Hidden Markov Models: Decoding and Prediction

With slides from Dan Klein and Pieter Abbeel

Project 3: Ghostbusters!

- Due ~~**Wed March 21st**~~ **Friday March 23rd**
- http://www.mathcs.emory.edu/~eugene/cs557/p4/

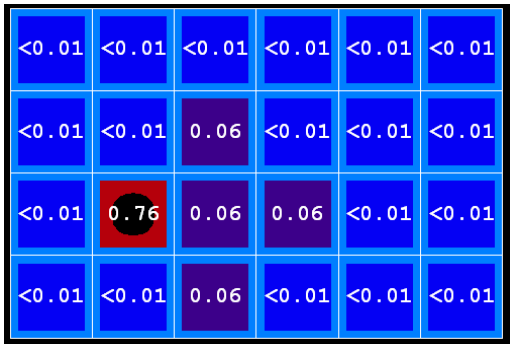
Recap: Filtering

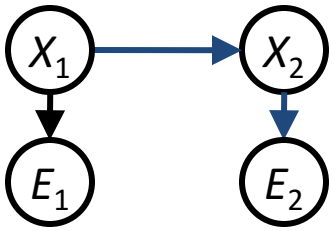
Elapse time: compute P(X_t | $e_{1:t-1}$)

$$P(x_t|e_{1:t-1}) = \sum_{x_{t-1}} P(x_{t-1}|e_{1:t-1}) \cdot P(x_t|x_{t-1})$$

Observe: compute P(X_t | $e_{1:t}$)

$$P(x_t|e_{1:t}) \propto P(x_t|e_{1:t-1}) \cdot P(e_t|x_t)$$



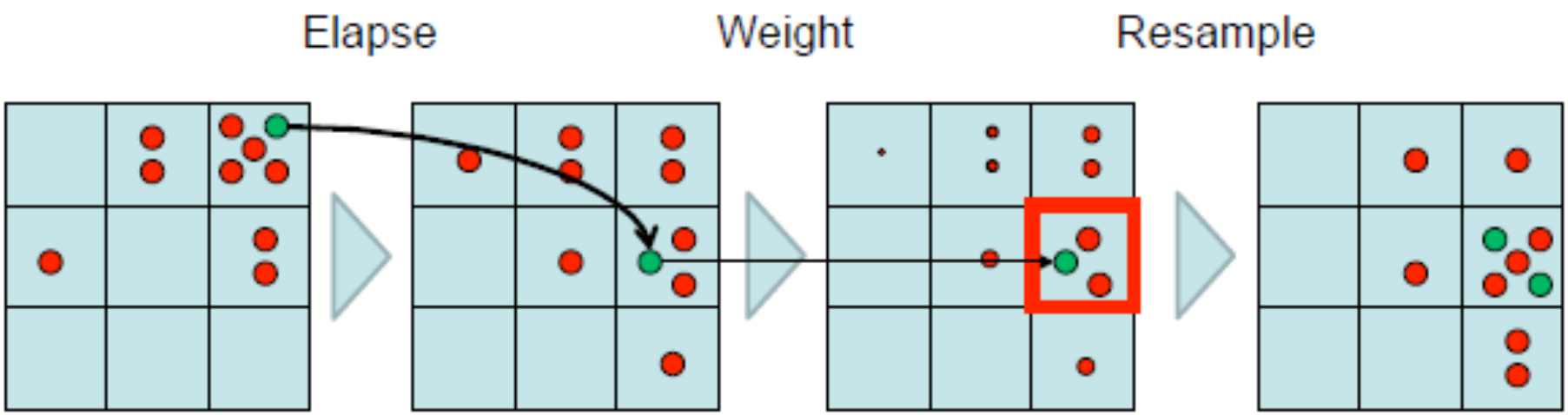


Belief: <P(rain), P(sun)>

$P(X_1)$	<0.5, 0.5>	*Prior on X_1*
$P(X_1 \mid E_1 = umbrella)$	<0.82, 0.18>	*Observe*
$P(X_2 \mid E_1 = umbrella)$	<0.63, 0.37>	*Elapse time*
$P(X_2 \mid E_1 = umb, E_2 = umb)$	<0.88, 0.12>	*Observe*

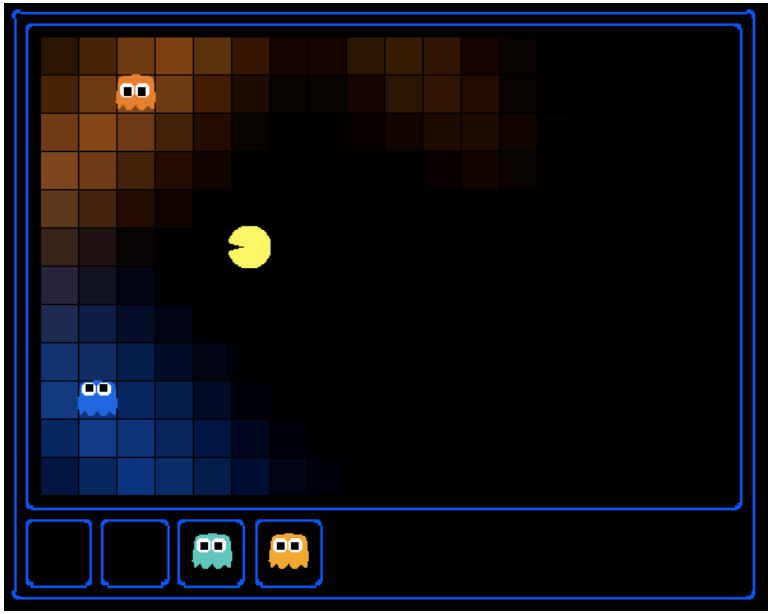
Recap: Particle Filtering

- Particles: track samples of states rather than an explicit distribution



Particles:	Particles:	Particles:	(New) Particles:
(3,3)	(3,2)	(3,2) w=.9	(3,2)
(2,3)	(2,3)	(2,3) w=.2	(2,2)
(3,3)	(3,2)	(3,2) w=.9	(3,2)
(3,2)	(3,1)	(3,1) w=.4	(2,3)
(3,3)	(3,3)	(3,3) w=.4	(3,3)
(3,2)	(3,2)	(3,2) w=.9	(3,2)
(1,2)	(1,3)	(1,3) w=.1	(1,3)
(3,3)	(2,3)	(2,3) w=.2	(2,3)
(3,3)	(3,2)	(3,2) w=.9	(3,2)
(2,3)	(2,2)	(2,2) w=.4	(3,2)

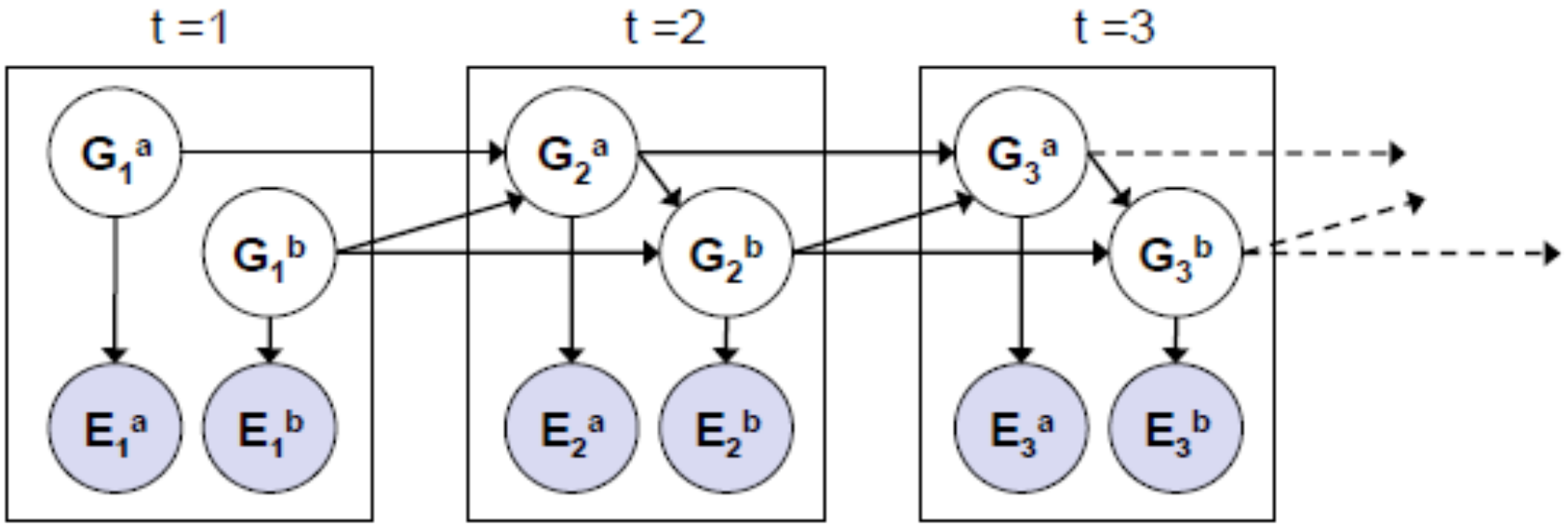
What if Ghosts Avoid Each Other?



- Since the ghosts' transition models are no longer independent, all ghosts must be tracked jointly.
- How?

Dynamic Bayes Nets (DBNs)

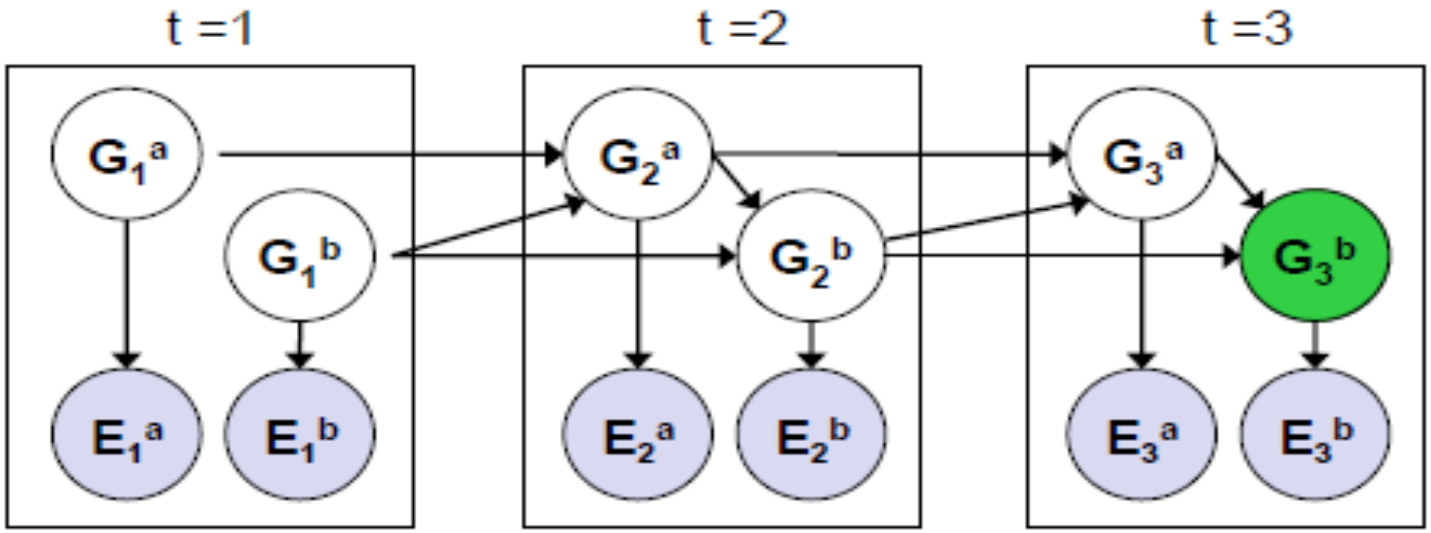
- We want to track multiple variables over time, using multiple sources of evidence
- Idea: Repeat a fixed Bayes net structure at each time
- Variables from time *t* can condition on those from *t-1*



- Discrete valued dynamic Bayes nets are also HMMs

Exact Inference in DBNs

- Procedure: "unroll" the network for T time steps, then eliminate variables until $P(X_T|e_{1:T})$ is computed



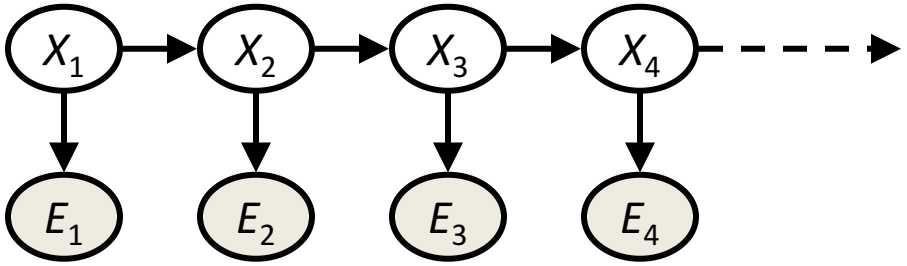
DBN Particle Filters

- A particle is a complete sample for a time step
- **Initialize**: Generate prior samples for the t=1 Bayes net
  - Example particle: $\mathbf{G}_1^a = (3,3)$ $\mathbf{G}_1^b = (5,3)$
- **Elapse time**: Sample a successor for each particle
  - Example successor: $\mathbf{G}_2^a = (2,3)$ $\mathbf{G}_2^b = (6,3)$
- **Observe**: Weight each *entire* sample by the likelihood of the evidence conditioned on the sample
  - Likelihood: $P(\mathbf{E}_1^a \mid \mathbf{G}_1^a) * P(\mathbf{E}_1^b \mid \mathbf{G}_1^b)$
- **Resample:** Select prior samples (tuples of values) in proportion to their likelihood

Most Likely Explanation

HMMs: MLE Queries

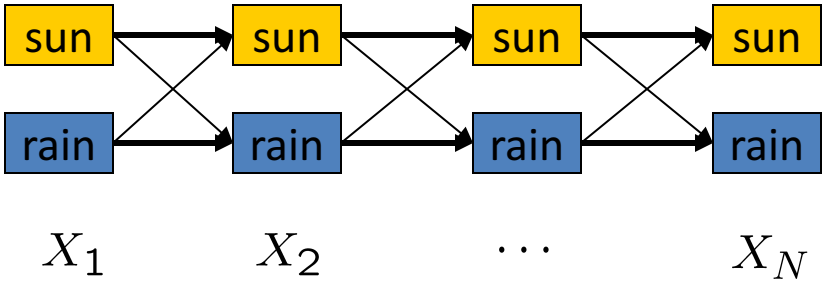
- HMMs defined by
  - States X
  - Observations E
  - Initial distribution: $P(X_1)$
  - Transitions: $P(X|X_{-1})$
  - Emissions: $P(E|X)$



- New query: most likely explanation: $\arg\max_{x_{1:t}} P(x_{1:t}|e_{1:t})$
- New method: the Viterbi algorithm

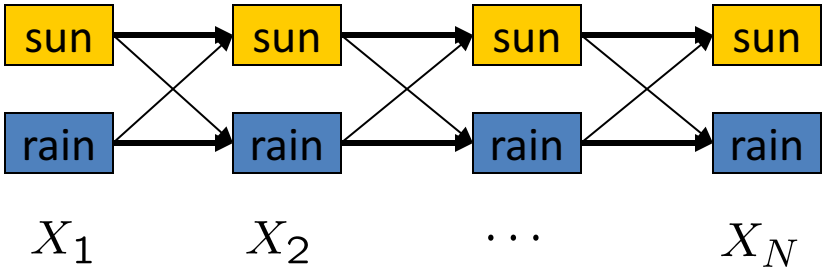
State Trellis

- State trellis: graph of states and transitions over time



- Each arc represents some transition $x_{t-1} \rightarrow x_t$
- Each arc has weight $P(x_t|x_{t-1})P(e_t|x_t)$
- Each path is a sequence of states
- The product of weights on a path is that sequence's probability along with the evidence
- Forward algorithm computes sums of paths, Viterbi computes best paths

Forward / Viterbi Algorithms



Forward Algorithm (Sum)

$$f_t[x_t] = P(x_t, e_{1:t})$$

$$= P(e_t|x_t) \sum_{x_{t-1}} P(x_t|x_{t-1}) f_{t-1}[x_{t-1}]$$

Viterbi Algorithm (Max)

$$m_t[x_t] = \max_{x_{1:t-1}} P(x_{1:t-1}, x_t, e_{1:t})$$

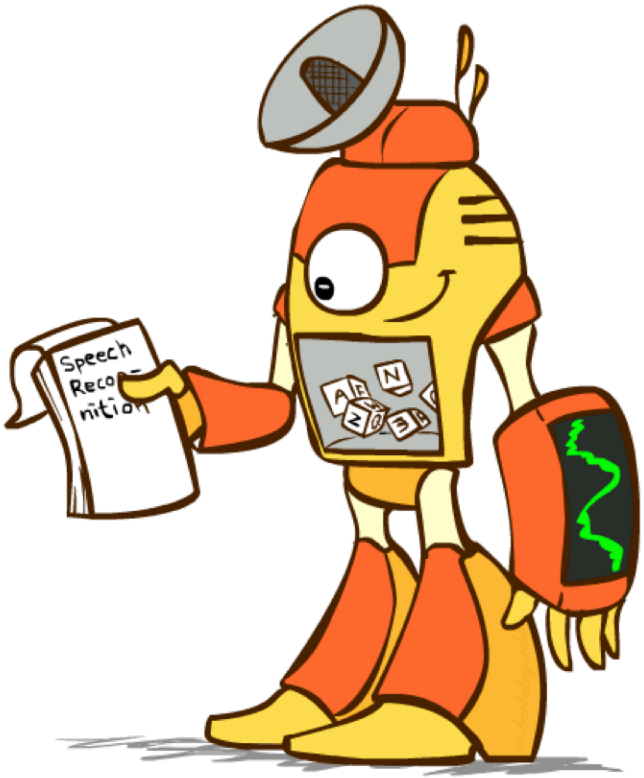
$$= P(e_t|x_t) \max_{x_{t-1}} P(x_t|x_{t-1}) m_{t-1}[x_{t-1}]$$

Large problems? bad

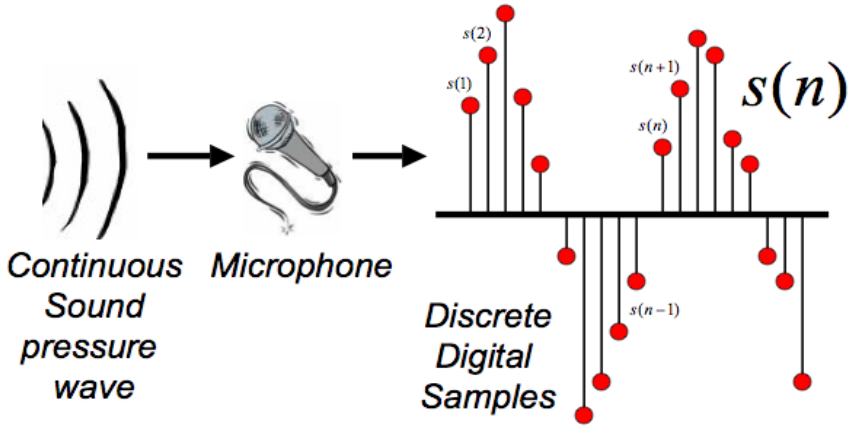
- Numerical stability?

- http://crawlingrobotfortress.blogspot.com/2016/07/python-recipe-for-numerically-stable.html

HMMs for Speech Recognition



Digitizing Speech



Thanks to Bryan Pellom for this slide!

Speech Overview

- Speech input is an acoustic waveform

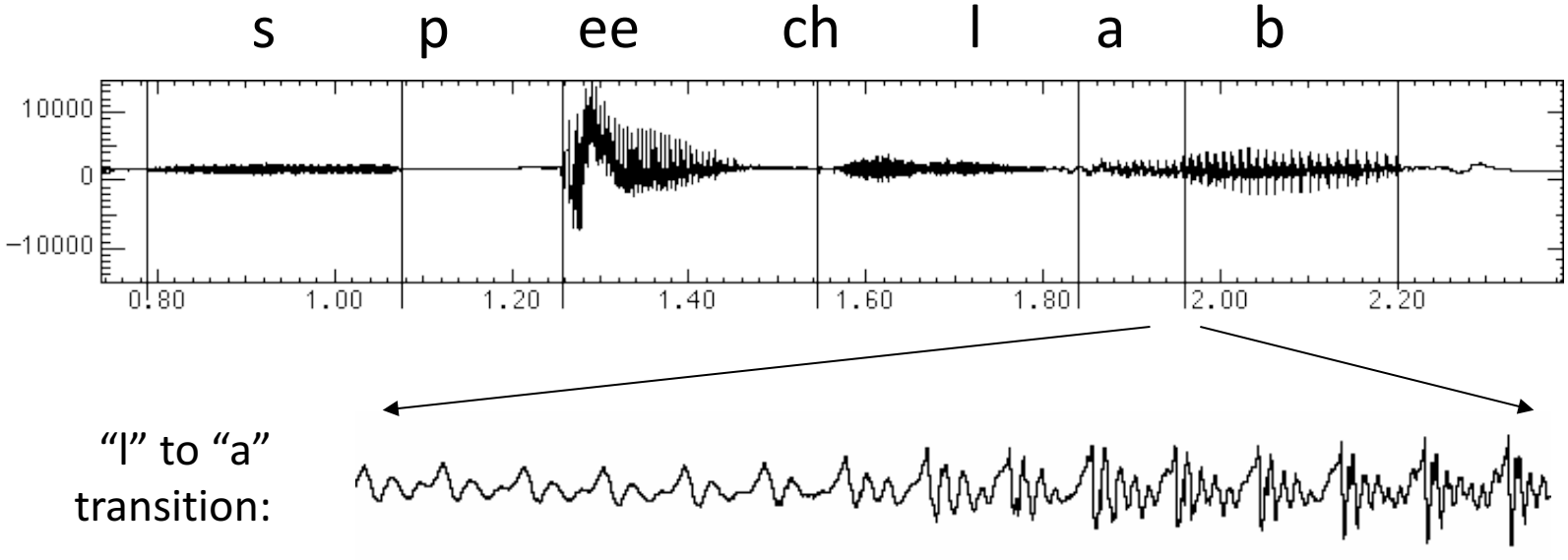
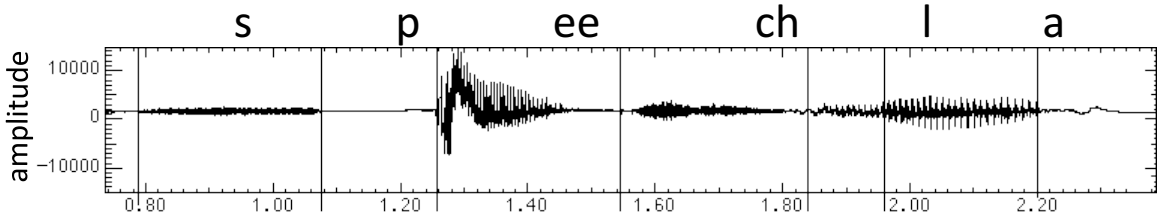


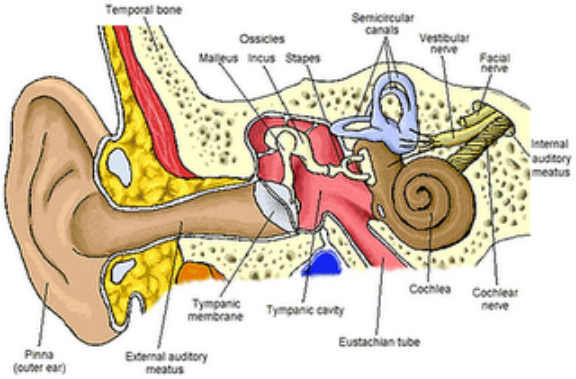
Figure: Simon Arnfield, http://www.psyc.leeds.ac.uk/research/cogn/speech/tutorial/

Spectral Analysis

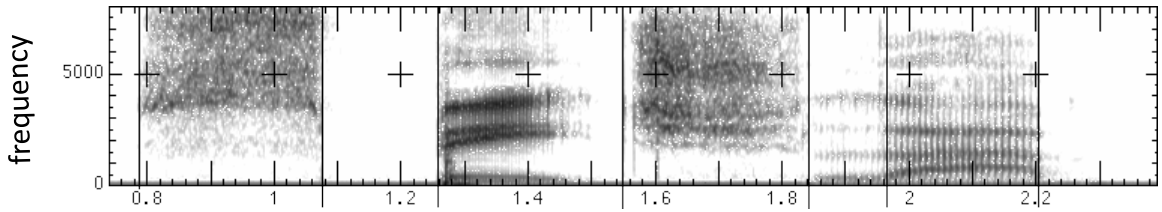
- Frequency gives pitch; amplitude gives volume
  - Sampling at ~8 kHz (phone), ~16 kHz (mic) (kHz=1000 cycles/sec)

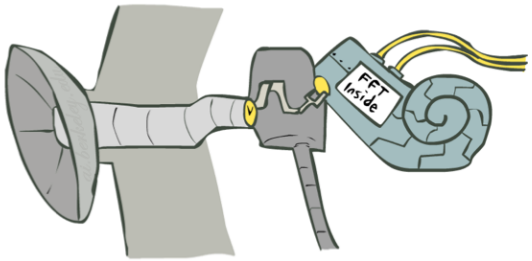


b



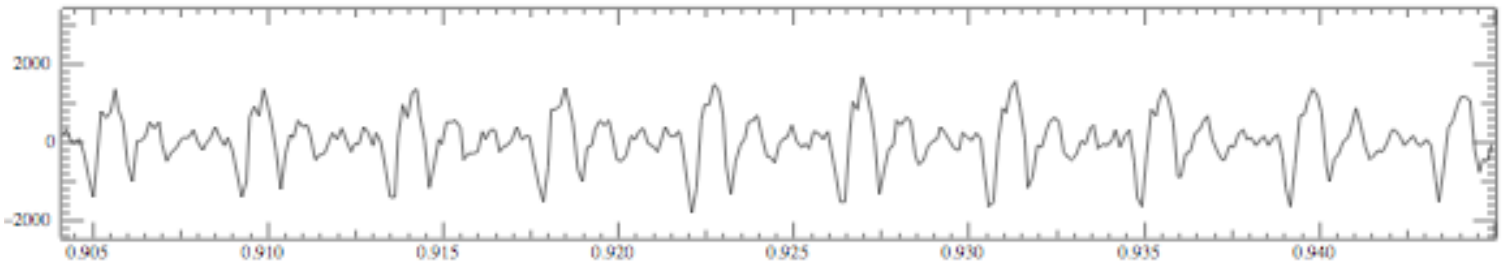
- Fourier transform of wave displayed as a spectrogram
  - Darkness indicates energy at each frequency



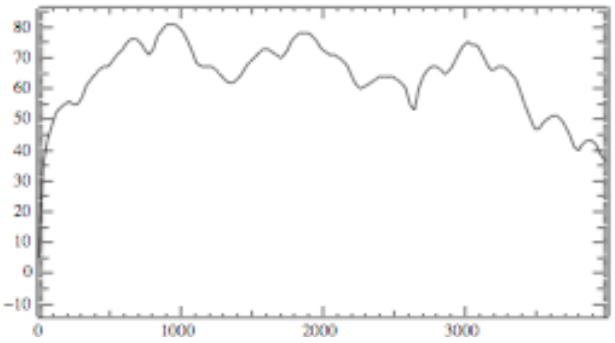


Human ear figure: depion.blogspot.com

Part of [ae] from “lab”

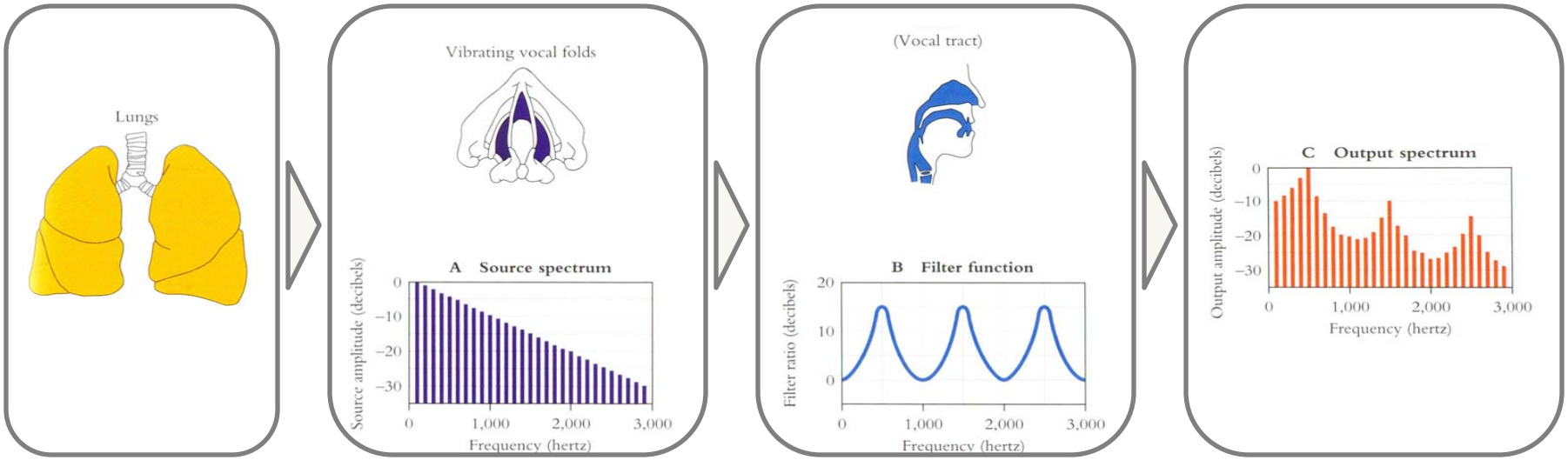


- Complex wave repeating nine times
  - Plus smaller wave that repeats 4x for every large cycle
  - Large wave: freq of 250 Hz (9 times in .036 seconds)
  - Small wave roughly 4 times this, or roughly 1000 Hz



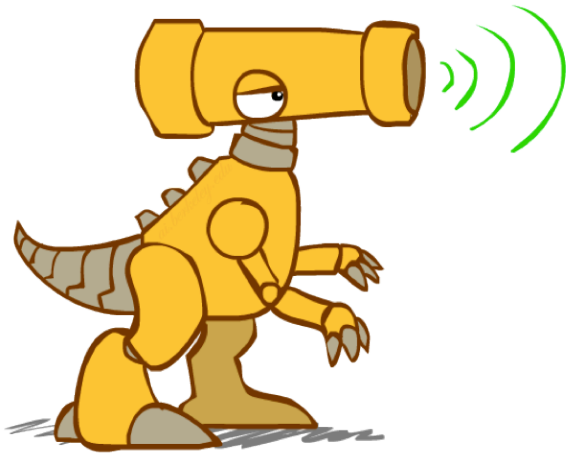
Why These Peaks?

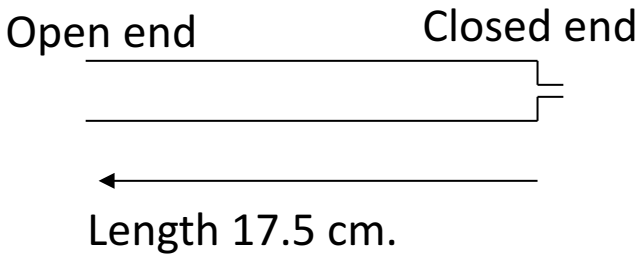
- Articulator process:
  - Vocal cord vibrations create harmonics
  - The mouth is an amplifier
  - Depending on shape of mouth, some harmonics are amplified more than others

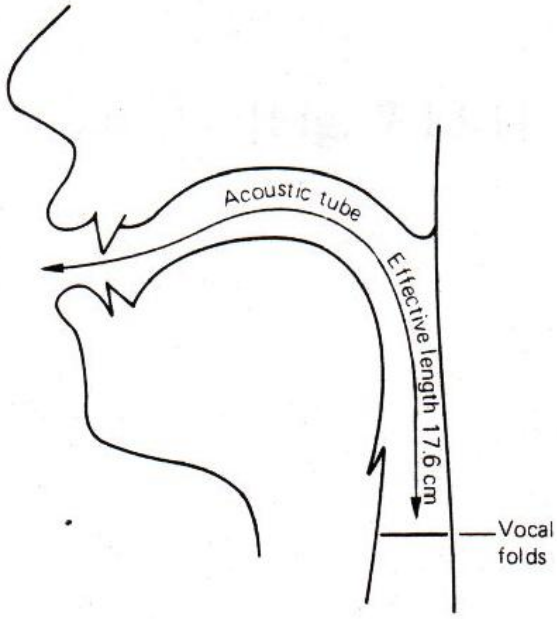


Resonances of the Vocal Tract

- The human vocal tract as an open tube







- Air in a tube of a given length will tend to vibrate at resonance frequency of tube
- Constraint: Pressure differential should be maximal at (closed) glottal end and minimal at (open) lip end

Figure: W. Barry Speech Science slides

Spectrum Shapes

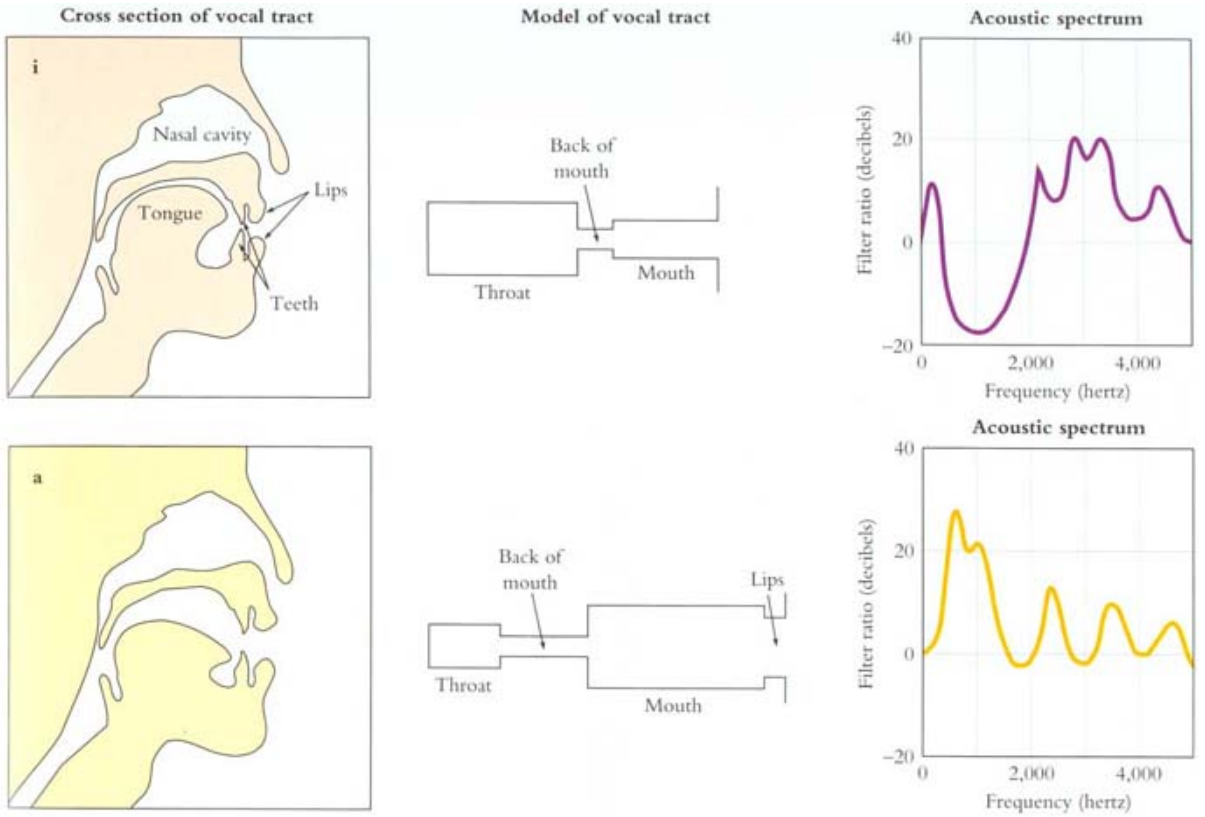
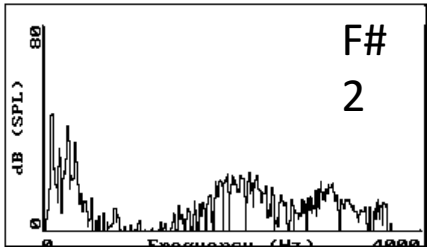
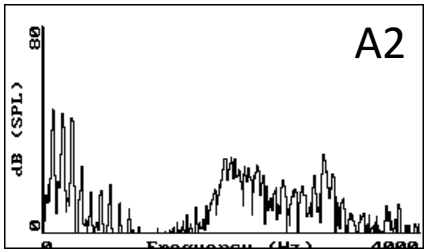
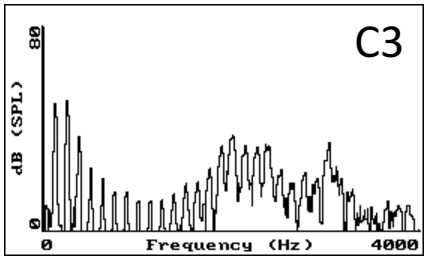


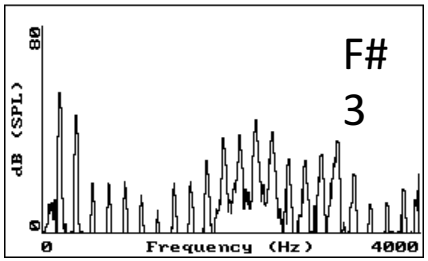
Figure: Mark Liberman

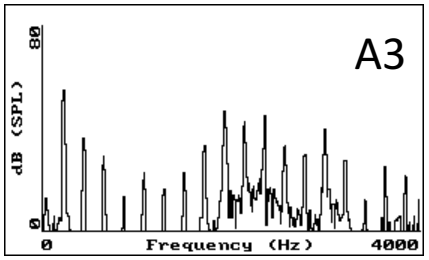
Vowel [i] sung at successively higher pitches

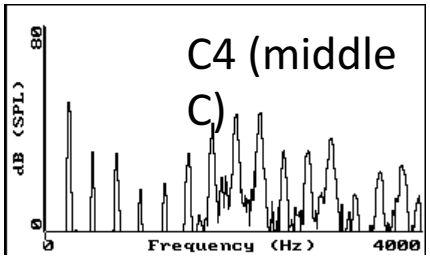


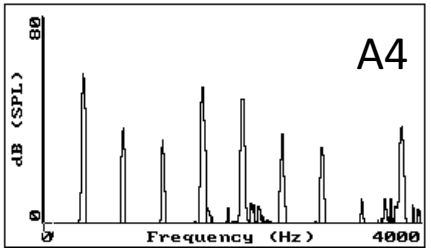


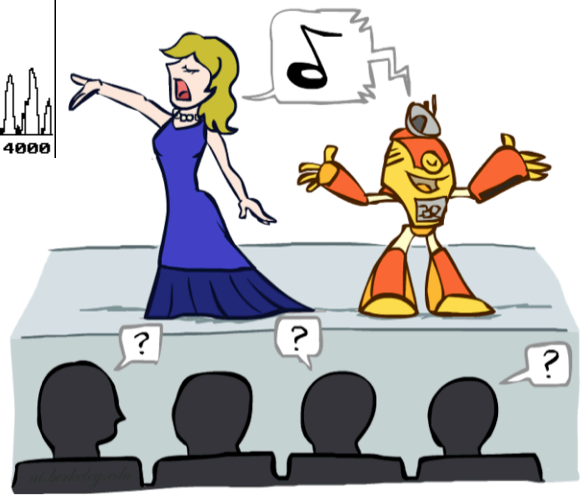








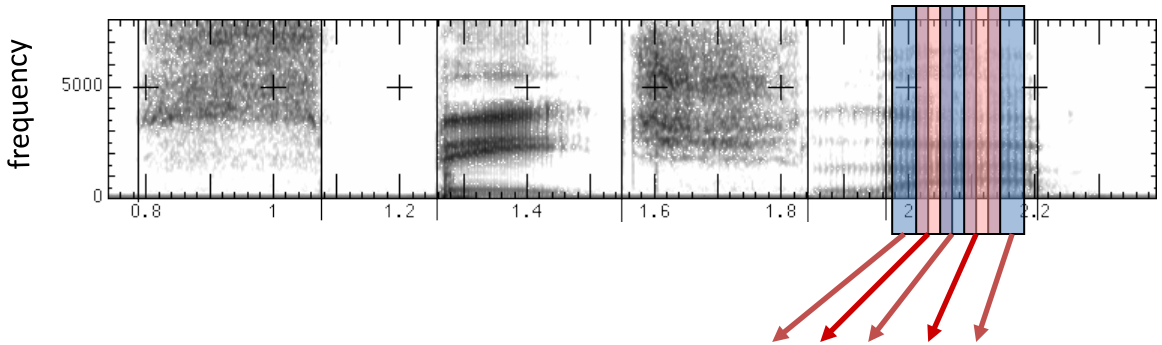




Graphs: Ratree Wayland

Acoustic Feature Sequence

- Time slices are translated into acoustic feature vectors (~39 real numbers per slice)



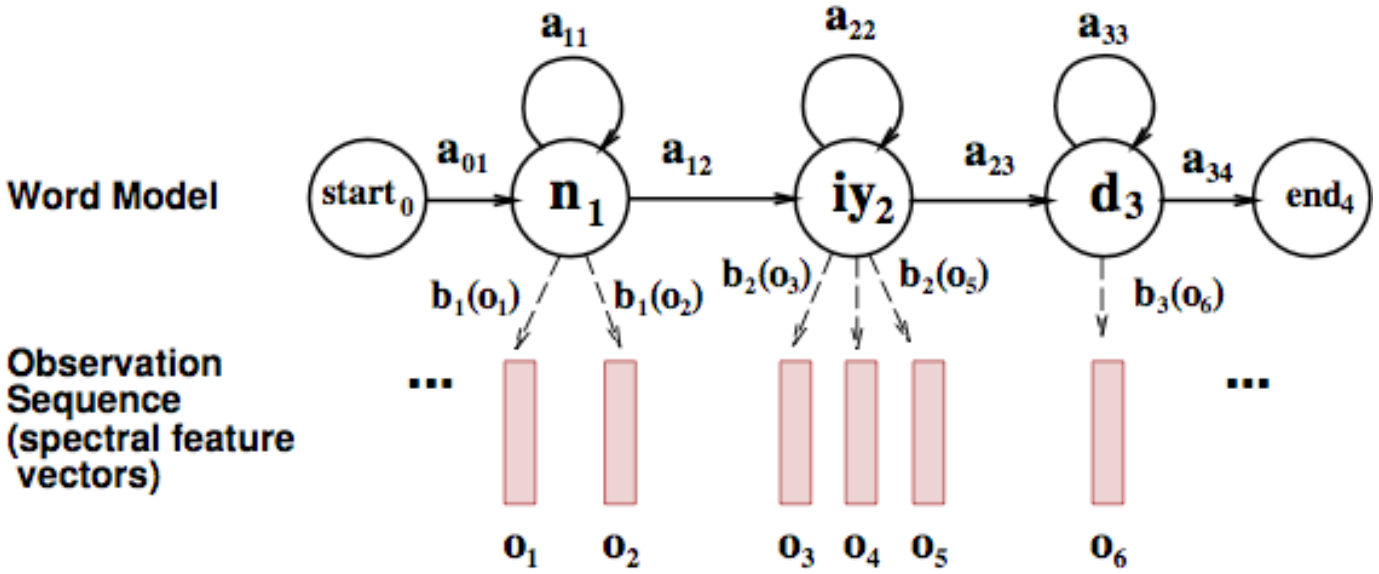
- These are the observations E, now we need the hidden states X

Speech State Space

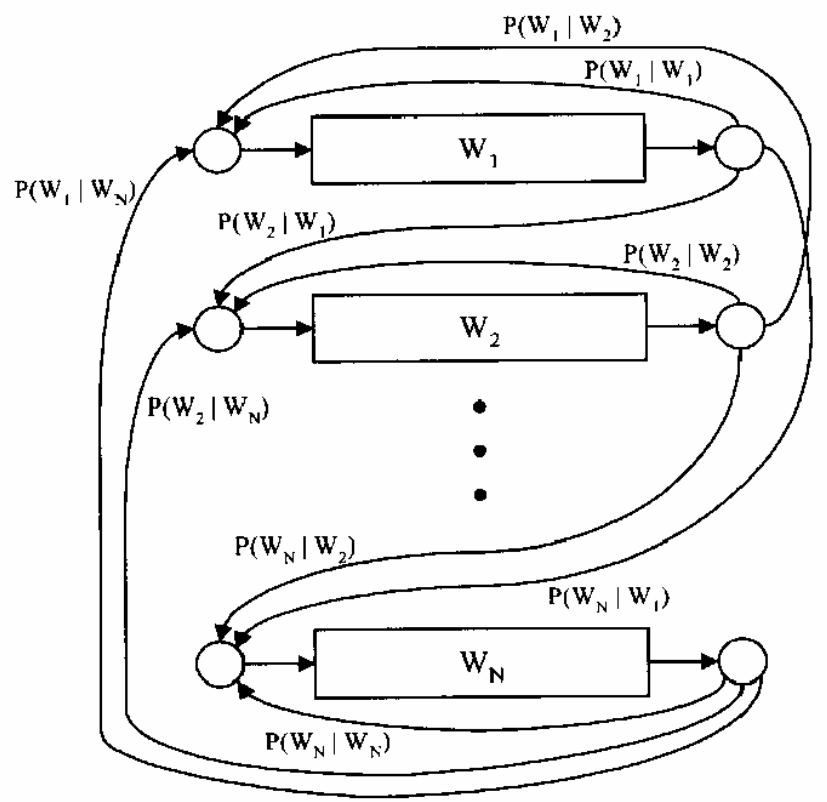
- HMM Specification
  - P(E|X) encodes which acoustic vectors are appropriate for each phoneme (each kind of sound)
  - P(X|X') encodes how sounds can be strung together

- State Space
  - We will have one state for each sound in each word
  - Mostly, states advance sound by sound
  - Build a little state graph for each word and chain them together → state space X

States in a Word



Transitions with a Bigram Model



Training Counts

198015222 the first
194623024 the same
168504105 the following
158562063 the world
...
14112454 the door

23135851162 the *

$$\hat{P}(\text{door}|\text{the}) = \frac{14112454}{23135851162} = 0.0006$$

Figure: Huang et al, p. 618

Decoding

- Finding the words given the acoustics is an HMM inference problem
- Which state sequence $x_{1:T}$ is most likely given the evidence $e_{1:T}$?

$$x^*_{1:T} = \arg\max_{x_{1:T}} P(x_{1:T}|e_{1:T}) \quad = \arg\max_{x_{1:T}} P(x_{1:T}, e_{1:T})$$

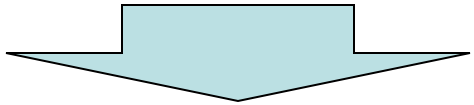
- From the sequence x, we can read off the word candidates
- Pick most likely next word using word statistics (e.g., bigram model last slide)

HMMs for Natural Language Processing

- HMMs are very common in NLP:
  - Speech recognition (observed: acoustic signal, hidden: words)
  - Handwriting recognition (observed: image, hidden: words)
  - ➢ **Part-of-speech tagging (observed: words, hidden: part-of-speech tags)**
  - Machine translation (observed: foreign words, hidden: words in target language)

HMMs for POS Tagging

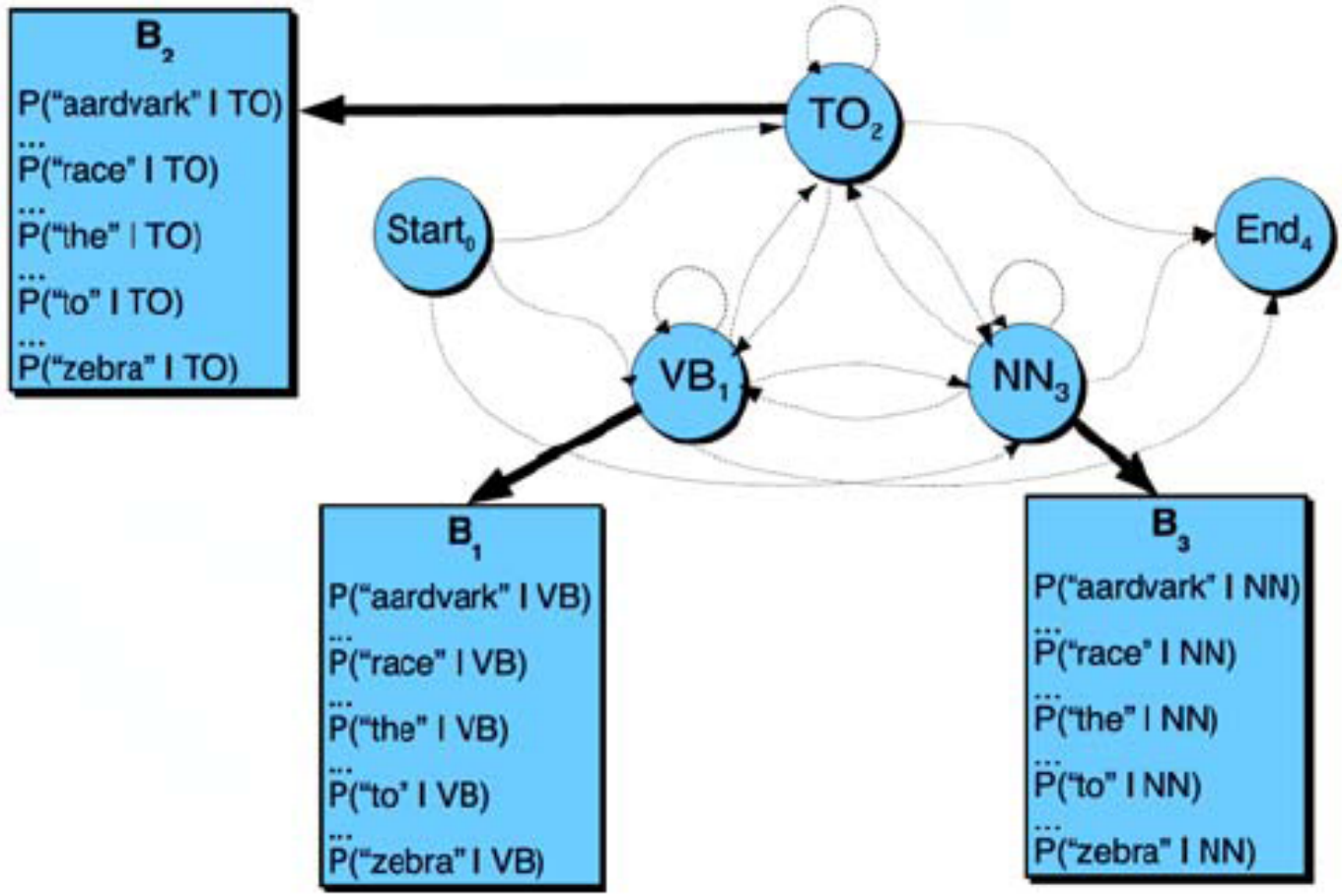
The Georgia branch had taken on loan commitments …



DT NNP NN VBD VBN RP NN NNS

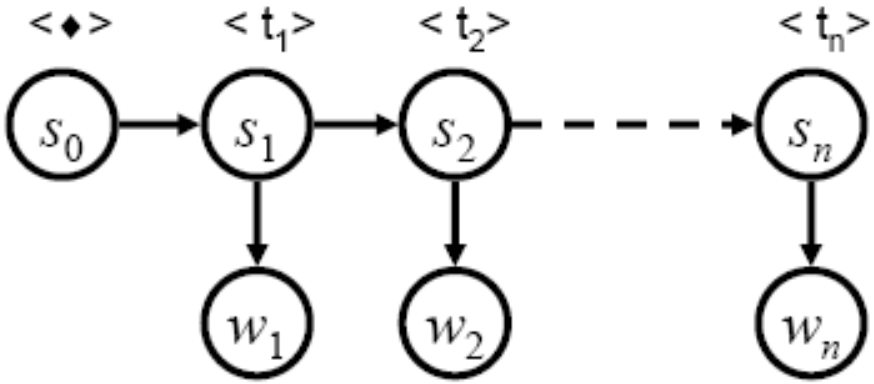
- HMM Model:
  - States Y = {DT, NNP, NN, ... } are the POS tags
  - Observations X = V are words
  - Transition dist'n $q(y_i | y_{i-1})$ models the tag sequences
  - Emission dist'n $e(x_i | y_i)$ models words given their POS
- Q: How to we represent n-gram POS taggers?

Transitions and Emissions

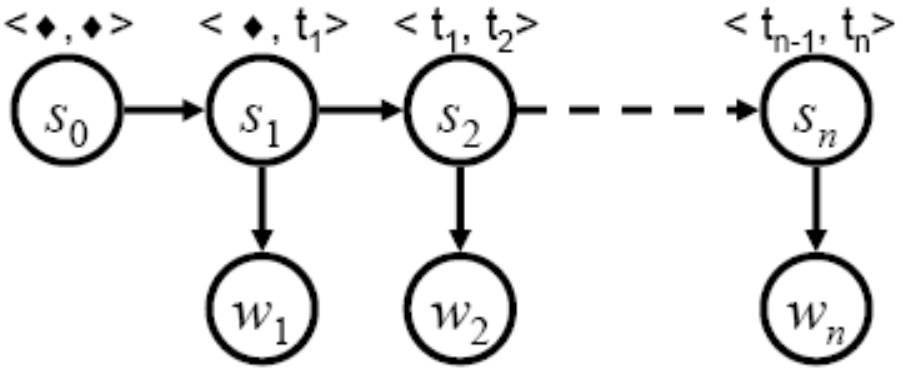


Transitions

- Transitions P(s|s') encode well-formed tag sequences
  - In a bigram tagger, states = tags



  - In a trigram tagger, states = tag pairs



Current Performance

Input: `the lead paint is unsafe`

Output: `the/Det lead/N paint/N is/V unsafe/Adj`

- How many tags are correct?
  - About 97% currently
  - But baseline is already 90%
    - Baseline is performance of simplest possible method:
      - Tag every word with its most frequent tag
      - Tag unknown words as nouns

Training an HMM (Supervised)

1. Define model topology (states, possible arcs)
2. Obtain labeled/tagged data
3. Estimate HMM parameters:
    1. Transition probabilities
    2. Emission probabilities
4. Validate on hold-out data
5. Done.

MLE Example (from Grishman, Fall 2004)

- Consider a random pig:
  - Observations:
    - squeeze it 10 times, and 8 times it goes "oink" and 2 times it goes “wee“
  - Model topology:
    - assume it is a stateless pig (no hidden states)
  - Parameter estimation:
    - Maximum likelihood estimate (MLE) to model the pig as a random emitter with P("oink") = 0.8 and P('“wee") = 0.2.
    - P(x) = count(X) / total number of trials

Learning: Maximum Likelihood

$$p(x_1 \ldots x_n, y_1 \ldots y_n) = q(STOP|y_n) \prod_{i=1}^{n} q(y_i|y_{i-1}) e(x_i|y_i)$$

- Learning
  - Maximum likelihood methods for estimating transitions q and emissions e

$$q_{ML}(y_i|y_{i-1}) = \frac{c(y_{i-1}, y_i)}{c(y_{i-1})} \qquad e_{ML}(x|y) = \frac{c(y, x)}{c(y)}$$

  - Will these estimates be high quality?
    - Which is likely to be more sparse, q or e?

Estimating Transitions: N-gram model

- Each word is predicted according to a conditional distribution based on a limited prior context
- Conditional Probability Table (CPT): $P(X|both)$
  - $P(of|both) = 0.066$
  - $P(to|both) = 0.041$
  - $P(in|both) = 0.038$
- From 1940s onward (or even 1910s – Markov 1913)
- a.k.a. Markov (chain) models

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]

The Words They Used

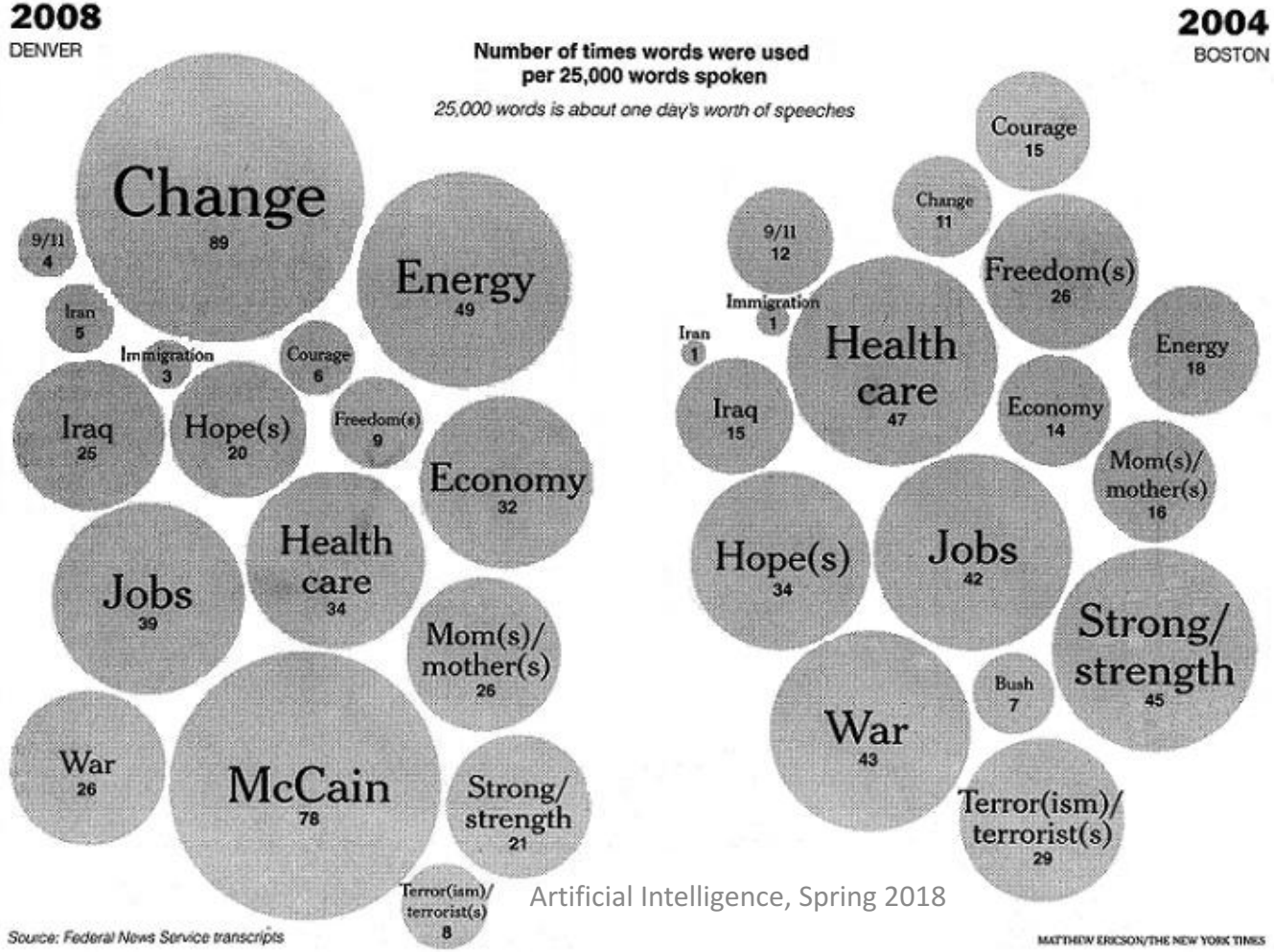
The words that the speakers have used during the Democratic convention suggest how the party's themes have changed since the last presidential campaign.

Speakers have hammered home Barack Obama's "change" theme, using the word about eight times as often as they did in 2004.

Also, unlike 2004, when the Kerry campaign sought to avoid direct attacks on the president at the convention, the speakers have regularly have been mentioning John McCain by name. Speakers in 2004 practiced "the art of the implicit slam," a veteran Democratic speechwriter said then, indirectly bashing Mr. Bush while barely using his name.

Also on the upswing: more mentions of the economy, energy, Iran and Iraq.

Words less frequently used: freedom, Sept. 11 and terrorism.



Bigram Models (Markov)

- 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]

Google Billion Word Corpus: Word-N-Gram Statistics

- serve as the incoming 92
- serve as the incubator 99
- serve as the independent 794
- serve as the index 223
- serve as the indication 72
- serve as the indicator 120
- serve as the indicators 45
- serve as the indispensable 111
- serve as the indispensible 40
- serve as the individual 234

Estimating Emission Probabilities

$$P(\mathbf{s}, \mathbf{w}) = \prod_i P(s_i|s_{i-1})P(w_i|s_i)$$

- Emissions are tricker:
  - Words we've never seen before
  - Words which occur with tags we've never seen
  - Will use new method: "smoothing"
  - Issue: words aren't black boxes:

    343,127.23 11-year Minteria reintroducibly

  - Unknown words usually broken into word classes

    D^+,D^+.D^+ D^+-x^+ Xx^+ x^+"ly"

  - Another option: decompose words into features and use a maxent model along with Bayes' rule

$$P(w \mid t) = P_{MAXENT}(t \mid w)P(w) / P(t)$$

Insufficient data

- Zero probability $p(t \mid M_d) = 0$
  - May not wish to assign a probability of zero to a document that is missing one or more of the query terms [gives conjunction semantics]
- General approach $tf_{(t,d)} = 0 \quad p(t \mid M_d) = \frac{cf_t}{cs}$
  - A non-occurring term is possible, but no more likely than would be expected by chance in the collection.
  - If ,

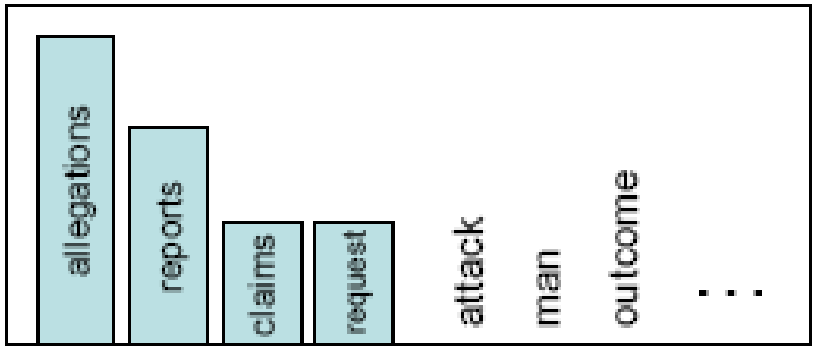
    cf_t : raw count of term t in the collection

    cs : raw collection size(total number of tokens in the collection)

Smoothing

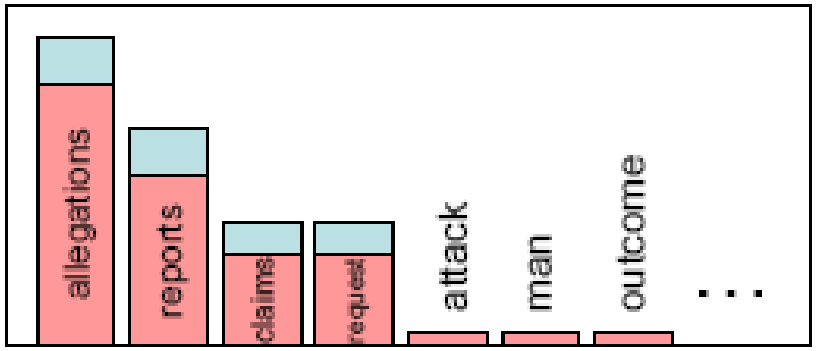
- We often want to make estimates from sparse statistics:

P(w | denied the)
3 allegations
2 reports
1 claims
1 request
7 total



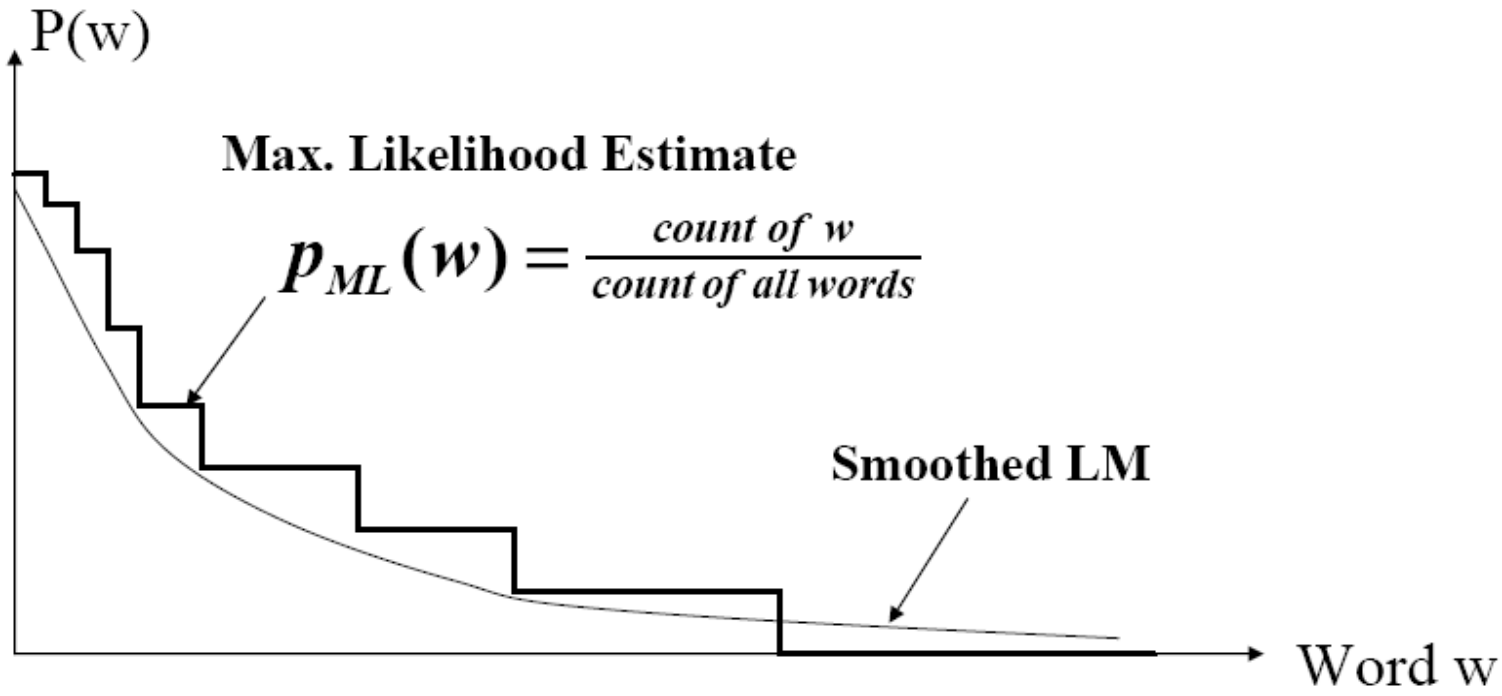
- Smoothing flattens spiky distributions so they generalize better

P(w | denied the)
2.5 allegations
1.5 reports
0.5 claims
0.5 request
2 other
7 total



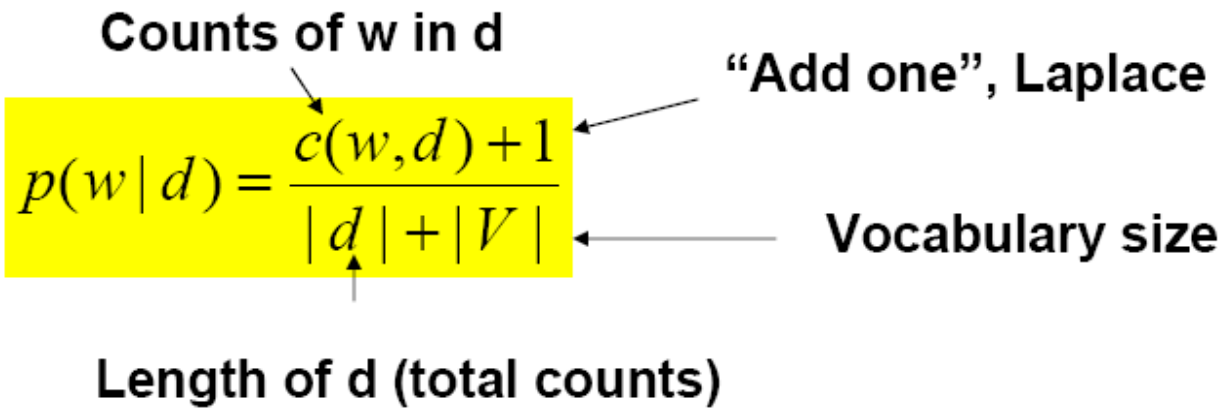
- Very important all over NLP, but easy to do badly!
- We'll illustrate with bigrams today (h = previous word, could be anything).

Language Model Smoothing (Illustration)



How to Smooth?

- All smoothing methods try to
  - **discount the probability of words seen in a document**
  - **re-allocate the extra counts so that unseen words will have a non-zero count**
- Method 1 Additive smoothing [Chen & Goodman 98]: **Add a constant δ to the counts of each word, e.g., "add 1"**



Smoothing

- Estimating multinomials
- We want to know what words follow some history h
  - There's some true distribution P(w | h)
  - 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
- Two issues:
  - Discounting: how to reserve mass what we haven't seen
  - Interpolation: how to allocate that mass amongst unseen events

Laplace smoothing

- Idea: pretend we saw every word once more than we actually did [Laplace]
  - Corresponds to a uniform prior over vocabulary
  - Think of it as taking items with observed count r > 1 and treating them as having count r* < r
  - Holds out V/(N+V) for "fake" events
  - N1+/N of which is distributed back to seen words
  - N0/(N+V) actually passed on to unseen words (nearly all!)
  - Actually tells us not only how much to hold out, but where to put it
- Works **poorly** in practice
- Quick fix: add some small δ instead of 1 [Lidstone, Jefferys]
- Slightly better, holds out less mass, still a bad idea

How Much to subtract?

- Remember the key discounting problem:
  - What count should r* should we use for an event that occurred r times in N samples?
  - r is too big
- Idea: held-out data [Jelinek and Mercer]
  - Get another N samples
  - See what the average count of items occurring r times is (e.g. doubletons on average might occur 1.78 times)
  - Use those averages as r*
  - Much better than add-one, etc.

Smoothing add-one summary

c	number of word tokens in training data
$c(w)$	count of word w in training data
$c(w,w_{-1})$	count of word w following word w_{-1}
V	total vocabulary size
N_k	number of word types with count k

- One class of smoothing functions:
  - Add-one / delta: assumes a uniform prior

$$P_{ADD-\delta}(w \mid w_{-1}) = \frac{c(w,w_{-1}) + \delta(1/V)}{c(w_{-1}) + \delta}$$

  - Better to assume a unigram prior

$$P_{UNI-PRIOR}(w \mid w_{-1}) = \frac{c(w,w_{-1}) + \delta\hat{P}(w)}{c(w_{-1}) + \delta}$$

Smoothing → Word Classes

$$p(x_1 \ldots x_n, y_1 \ldots y_n) = q(STOP|y_n) \prod_{i=1}^{n} q(y_i|y_{i-1}) e(x_i|y_i)$$

- Typically, linear interpolation works well for transitions

$$q(y_i|y_{i-1}) = \lambda_1 q_{ML}(y_i|y_{i-1}) + \lambda_2 q_{ML}(y_i)$$

- However, other approaches used for emissions
  - Step 1: Split the vocabulary
    - *Frequent words:* appear more than M (often 5) times
    - *Low frequency:* everything else
  - Step 2: Map each low frequency word to one of a small, finite set of possibilities
    - For example, based on prefixes, suffixes, etc.
  - Step 3: Learn model for this new space of possible word sequences

Low Frequency Words: An Example

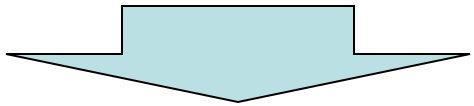
Named Entity Recognition [Bickel et. al, 1999]

- Used the following word classes for infrequent words:

Word class	Example	Intuition
twoDigitNum	90	Two digit year
fourDigitNum	1990	Four digit year
containsDigitAndAlpha	A8956-67	Product code
containsDigitAndDash	09-96	Date
containsDigitAndSlash	11/9/89	Date
containsDigitAndComma	23,000.00	Monetary amount
containsDigitAndPeriod	1.00	Monetary amount,percentage
othernum	456789	Other number
allCaps	BBN	Organization
capPeriod	M.	Person name initial
firstWord	first word of sentence	no useful capitalization information
initCap	Sally	Capitalized word
lowercase	can	Uncapitalized word
other	,	Punctuation marks, all other words

Low Frequency Words: An Example

- Profits/NA soared/NA at/NA Boeing/SC Co./CC ,/NA easily/NA topping/NA forecasts/NA on/NA Wall/SL Street/CL ,/NA as/NA their/NA CEO/NA Alan/SP Mulally/CP announced/NA first/NA quarter/NA results/NA ./NA



- *firstword*/NA soared/NA at/NA *initCap*/SC Co./CC ,/NA easily/NA *lowercase*/NA forecasts/NA on/NA *initCap*/SL Street/CL ,/NA as/NA their/NA CEO/NA Alan/SP *initCap*/CP announced/NA first/NA quarter/NA results/NA ./NA

NA = No entity
SC = Start Company
CC = Continue Company
SL = Start Location
CL = Continue Location
…

Summary: HMM Inference and Learning

- Learning
  - Maximum likelihood: transitions q and emissions e

$$p(x_1 \dots x_n, y_1 \dots y_n) = q(STOP|y_n) \prod_{i=1}^{n} q(y_i|y_{i-1}) e(x_i|y_i)$$

- Inference (linear time in sentence length!)
  - Viterbi:

$$y^* = \arg \max_{y_1 \dots y_n} p(x_1 \dots x_n, y_1 \dots y_n)$$