Language Models

Natural Language Processing
CS 4120/6120—Spring 2017
Northeastern University

David Smith
PLUTARCH'S LIVES.

Dionysius, both of them Colophonians, with all their nerves and strength one finds in them, appear to be too much labored, and smell too much of the lamp; whereas the paintings of Nicomachus† and the verses of Homer, beside their other excellencies and graces, seem to have some other excellencies also more effectually decomposing the iron ore. The same quantity of fuel, applied at once to the furnace, would only prolong the duration of its heat, not augment its intensity.
Predicting Language
Predicting Language

A SMALL OBLONG READING LAMP ON THE DESK
Predicting Language

A SMALL OBLONG READING LAMP ON THE DESK

--SM----OBL----REA----------O-------D---
Predicting Language

A SMALL OBLONG READING LAMP ON THE DESK

What informs this prediction?
Predicting Language

- Optical character recognition
- Automatic speech recognition
- Machine translation
- Spelling/grammar correction
- Restoring redacted texts
Scoring Language

- Language identification
- Text categorization
- Grading essays (!)
- Information retrieval
in the seventh heavens. Elsewhere match that bloom of theirs, ye cannot, s
ey all stand before me; and I their match. Oh, hard! that to fire others,
h, hard! that to fire others, the match itself must needs be wasting! What
so sweet on earth -- heaven may not match it! -- as those swift glances of war
end; but hardly had he ignited his match across the rough sandpaper of his ha
utting the lashing of the waterproof match keg, after many failures Starbuck c
asks heaped up in him and the slow-match silently burning along towards them
followed by Stubb's producing his match and igniting his pipe, for now a re
aspect, Pip and Dough-Boy made a match, like a black pony and a white one

is regarded her disapprobation of the match. Mr. John Dashwood told his mother
ced of it. It would be an excellent match, for HE was rich, and SHE was hand
you have any reason to expect such a match." "Don't pretend to deny it, be
ry much. But mama did not think the match good enough for me, otherwise Sir J
on't we all know that it must be a match, that they were over head and ears
ght. It will be all to one a better match for your sister. Two thousand a yea
the other an account of the intended match, in a voice so little attempting co
end of a week that it would not be a match at all. The good understanding betw
with you and your family. It is a match that must give universal satisfactio
le on him a thousand a year, if the match takes place. The lady is the Hon.
before, that she thought to make a match between Edward and some Lord's dau
e, with all my heart, it will be a match in spite of her. Lord! what a taki
certain penury that must attend the match. His own two thousand pounds she pr
man nature. When Edward's unhappy match takes place, depend upon it his mot
m myself, and dissuade him from the match; but it was too late THEN, I found
Language Models

- Probability distribution over strings of text
- There may be hidden variables
  - Grammatical structure, topics, NN state
- Hidden variables may perform classification
Probability
Axioms of Probability

• Define event space  \( \bigcup_i \mathcal{F}_i = \Omega \)

• Probability function, s.t.  \( P: \mathcal{F} \rightarrow [0, 1] \)

• Disjoint events sum  \( A \cap B = \emptyset \Leftrightarrow P(A \cup B) = P(A) + P(B) \)

• All events sum to one  \( P(\Omega) = 1 \)

• Show that:  \( P(\bar{A}) = 1 - P(A) \)
Conditional Probability

\[ P(A \mid B) = \frac{P(A, B)}{P(B)} \]

\[ P(A, B) = P(B)P(A \mid B) = P(A)P(B \mid A) \]

\[ P(A_1, A_2, \ldots, A_n) = P(A_1)P(A_2 \mid A_1)P(A_3 \mid A_1, A_2) \cdots P(A_n \mid A_1, \ldots, A_{n-1}) \]

*Chain rule*
Independence

\[ P(A, B) = P(A)P(B) \]
\[ \iff \]
\[ P(A \mid B) = P(A) \quad \land \quad P(B \mid A) = P(B) \]

In coding terms, knowing \( B \) doesn’t help in decoding \( A \), and vice versa.
Markov Models

\[ p(w_1, w_2, \ldots, w_n) = p(w_1)p(w_2 | w_1)p(w_3 | w_1, w_2) \]
\[ p(w_4 | w_1, w_2, w_3) \cdots p(w_n | p_1, \ldots, p_{n-1}) \]
Markov Models

\[ p(w_1, w_2, \ldots, w_n) = p(w_1)p(w_2 | w_1)p(w_3 | w_1, w_2) \]
\[ p(w_4 | w_1, w_2, w_3) \cdots p(w_n | p_1, \ldots, p_{n-1}) \]

Markov independence assumption

\[ p(w_i | w_1, \ldots w_{i-1}) \approx p(w_i | w_{i-1}) \]
Markov Models

\[ p(w_1, w_2, \ldots, w_n) = p(w_1)p(w_2 \mid w_1)p(w_3 \mid w_1, w_2) \]
\[ p(w_4 \mid w_1, w_2, w_3) \cdots p(w_n \mid p_1, \ldots, p_{n-1}) \]

Markov independence assumption

\[ p(w_i \mid w_1, \ldots w_{i-1}) \approx p(w_i \mid w_{i-1}) \]

\[ p(w_1, w_2, \ldots, w_n) \approx p(w_1)p(w_2 \mid w_1)p(w_3 \mid w_2) \]
\[ p(w_4 \mid w_3) \cdots p(w_n \mid p_{n-1}) \]
Another View

Bigram model as (dynamic) Bayes net

Trigram model as (dynamic) Bayes net
Another View

Bigram model as (dynamic) Bayes net

Trigram model as (dynamic) Bayes net
Another View

Bigram model as (dynamic) Bayes net

W₁ \rightarrow \text{The} \rightarrow W₂ \rightarrow W₃ \rightarrow W₄

Trigram model as (dynamic) Bayes net
Another View

Bigram model as (dynamic) Bayes net

Trigram model as (dynamic) Bayes net
Another View

The results

Bigram model as (dynamic) Bayes net

W₁

The

W₂

results

p(w₂|The)

W₃

p(w₃|results)

W₄

Trigram model as (dynamic) Bayes net
Another View

Bigram model as (dynamic) Bayes net

Trigram model as (dynamic) Bayes net
Another View

The results have

Bigram model as (dynamic) Bayes net

Trigram model as (dynamic) Bayes net
The results have shown that:\n\[ p(w_2|\text{The}) \quad p(w_3|\text{results}) \quad p(w_4|\text{have}) \]

Bigram model as (dynamic) Bayes net

Trigram model as (dynamic) Bayes net
The results have shown

Bigram model as (dynamic) Bayes net

Trigram model as (dynamic) Bayes net
Another View

Directed graphical models: *lack* of edge means conditional independence

Bigram model as (dynamic) Bayes net

Trigram model as (dynamic) Bayes net
Another View

Directed graphical models: lack of edge means conditional independence

The results have shown

Bigram model as (dynamic) Bayes net

Trigram model as (dynamic) Bayes net
The results have shown that:

1. Directed graphical models: *lack* of edge means conditional independence.
2. Bigram model as (dynamic) Bayes net.
3. Trigram model as (dynamic) Bayes net.
The results have shown that $p(w_2|\text{The})$, $p(w_3|\text{results})$, $p(w_4|\text{have})$, and $p(w_5|\text{shown})$.

Directed graphical models: lack of edge means conditional independence.

**Bigram model as (dynamic) Bayes net**

**Trigram model as (dynamic) Bayes net**
Yet Another View

Bigram model as finite state machine

What about a trigram model?
Classifiers:
Language under Different Conditions
Movie Reviews
there's some movies i enjoy even though i know i probably shouldn't and have a difficult time trying to explain why i did. "lucky numbers" is a perfect example of this because it's such a blatant rip-off of "fargo" and every movie based on an elmore leonard novel and yet it somehow still works for me. i know i'm in the minority here but let me explain. the film takes place in harrisburg, pa in 1988 during an unseasonably warm winter. ...
Movie Reviews

there's some movies i enjoy even though i know i probably shouldn't and have a difficult time trying to explain why i did. "lucky numbers" is a perfect example of this because it's such a blatant rip-off of "fargo" and every movie based on an elmore leonard novel and yet it somehow still works for me. i know i'm in the minority here but let me explain. the film takes place in harrisburg, pa in 1988 during an unseasonably warm winter. ...
there's some movies i enjoy even though i know i probably shouldn't and have a difficult time trying to explain why i did. "lucky numbers" is a perfect example of this because it's such a blatant rip-off of "fargo" and every movie based on an elmore leonard novel and yet it somehow still works for me. i know i'm in the minority here but let me explain. the film takes place in harrisburg, pa in 1988 during an unseasonably warm winter.

seen at: amc old pasadena 8, pasadena, ca (in sdds)
paul verhoeven's last movie, showgirls, had a bad script, bad acting, and a "plot" (i use the word in its loosest possible sense) that served only to allow lots of sex and nudity. it stank. starship troopers has a bad script, bad acting, and a "plot" that serves only to allow lots of violence and gore. it stinks. nobody will watch this movie for the plot, ...
there's some movies i enjoy even though i know i probably shouldn't and have a difficult time trying to explain why i did. "lucky numbers" is a perfect example of this because it's such a blatant rip-off of "fargo" and every movie based on an elmore leonard novel and yet it somehow still works for me. i know i'm in the minority here but let me explain. the film takes place in harrisburg, pa in 1988 during an unseasonably warm winter.

seen at: amc old pasadena 8, pasadena, ca (in sdds) paul verhoeven's last movie, showgirls, had a bad script, bad acting, and a "plot" (i use the word in its loosest possible sense) that served only to allow lots of sex and nudity. it stank. starship troopers has a bad script, bad acting, and a "plot" that serves only to allow lots of violence and gore. it stinks. nobody will watch this movie for the plot.
there's some movies i enjoy even though i know i probably shouldn't and have a difficult time trying to explain why i did. "lucky numbers" is a perfect example of this because it's such a blatant rip-off of "fargo" and every movie based on an elmore leonard novel and yet it somehow still works for me. i know i'm in the minority here but let me explain. the film takes place in harrisburg, pa in 1988 during an unseasonably warm winter.

seen at: amc old pasadena 8, pasadena, ca (in sdds) paul verhoeven's last movie, showgirls, had a bad script, bad acting, and a "plot" (i use the word in its loosest possible sense) that served only to allow lots of sex and nudity. it stank. starship troopers has a bad script, bad acting, and a "plot" that serves only to allow lots of violence and gore. it stinks. nobody will watch this movie for the plot,...

the rich legacy of cinema has left us with certain indelible images. the tinkling christmas tree bell in "it's a wonderful life." bogie's speech at the airport in "casablanca." little elliott's flying bicycle, silhouetted by the moon in "e.t." and now," starship troopers" director paul verhoeven adds one more image that will live in our memories forever: doogie houser doing a vulcan mind meld with a giant slug." starship troopers," loosely based on
there ' s some movies i enjoy even though i know i probably shouldn ' t and have a difficult time trying to explain why i did . " lucky numbers " is a perfect example of this because it ' s such a blatant rip - off of " fargo " and every movie based on an elmore leonard novel and yet it somehow still works for me . i know i ' m in the minority here but let me explain . the film takes place in harrisburg , pa in 1988 during an unseasonably warm winter . .

seen at : amc old pasadena 8 , pasadena , ca ( in sdds ) paul verhoeven ' s last movie , showgirls , had a bad script , bad acting , and a " plot " ( i use the word in its loosest possible sense ) that served only to allow lots of sex and nudity . it stank . starship troopers has a bad script , bad acting , and a " plot " that serves only to allow lots of violence and gore . it stinks . nobody will watch this movie for the plot , . .

the rich legacy of cinema has left us with certain indelible images . the tinkling christmas tree bell in " it ' s a wonderful life . " bogie ' s speech at the airport in " casablanca . " little elliott ' s flying bicycle , silhouetted by the moon in " e . t . " and now , " starship troopers " director paul verhoeven adds one more image that will live in our memories forever : doogie houser doing a vulcan mind meld with a giant slug . " starship troopers , " loosely based on
Setting up a Classifier
Setting up a Classifier

- What we want:

\[ p(😊 \mid w_1, w_2, \ldots, w_n) > p(<Result> \mid w_1, w_2, \ldots, w_n) \]
Setting up a Classifier

• What we want:

\[ p(☺ \mid w_1, w_2, \ldots, w_n) > p(☹ \mid w_1, w_2, \ldots, w_n) \] ?

• What we know how to build:
Setting up a Classifier

• What we want:

\[ p(😊 | w_1, w_2, \ldots, w_n) > p(😃 | w_1, w_2, \ldots, w_n) \] ?

• What we know how to build:

  • A language model for each class
Setting up a Classifier

• What we want:
  \[ p(😊 | w_1, w_2, ..., w_n) > p(🙂 | w_1, w_2, ..., w_n) \]

• What we know how to build:
  • A language model for each class
  • \[ p(w_1, w_2, ..., w_n | 😊) \]
Setting up a Classifier

• What we want:

\[ p(😊 | w_1, w_2, ..., w_n) > p(😢 | w_1, w_2, ..., w_n) \]

• What we know how to build:

• A language model for each class

• \[ p(w_1, w_2, ..., w_n | 😊) \]

• \[ p(w_1, w_2, ..., w_n | 😢) \]
By the definition of conditional probability:

\[ P(A, B) = P(B)P(A \mid B) = P(A)P(B \mid A) \]

we can show:

\[ P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \]

Seemingly trivial result from 1763; interesting consequences...
A “Bayesian” Classifier

\[ p(R \mid w_1, w_2, \ldots, w_n) = \frac{p(R)p(w_1, w_2, \ldots, w_n \mid R)}{p(w_1, w_2, \ldots, w_n)} \]

\[ \max_{R \in \{\sim, \bar{\sim}\}} p(R \mid w_1, w_2, \ldots, w_n) = \max_{R \in \{\sim, \bar{\sim}\}} p(R)p(w_1, w_2, \ldots, w_n \mid R) \]
A “Bayesian” Classifier

Nowadays also means modeling uncertainty about \( p \)

\[
p(R \mid w_1, w_2, \ldots, w_n) = \frac{p(R)p(w_1, w_2, \ldots, w_n \mid R)}{p(w_1, w_2, \ldots, w_n)}
\]

\[
\max_{R \in \{\circlearrowleft, \circlearrowright\}} p(R \mid w_1, w_2, \ldots, w_n) = \max_{R \in \{\circlearrowleft, \circlearrowright\}} p(R)p(w_1, w_2, \ldots, w_n \mid R)
\]
Naive Bayes Classifier

No dependencies among words!
NB on Movie Reviews

- Train models for positive, negative
- For each review, find higher posterior
- Which word probability ratios are highest?

```python
>>> classifier.show_most_informative_features(5)

classifier.show_most_informative_features(5)
Most Informative Features
contains(outstanding) = True            pos : neg    =     14.1 : 1.0
contains(mulan) = True                 pos : neg    =      8.3 : 1.0
contains(seagal) = True                neg : pos    =      7.8 : 1.0
contains(wonderfully) = True           pos : neg    =      6.6 : 1.0
contains(damon) = True                 pos : neg    =      6.1 : 1.0
```
What’s Wrong With NB?

• What happens when word dependencies are strong?

• What happens when some words occur only once?

• What happens when the classifier sees a new word?
LMs in IR

• Three possibilities:
  • probability of generating the query text from a document language model
  • probability of generating the document text from a query language model
  • comparing the language models representing the query and document topics
Query Likelihood in IR

• Rank documents by the probability that the query could be generated by language model estimated from that document

• Given user query, start with $p(D \mid Q)$

• Using Bayes’ Rule

$$p(D \mid Q)^{\text{rank}} = p(Q \mid D)P(D)$$

• Assuming prior is uniform, use unigram LM

$$p(Q \mid D) = \prod_{i=1}^{n} p(q_i \mid D)$$
Codes and Entropy
Codes Again

- How much information is conveyed in language?
- How uncertain is a classifier?
- How short of a message do we need to send to communicate given information?
- Basic idea of compression: common data elements use short codes while uncommon data elements use longer codes
Compression and Entropy

- **Entropy** measures “randomness”
- Inverse of compressability

\[ H(X) = - \sum_{i=1}^{n} p(X = x_i) \log p(X = x_i) \]

- Lg (base 2): measured in *bits*
- Upper bound: \( \lg n \)
- Example curve for binomial
Compression and Entropy

- Entropy bounds compression rate
  - Theorem: $H(X) \leq E[|\text{encoded}(X)|]$ 
  - Recall: $H(X) \leq \lg n$
  - $n$ is the size of the domain of $X$
- Standard binary encoding of integers optimizes for the worst case
- With knowledge of $p(X)$, we can do better:
  - $H(X) \leq E[|\text{encoded}(X)|] < H(X) + 1$
- Bound achieved by *Huffman codes*
Predicting Language

A small oblong reading lamp on the desk

What informs this prediction?
Predicting Language

Fig. 2—Communication system using reduced text.

Predicting Language

There is no reverse on a motorcycle.
The Shannon Game
http://www.ccs.neu.edu/course/cs6120sp17/shannon/

Results to
dasmith@ccs.neu.edu
Fig. 4—Upper and lower experimental bounds for the entropy of 27-letter English.
Estimation
Simple Estimation

- Probability courses usually start with equiprobable events
- Coin flips, dice, cards
- How likely to get a 6 rolling 1 die?
- How likely the sum of two dice is 6?
- How likely to see 3 heads in 10 flips?
Binomial Distribution

For $n$ trials, $k$ successes, and success probability $p$:

$$P(k) = \binom{n}{k} p^k (1 - p)^{n-k}$$  \hspace{1cm} \text{Prob. mass function}

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Estimation problem: If we observe $n$ and $k$, what is $p$?
Maximum Likelihood

Say we win 40 games out of 100.

\[ P(40) = \binom{100}{40} p^{40} (1 - p)^{60} \]

The maximum likelihood estimator for \( p \) solves:

\[ \max_P P(\text{observed data}) = \max_P \binom{100}{40} p^{40} (1 - p)^{60} \]
Maximum Likelihood

Likelihood of 40/100 wins

P(40)
Maximum Likelihood

How to solve

$$\max_p \binom{100}{40} p^{40} (1 - p)^{60}$$
How to solve \[ \max_p \left( \binom{100}{40} p^{40} (1 - p)^{60} \right) \]

\[
0 \; = \; \frac{\partial}{\partial p} \left( \binom{100}{40} p^{40} (1 - p)^{60} \right) \\
= \; 40p^{39} (1 - p)^{60} - 60p^{40} (1 - p)^{59} \\
= \; p^{39} (1 - p)^{59} [40(1 - p) - 60p] \\
= \; p^{39} (1 - p)^{59} 40 - 100p \]
Maximum Likelihood

How to solve

$$\max_p \binom{100}{40} p^{40} (1 - p)^{60}$$

$$0 = \frac{\partial}{\partial p} \binom{100}{40} p^{40} (1 - p)^{60}$$

$$= 40p^{39}(1 - p)^{60} - 60p^{40}(1 - p)^{59}$$

$$= p^{39}(1 - p)^{59}[40(1 - p) - 60p]$$

$$= p^{39}(1 - p)^{59} 40 - 100p$$

Solutions: 0, 1, .4
Maximum Likelihood

How to solve

\[ \max_p \binom{100}{40} p^{40} (1 - p)^{60} \]

\[
0 = \frac{\partial}{\partial p} \binom{100}{40} p^{40} (1 - p)^{60} \\
= 40p^{39}(1 - p)^{60} - 60p^{40}(1 - p)^{59} \\
= p^{39}(1 - p)^{59} [40(1 - p) - 60p] \\
= p^{39}(1 - p)^{59} 40 - 100p
\]

Solutions: 0, 1, .4

The maximizer!
Maximum Likelihood

How to solve

$$\max_p \binom{100}{40} p^{40} (1 - p)^{60}$$

$$0 = \frac{\partial}{\partial p} \binom{100}{40} p^{40} (1 - p)^{60}$$

$$= 40p^{39} (1 - p)^{60} - 60p^{40} (1 - p)^{59}$$

$$= p^{39} (1 - p)^{59} [40(1 - p) - 60p]$$

$$= p^{39} (1 - p)^{59} 40 - 100p$$

In general, $k/n$

Solutions: 0, 1, .4

The maximizer!
Maximum Likelihood

How to solve

$$\max_p \left( \frac{100}{40} \right) p^{40} (1 - p)^{60}$$

0  =  \frac{\partial}{\partial p} \left( \frac{100}{40} \right) p^{40} (1 - p)^{60}

=  40 p^{39} (1 - p)^{60} - 60 p^{40} (1 - p)^{59}

=  p^{39} (1 - p)^{59} [40(1 - p) - 60p]

=  p^{39} (1 - p)^{59} 40 - 100p

In general, $k/n$

Solutions: 0, 1, .4

This is trivial here, but a widely useful approach.
ML for Language Models

• Say the corpus has “in the” 100 times
• If we see “in the beginning” 5 times,
  \[ p_{ML}(\text{beginning} \mid \text{in the}) = ? \]
• If we see “in the end” 8 times,
  \[ p_{ML}(\text{end} \mid \text{in the}) = ? \]
• If we see “in the kitchen” 0 times,
  \[ p_{ML}(\text{kitchen} \mid \text{in the}) = ? \]
ML for Naive Bayes

• Recall: \( p(+) \mid \text{Damon movie} \)
  
  \[ = p(\text{Damon} \mid +) \cdot p(\text{movie} \mid +) \cdot p(+) \]

• If corpus of positive reviews has 1000 words, and “Damon” occurs 50 times,
  
  \( p_{\text{ML}}(\text{Damon} \mid +) = ? \)

• If pos. corpus has “Affleck” 0 times,
  
  \( p(+) \mid \text{Affleck Damon movie}) = ? \)
Will the Sun Rise Tomorrow?
Will the Sun Rise Tomorrow?

Laplace’s Rule of Succession:
On day \(n+1\), we’ve observed that the sun has risen \(s\) times before.

\[
p_{Lap}(S_{n+1} = 1 \mid S_1 + \cdots + S_n = s) = \frac{s + 1}{n + 2}
\]

What’s the probability on day 0?
On day 1?
On day \(10^6\)?
Start with prior assumption of equal rise/not-rise probabilities; \textit{update} after every observation.
Laplace (Add One) Smoothing

- From our earlier example:
  \[ p_{ML}(\text{beginning} \mid \text{in the}) = 5/100? \quad \text{reduce!} \]
  \[ p_{ML}(\text{end} \mid \text{in the}) = 8/100? \quad \text{reduce!} \]
  \[ p_{ML}(\text{kitchen} \mid \text{in the}) = 0/100? \quad \text{increase!} \]
Laplace (Add One) Smoothing

- Let $V$ be the vocabulary size:
  i.e., the number of unique words that could follow “in the”

- From our earlier example:
  \[
  p_{ML}(\text{beginning} | \text{in the}) = \frac{5 + 1}{100 + V}
  \]
  \[
  p_{ML}(\text{end} | \text{in the}) = \frac{8 + 1}{100 + V}
  \]
  \[
  p_{ML}(\text{kitchen} | \text{in the}) = \frac{0 + 1}{100 + V}
  \]
Generalized Additive Smoothing

- Laplace add-one smoothing generally assigns *too much* probability to unseen words

- More common to use $\lambda$ instead of 1:

$$p(w_3 | w_1, w_2) = \frac{C(w_1, w_2, w_3) + \lambda}{C(w_1, w_2) + \lambda V}$$

$$= \mu \frac{C(w_1, w_2, w_3)}{C(w_1, w_2)} + (1 - \mu) \frac{1}{V}$$

$$\mu = \frac{C(w_1, w_2)}{C(w_1, w_2) + \lambda V}$$
Generalized Additive Smoothing

• Laplace add-one smoothing generally assigns *too much* probability to unseen words

• More common to use $\lambda$ instead of 1:

$$
p(w_3 \mid w_1, w_2) = \frac{C(w_1, w_2, w_3) + \lambda}{C(w_1, w_2) + \lambda V}
$$

interpolation

$$
\mu = \frac{C(w_1, w_2)}{C(w_1, w_2) + \lambda V}
$$
Generalized Additive Smoothing

• Laplace add-one smoothing generally assigns too much probability to unseen words

• More common to use $\lambda$ instead of 1:

$$p(w_3 \mid w_1, w_2) = \frac{C(w_1, w_2, w_3) + \lambda}{C(w_1, w_2) + \lambda V}$$

$$\mu = \frac{C(w_1, w_2)}{C(w_1, w_2) + \lambda V}$$
Picking Parameters

• What happens if we optimize parameters on training data, i.e. the same corpus we use to get counts?

• Maximum likelihood estimate!

• Use *held-out data* aka *development data*
Good-Turing Smoothing

- Intuition: Can judge rate of novel events by rate of singletons
  - Developed to estimate # of unseen species in field biology
- Let $N_r = \#$ of word types with $r$ training tokens
  - e.g., $N_0 = \text{number of unobserved words}$
  - e.g., $N_1 = \text{number of singletons (hapax legomena)}$
- Let $N = \sum r N_r = \text{total \# of training tokens}$
Good-Turing Smoothing

- Max. likelihood estimate if w has r tokens? \( r/N \)
- Total max. likelihood probability of all words with r tokens? \( N_r r/N \)
- Good-Turing estimate of this total probability:
  - Defined as: \( N_{r+1} (r+1) / N \)
  - So proportion of novel words in test data is estimated by proportion of singletons in training data.
  - Proportion in test data of the \( N_1 \) singletons is estimated by proportion of the \( N_2 \) doubletons in training data. etc.
  - \( p(\text{any given word } w/\text{freq. } r) = N_{r+1} (r+1) / (N N_r) \)
- NB: No parameters to tune on held-out data
Backoff

• Say we have the counts:

\[ C(\text{in the kitchen}) = 0 \]
\[ C(\text{the kitchen}) = 3 \]
\[ C(\text{kitchen}) = 4 \]
\[ C(\text{arboretum}) = 0 \]

• ML estimates seem counterintuitive:

\[ p(\text{kitchen} \mid \text{in the}) = p(\text{arboretum} \mid \text{in the}) = 0 \]
Backoff

• Clearly we shouldn’t treat “kitchen” the same as “arboretum”

• Basic add-\(\lambda\) (and similar) smoothing methods assign the same prob. to all unseen events

• **Backoff** divides up prob. of unseen unevenly in proportion to, e.g., lower-order n-grams

• If \(p(z \mid x,y) = 0\), use \(p(z \mid y)\), etc.
Deleted Interpolation

• Simplest form of backoff (Jelinek-Mercer)
• Form a mixture of different order n-gram models; learn weights on held-out data

\[ p_{del}(z \mid x, y) = \alpha_3 p(z \mid x, y) + \alpha_2 p(z \mid y) + \alpha_1 p(z) \]
\[ \sum \alpha_i = 1 \]

• How else could we back off?
Reading


- LM background: Jurafsky & Martin, c.4