a frequency proportional to $1/n$

"a word's frequency in a corpus is inversely proportional to its rank"

The parameters of Zipf's law are language-dependent

Zipf's law:

$$f_w \approx \frac{k}{r_w{}^\alpha}$$

where

f_w: frequency of word w

r_w: frequency rank of word w

α, k: constants (which vary with the language)

e.g. α is around 1 for English but 1.3 for German

The parameters of Zipf's law are language-dependent

Actually...

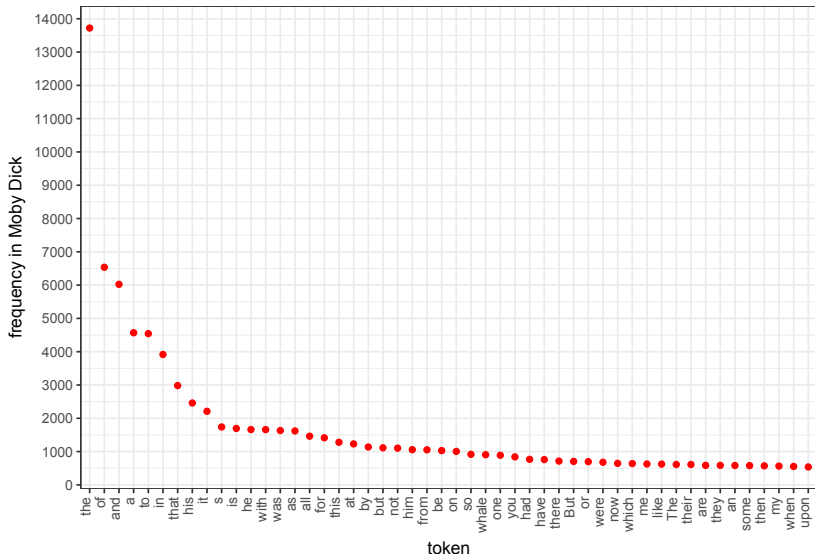$$f_w \approx \frac{k}{(r_w + \beta)^\alpha}$$

where

β: a shift in the rank

see summary paper by Piantadosi
`https://link.springer.com/article/10.3758/`
`s13423-014-0585-6`

we won't worry about the rank-shift today

There are a small number of high-frequency words...

Moby Dick has 206,052 words in total.

Similar sorts of high-frequency words across languages

Top 10 most frequent words in some large language samples:

Similar sorts of high-frequency words across languages

Top 10 most frequent words in some large language samples:

English

1	the	6,187,267
2	of	2,941,444
3	and	2,682,863
4	a	2,126,369
5	in	1,812,609
6	to	1,620,850
7	it	1,089,186
8	is	998,389
9	was	923,948
10	to	917,579

BNC,
100Mw

Similar sorts of high-frequency words across languages

Top 10 most frequent words in some large language samples:

	English			German	
1	the	6,187,267	1	der	7,377,879
2	of	2,941,444	2	die	7,036,092
3	and	2,682,863	3	und	4,813,169
4	a	2,126,369	4	in	3,768,565
5	in	1,812,609	5	den	2,717,150
6	to	1,620,850	6	von	2,250,642
7	it	1,089,186	7	zu	1,992,268
8	is	998,389	8	das	1,983,589
9	was	923,948	9	mit	1,878,243
10	to	917,579	10	sich	1,680,106

BNC, 100Mw

"Deutscher Wortschatz", 500Mw

Similar sorts of high-frequency words across languages

Top 10 most frequent words in some large language samples:

	English			German			Spanish	
1	the	6,187,267	1	der	7,377,879	1	que	32,894
2	of	2,941,444	2	die	7,036,092	2	de	32,116
3	and	2,682,863	3	und	4,813,169	3	no	29,897
4	a	2,126,369	4	in	3,768,565	4	a	22,313
5	in	1,812,609	5	den	2,717,150	5	la	21,127
6	to	1,620,850	6	von	2,250,642	6	el	18,112
7	it	1,089,186	7	zu	1,992,268	7	es	16,620
8	is	998,389	8	das	1,983,589	8	y	15,743
9	was	923,948	9	mit	1,878,243	9	en	15,303
10	to	917,579	10	sich	1,680,106	10	lo	14,010

BNC, 100Mw

"Deutscher Wortschatz", 500Mw

subtitles, 27.4Mw

Similar sorts of high-frequency words across languages

Top 10 most frequent words in some large language samples:

	English			German			Spanish			Italian	
1	the	6,187,267	1	der	7,377,879	1	que	32,894	1	non	25,757
2	of	2,941,444	2	die	7,036,092	2	de	32,116	2	di	22,868
3	and	2,682,863	3	und	4,813,169	3	no	29,897	3	che	22,738
4	a	2,126,369	4	in	3,768,565	4	a	22,313	4	è	18,624
5	in	1,812,609	5	den	2,717,150	5	la	21,127	5	e	17,600
6	to	1,620,850	6	von	2,250,642	6	el	18,112	6	la	16,404
7	it	1,089,186	7	zu	1,992,268	7	es	16,620	7	il	14,765
8	is	998,389	8	das	1,983,589	8	y	15,743	8	un	14,460
9	was	923,948	9	mit	1,878,243	9	en	15,303	9	a	13,915
10	to	917,579	10	sich	1,680,106	10	lo	14,010	10	per	10,501

BNC, 100Mw

"Deutscher Wortschatz", 500Mw

subtitles, 27.4Mw

subtitles, 5.6Mw

Similar sorts of high-frequency words across languages

Top 10 most frequent words in some large language samples:

English		German		Spanish		Italian		Dutch	
1 the	6,187,267	1 der	7,377,879	1 que	32,894	1 non	25,757	1 de	4,770
2 of	2,941,444	2 die	7,036,092	2 de	32,116	2 di	22,868	2 en	2,709
3 and	2,682,863	3 und	4,813,169	3 no	29,897	3 che	22,738	3 het/'t	2,469
4 a	2,126,369	4 in	3,768,565	4 a	22,313	4 è	18,624	4 van	2,259
5 in	1,812,609	5 den	2,717,150	5 la	21,127	5 e	17,600	5 ik	1,999
6 to	1,620,850	6 von	2,250,642	6 el	18,112	6 la	16,404	6 te	1,935
7 it	1,089,186	7 zu	1,992,268	7 es	16,620	7 il	14,765	7 dat	1,875
8 is	998,389	8 das	1,983,589	8 y	15,743	8 un	14,460	8 die	1,807
9 was	923,948	9 mit	1,878,243	9 en	15,303	9 a	13,915	9 in	1,639
10 to	917,579	10 sich	1,680,106	10 lo	14,010	10 per	10,501	10 een	1,637

BNC, 100Mw

"Deutscher Wortschatz", 500Mw

subtitles, 27.4Mw

subtitles, 5.6Mw

subtitles, 800Kw

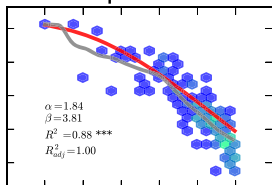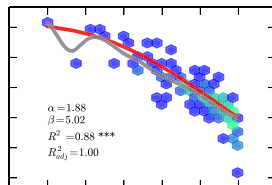# It is helpful to plot Zipf curves in log-space

Reuters dataset: taken from `https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf` – chapter 5
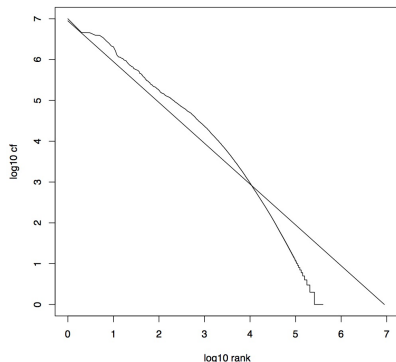
By fitting a simple line to the data in log-space we can estimate the language specific parameters α and k (we will do this today!)

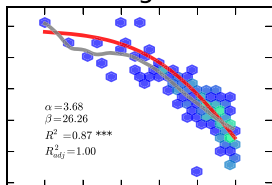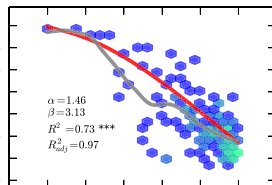# In log-space we can more easily estimate the language specific parameters

From Piantadosi `https://link.springer.com/article/10.3758/s13423-014-0585-6`

Zipfian (or near-Zipfian) distributions occur in many collections

- Sizes of settlements
- Frequency of access to web pages
- Size of earthquakes
- Word senses per word
- Notes in musical performances
- machine instructions
- . . .

Zipfian (or near-Zipfian) distributions occur in many collections

world city populations for 8 countries
log-size vs log-rank
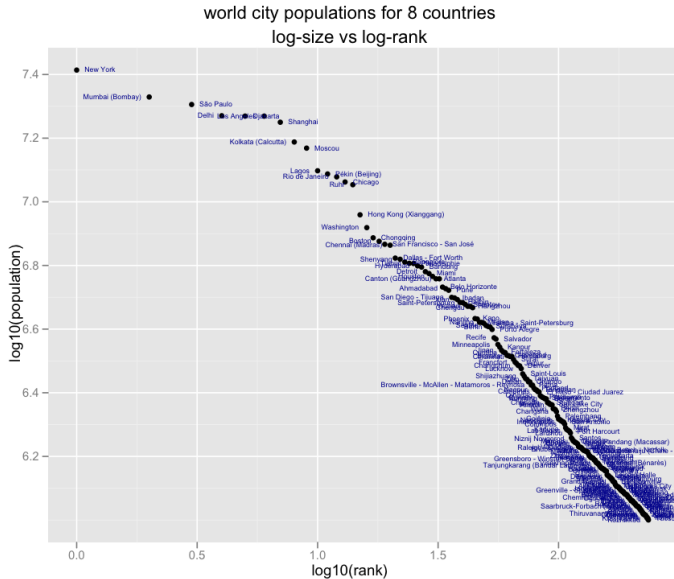
There is also a relationship between vocabulary size and text length

So far we have been thinking about frequencies of particular words:

- we call any unique word a type: *the* is a word type
- we call an instance of a type a token: there are 13721 *the* tokens in Moby Dick
- the number of types in a text is the size of the vocabulary (also called dictionary)

Today you will also explore this relationship.

Heaps' law describes the relationship between vocabulary and text-length

Heaps' Law:

The relationship between the size of a vocabulary and the size of text that gave rise to it is

$$u_n = kn^{\beta}$$

where

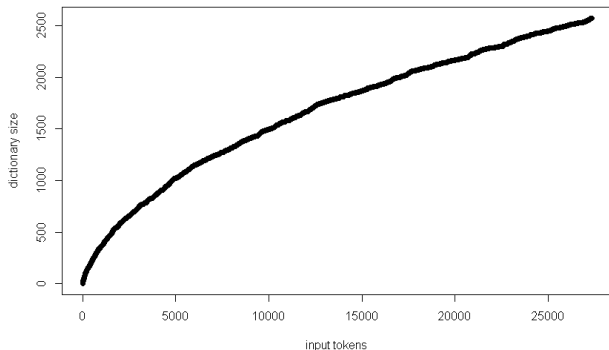u_n: number of types (unique items), i.e. vocabulary size

n: total number of tokens, i.e. text size

β, k: constants (language-dependent)
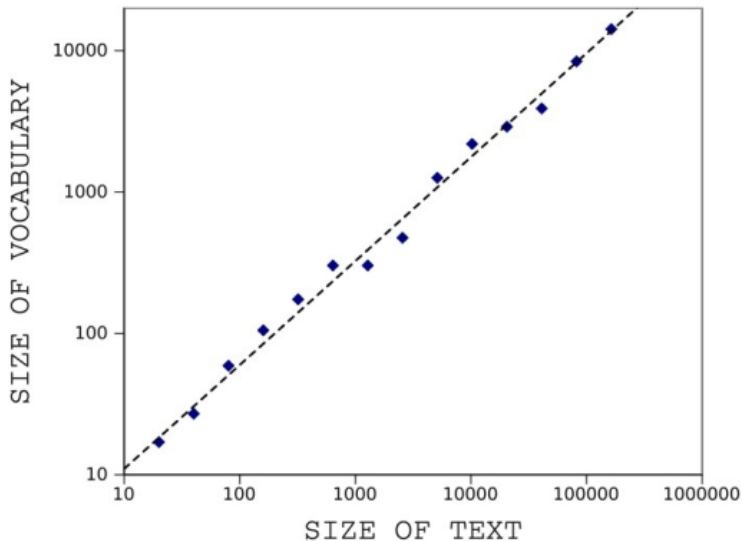
β is around $\frac{1}{2}$

$30 \leq k \leq 100$

Heaps' Law

- No saturation: there will always be more new types
- As we progress through a text it takes longer and longer to encounter a new type
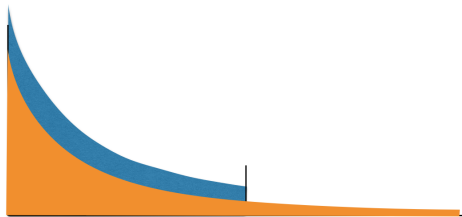
It is helpful to plot Heaps' law in log-space

Zipf's law and Heaps' law affected our classifier

- The Zipfian curve has a lot of probability mass in the long tail.
- By Heaps' law, we need increasing amounts of text to see new word types in the tail
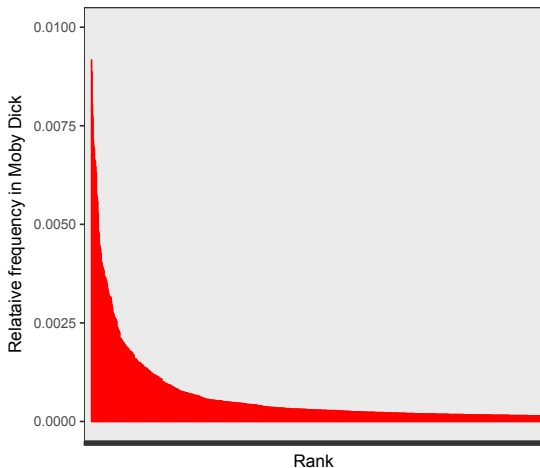
Zipf's law and Heaps' law affected our classifier

- With MLE, only seen types receive a probability estimate:

e.g. we used:

$$\hat{P}_{MLE}(w_i|c) = \frac{count(w_i, c)}{\sum_{w \in V_{training}} count(w, c)}$$

- True probability (e.g. for NEG class): orange; MLE: blue
- Total probabilities must sum to 1; in MLE all that probability mass is given to seen types
- MLE overestimates the probability of seen types (as opposed to unseen)

Smoothing redistributes the probability mass

- Add-one smoothing redistributes the probability mass.

e.g. we used:

$$\hat{P}(w_i|c) = \frac{count(w_i, c) + 1}{\sum_{w \in V}(count(w, c) + 1)} = \frac{count(w_i, c) + 1}{(\sum_{w \in V} count(w, c)) + |V|}$$

- It takes some portion away from the MLE overestimate.
- It redistributes this portion to the unseen types.
- Better estimate; still not perfect.

Today we will investigate Zipf's and Heaps' law in movie reviews

Follow task instructions on moodle to:

- Plot a frequency vs rank graph for larger set of movie reviews (you are given chart plotting code)
- Plot a log frequency vs log rank graph
- Indicate the location of your 10 chosen words from Tick 1, e.g. in colour, on this plot.
- Use least-squares algorithm to fit a line to the log-log plot (you are given `best-fit` code)
- Estimate the parameters of the Zipf equation
- Plot type vs token graph for the movie reviews

Ticking for Task 3

There is no automatic ticker for Task 3

- Write everything in your lab book
- Save all your graphs (as screenshots or otherwise)