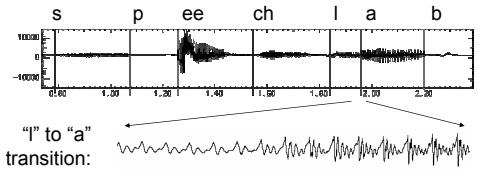
CS 294-5: Statistical Natural Language Processing

Dan Klein
MF 1:10-2:30pm
Soda Hall 310

Speech in a Slide (or Three)

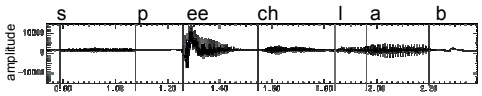
- Speech input is an acoustic wave form



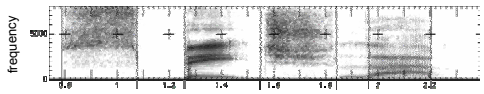
Graphs from Simon Arnfield's web tutorial on speech, Sheffield:
http://www.psyc.leeds.ac.uk/research/cogn/speech/tutorial/
Some later bits from Joshua Goodman's LM tutorial

Spectral Analysis

- Frequency gives pitch; amplitude gives volume
  - sampling at ~8 kHz phone, ~16 kHz mic (kHz=1000 cycles/sec)

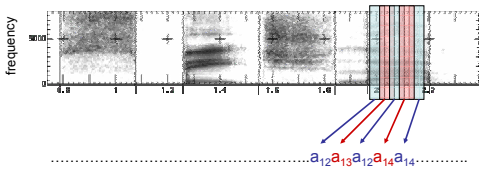


- Fourier transform of wave displayed as a spectrogram
  - darkness indicates energy at each frequency



Acoustic Feature Sequence

- Time slices are translated into acoustic feature vectors (~15 real numbers per slice)



- Now we have to figure out a mapping from sequences of acoustic observations to words.

The Speech Recognition Problem

- We want to predict a sentence given an acoustic sequence:

$$s^* = \arg\max_s P(s \mid A)$$

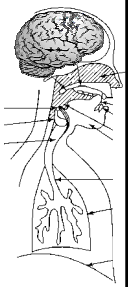
- The noisy channel approach:
  - Build a generative model of production (encoding)

$$P(A, s) = P(s)\,P(A \mid s)$$

  - To decode, we use Bayes' rule to write

$$s^* = \arg\max_s P(s \mid A)$$
$$= \arg\max_s P(s)P(A \mid s)/P(A)$$
$$= \arg\max_s P(s)P(A \mid s)$$

  - Now, we have to find a sentence maximizing this product
- Why is this progress?



Other Noisy-Channel Processes

- Handwriting recognition

$$P(text \mid strokes) \propto P(text)P(strokes \mid text)$$

- OCR

$$P(text \mid pixels) \propto P(text)P(pixels \mid text)$$

- Spelling Correction

$$P(text \mid typos) \propto P(text)P(typos \mid text)$$

- Translation?

$$P(english \mid french) \propto P(english)P(french \mid english)$$

Probabilistic Language Models

- Want to build models which assign scores to sentences.
  - P(I saw a van) >> P(eyes awe of an)
  - Not really grammaticality: P(artichokes intimidate zippers) ≈ 0
- One option: empirical distribution over sentences?
  - Problem: doesn't generalize (at all)
- Two ways of generalizing
  - Decomposition: sentences generated in small steps which can be recombined in other ways
  - Smoothing: allow for the possibility of unseen events

N-Gram Language Models

- No loss of generality to break sentence probability down with the chain rule

$$P(w_1 w_2 \ldots w_n) = \prod_i P(w_i \mid w_1 w_2 \ldots w_{i-1})$$

- Too many histories!
- N-gram solution: assume each word depends only on a short linear history

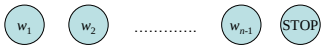
$$P(w_1 w_2 \ldots w_n) = \prod_i P(w_i \mid w_{i-k} \ldots w_{i-1})$$

Unigram Models

- Simplest case: unigrams

$$P(w_1 w_2 \ldots w_n) = \prod_i P(w_i)$$

- Generative process: pick a word, pick a word, …
- As a graphical model:



- To make this a proper distribution over sentences, we have to generate a special STOP symbol last. (Why?)
- Examples:
  - [fifth, an, of, futures, the, an, incorporated, a, a, the, inflation, most, dollars, quarter, in, is, mass.]
  - [thrift, did, eighty, said, hard, 'm, july, bullish]
  - [that, or, limited, the]
  - []
  - [after, any, on, consistently, hospital, lake, of, of, other, and, factors, raised, analyst, too, allowed, mexico, never, consider, fall, bungled, davison, that, obtain, price, lines, the, to, sass, the, the, further, board, a, details, machinists, the, companies, which, rivals, an, because, longer, oakes, percent, a, they, three, edward, it, currier, an, within, in, three, wrote, is, you, s., longer, institute, dentistry, pay, however, said, possible, to, rooms, hiding, eggs, approximate, financial, canada, the, so, workers, advancers, half, between, nasdaq]

Bigram Models

- Big problem with unigrams: P(the the the the) >> P(I like ice cream)!
- Condition on last word:

$$P(w_1 w_2 \ldots w_n) = \prod_i P(w_i \mid w_{i-1})$$

- Any better?
  - [texaco, rose, one, in, this, issue, is, pursuing, growth, in, a, boiler, house, said, mr., gurria, mexico, 's, motion, control, proposal, without, permission, from, five, hundred, fifty, five, yen]
  - [outside, new, car, parking, lot, of, the, agreement, reached]
  - [although, common, shares, rose, forty, six, point, four, hundred, dollars, from, thirty, seconds, at, the, greatest, play, disingenuous, to, be, reset, annually, the, buy, out, of, american, brands, vying, for, mr., womack, currently, sharedata, incorporated, believe, chemical, prices, undoubtedly, will, be, as, much, is, scheduled, to, conscientious, teaching]
  - [this, would, be, a, record, november]

Is This Working?

- The game isn't to pound out fake sentences!
- What we really want to know is:
  - Will our model prefer good sentences from bad ones?
  - Bad ≠ ungrammatical!
  - Bad = sentences that our acoustic model really likes but aren't the correct answer

Regular Languages?

- Weighted deterministic automaton
  - Why can't we model language like this?
    - "The computer which I had just put into the machine room on the fifth floor crashed."
  - Why CAN we?
    - 74% of dependencies in the Penn treebank are between adjacent words.
    - 95% have no more than 4 words intervening

"The cat scrambled up the tree in my yard"

Measuring Model Quality

- Word Error Rate (WER)

$$\frac{\text{insertions} + \text{deletions} + \text{substitutions}}{\text{true sentence size}}$$

Correct answer: Andy saw a part of the movie

Recognizer output: And he saw apart of a movie

WER: 4/7 = 57%

- The "right" measure:
  - Task error driven
  - For speech recognition
  - For a specific recognizer!
- For general evaluation, we want a measure which references only good text, not mistake text

Measuring Model Quality

- The Shannon Game:
  - How well can we predict the next word?
    - When I order pizza, I wipe off the ____
    - Many children are allergic to ____
    - I saw a ____
  - A unigram model is terrible at this! (Why?)

grease 0.5
sauce 0.4
dust 0.05
….
mice 0.0001
….
the 1e-100

- The "Entropy" Measure
  - Really: average cross-entropy of a text according to a model

$$H(S \mid M) = \frac{\log_2 P_M(S)}{|S|} = \frac{\sum_i \log_2 P_M(s_i)}{\sum_i |s_i|}$$

$$\sum_j \log_2 P_M(w_j \mid w_{j-1})$$

Measuring Model Quality

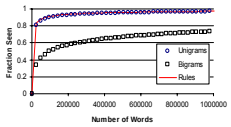
- Problem with entropy:
  - 0.1 bits of improvement doesn't sound so good
  - Solution: perplexity

$$P(S \mid M) = 2^{H(S|M)} = \sqrt[n]{\frac{1}{\prod_{i=1}^{n} P_M(w_i \mid h)}}$$

  - Note that even though our models require a stop step, we don't count it as a symbol when taking these averages.

Sparsity

- Problems with n-gram models:
  - New words appear all the time:
    - Synaptitute
    - 132,701.03
    - fuzzification al
  - New bigrams: even more often
  - Trigrams or more – still worse!



- Zipf's Law
  - Types (words) vs. tokens (word occurences)
  - Broadly: most word types are rare
  - Specifically:
    - Rank word types by token frequency
    - Frequency inversely proportional to rank
  - Not special to language: randomly generated character strings have this property

Smoothing

- Estimating multinomials
  - We want to know what words follow some history h
  - There's some true distribution P(w | h) over a large space of words
  - We saw some small sample of N words from P(w | h)
  - We want to reconstruct a useful approximation of P(w | h)
  - Counts of events we didn't see are always too low (0 < N P(w | h))
  - Counts of events we did see are *in aggregate* to high
- Example:

P(w | denied the)
3 allegations
2 reports
1 claims
1 speculation
…
1 request
13 total

P(w | affirmed the)
1 award

- Two issues:
  - Discounting: how to reserve mass what we haven't seen
  - Interpolation: how to allocate that mass amongst unseen events

What's Next

- Next class:
  - Smoothing Details
  - Text categorization
  - Naïve-Bayes models
  - Class-conditional language models
  - How to actually make these things work
- Reading: M+S 6, J+M 6-7, Chen + Goodman paper on web page