CS 188: Artificial Intelligence
Fall 2011

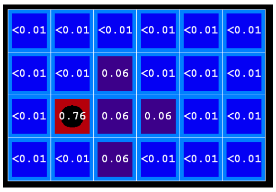
Lecture 20: HMMs / Speech / ML
11/8/2011

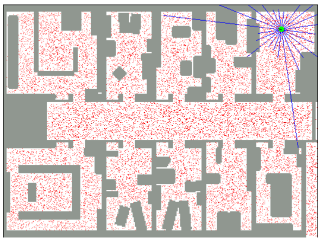
Dan Klein – UC Berkeley

Today

- HMMs
  - Demo bonanza!
  - Most likely explanation queries
- Speech recognition
  - A massive HMM!
  - Details of this section not required
- Start machine learning

Demo Bonanza!

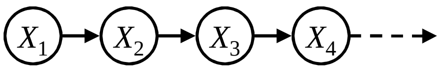




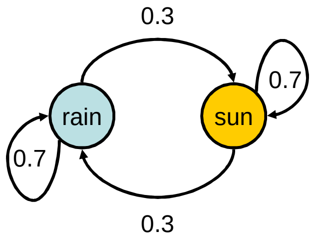
[DEMO: Stationary]

Recap: Reasoning Over Time

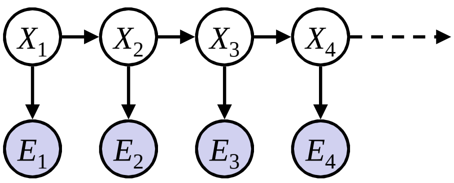
- Markov models



$P(X_1)$ $\qquad$ $P(X|X_{-1})$



- Hidden Markov models



$P(E|X)$

X	E	P
rain	umbrella	0.9
rain	no umbrella	0.1
sun	umbrella	0.2
sun	no umbrella	0.8

[DEMO: Exact Filtering]

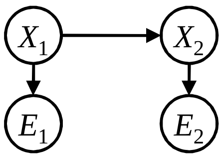
Recap: Filtering

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

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

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

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



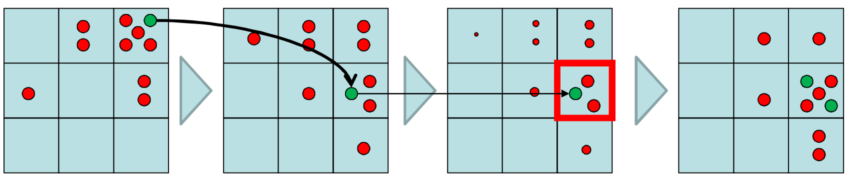
	Belief: <P(rain), P(sun)>	
$P(X_1)$	<0.5, 0.5>	*Prior on X_1*
$P(X_1 \mid E_1 = umbrella)$	<0.82, 0.18>	*Observe*
$P(X_2 \mid E_1 = umbrella)$	<0.63, 0.37>	*Elapse time*
$P(X_2 \mid E_1 = umb, E_2 = umb)$	<0.88, 0.12>	*Observe*

[DEMO: Localization / SLAM]

Recap: Particle Filtering

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

Elapse Weight Resample

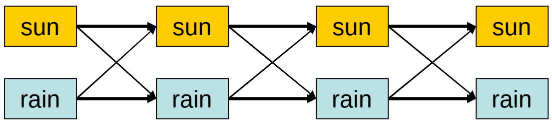


Particles:	Particles:	Particles:	(New) Particles:
(3,3)	(3,2)	(3,2) w=.9	(3,2)
(2,3)	(2,3)	(2,3) w=.2	(2,2)
(3,3)	(3,2)	(3,2) w=.9	(3,2)
(3,2)	(3,1)	(3,1) w=.4	(2,3)
(3,3)	(3,3)	(3,3) w=.4	(3,3)
(3,2)	(3,2)	(3,2) w=.9	(3,2)
(1,2)	(1,3)	(1,3) w=.1	(1,3)
(3,3)	(2,3)	(2,3) w=.2	(2,3)
(3,3)	(3,2)	(3,2) w=.9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

7

State Trellis

- State trellis: graph of states and transitions over time

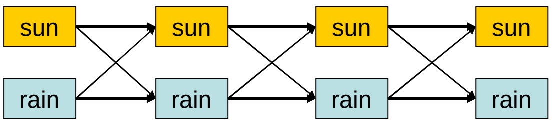


X_1 X_2 $\cdots$ X_N

- 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 the seq's probability
- Can think of the Forward (and now Viterbi) algorithms as computing sums of all paths (best paths) in this graph

8

Forward / Viterbi Algorithms



X_1 X_2 $\cdots$ X_N

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

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

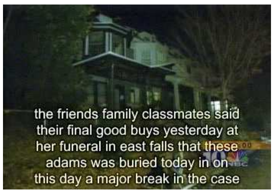
$$m_t[x_t] = \max_{x_{1:t-1}} P(x_{1:t-1}, x_t, e_{1:t})$$

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

9

Speech and Language

- Speech technologies
  - Automatic speech recognition (ASR)
  - Text-to-speech synthesis (TTS)
  - Dialog systems



- Language processing technologies
  - Machine translation

"Il est impossible aux journalistes de rentrer dans les régions tibétaines"

Bruno Philip, correspondant du "Monde" en Chine, estime que les journalistes de l'AFP qui ont été expulsés de la province tibétaine du Qinghai "n'étaient pas dans l'illégalité".

Les faits Le dalaï-lama dénonce l'"enfer" imposé au Tibet depuis sa fuite, en 1959

Vidéo Anniversaire de la rébellion

"It is impossible for journalists to enter Tibetan areas"

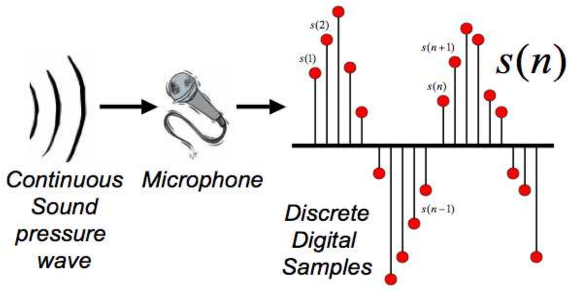
Philip Bruno, correspondent for "World" in China, said that journalists of the AFP who have been deported from the Tibetan province of Qinghai "were not illegal."

Facts The Dalai Lama denounces the "hell" imposed since he fled Tibet in 1959

Video Anniversary of the Tibetan rebellion: China on guard

  - Information extraction
  - Web search, question answering
  - Text classification, spam filtering, etc…

Digitizing Speech

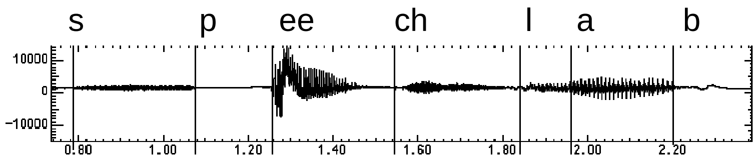


Thanks to Bryan Pellom for this slide!

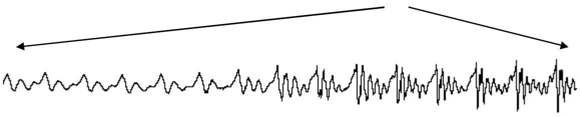
11

Speech in an Hour

- Speech input is an acoustic wave form



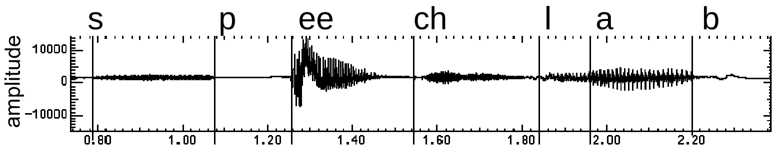
"l" to "a" transition:



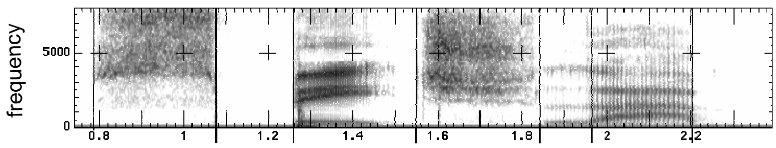
Graphs from Simon Arnfield's web tutorial on speech, Sheffield:
http://www.psyc.leeds.ac.uk/research/cogn/speech/tutorial/

Spectral Analysis

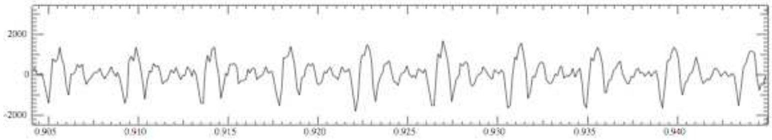
- Frequency gives pitch; amplitude gives volume
  - sampling at ~8 kHz phone, ~16 kHz mic (kHz=1000 cycles/sec)



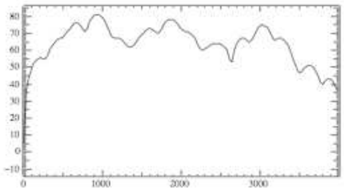
- Fourier transform of wave displayed as a spectrogram
  - darkness indicates energy at each frequency



Part of [ae] from "lab"



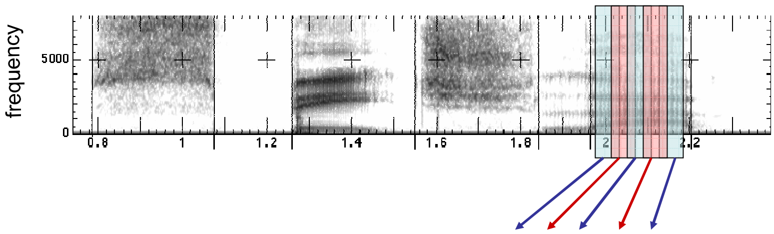
- Complex wave repeating nine times
  - Plus smaller wave that repeats 4x for every large cycle
  - Large wave: freq of 250 Hz (9 times in .036 seconds)
  - Small wave roughly 4 times this, or roughly 1000 Hz



[demo]

Acoustic Feature Sequence

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



……………………………………………….$e_{12}e_{13}e_{14}e_{15}e_{16}$……….

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

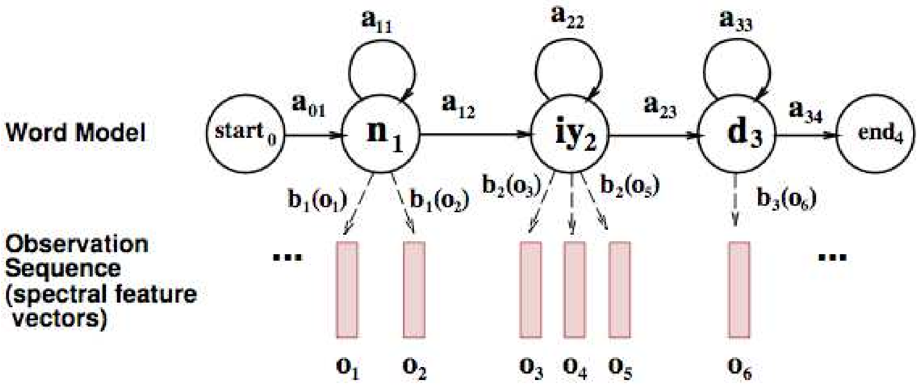
State Space

- 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
- We will have one state for each sound in each word
- From some state x, can only:
  - Stay in the same state (e.g. speaking slowly)
  - Move to the next position in the word
  - At the end of the word, move to the start of the next word
- We build a little state graph for each word and chain them together to form our state space X

16

HMMs for Speech



17

Transitions with Bigrams

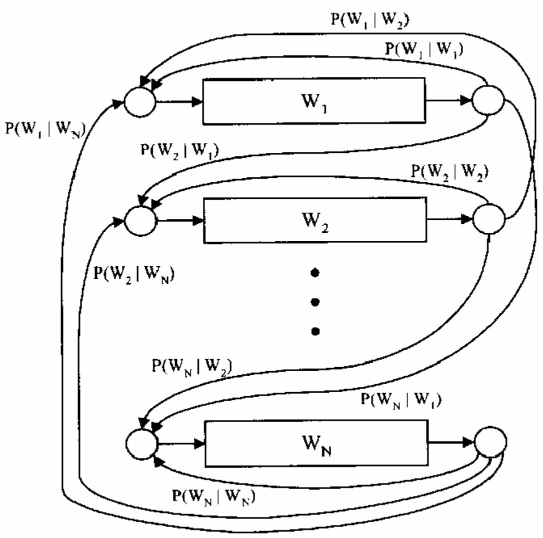


Figure from Huang et al page 618

Training Counts

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

23135851162 the *

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

$$= 0.0006$$

Decoding

- While there are some practical issues, finding the words given the acoustics is an HMM inference problem
- We want to know which state sequence $x_{1:T}$ is most likely given the evidence $e_{1:T}$:

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

- From the sequence x, we can simply read off the words

19

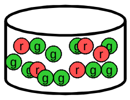
End of Part II!

- Now we're done with our unit on probabilistic reasoning
- Last part of class: machine learning

Machine Learning

- Up until now: how to reason in a model and how to make optimal decisions
- Machine learning: how to acquire a model on the basis of data / experience
  - Learning parameters (e.g. probabilities)
  - Learning structure (e.g. BN graphs)
  - Learning hidden concepts (e.g. clustering)

Parameter Estimation



- Estimating the distribution of a random variable
- *Elicitation:* ask a human (why is this hard?)
- *Empirically:* use training data (learning!)
  - E.g.: for each outcome x, look at the *empirical rate* of that value:

$$P_{\mathrm{ML}}(x) = \frac{\mathrm{count}(x)}{\text{total samples}}$$

$$P_{\mathrm{ML}}(\mathrm{r}) = 2/3$$

  - This is the estimate that maximizes the *likelihood of the data*

$$L(x,\theta) = \prod_i P_\theta(x_i)$$

Estimation: Smoothing

- Relative frequencies are the maximum likelihood estimates

$$\begin{aligned}\theta_{ML} &= \arg\max_\theta P(\mathbf{X}|\theta)\\ &= \arg\max_\theta \prod_i P_\theta(X_i)\end{aligned} \quad \Rightarrow \quad P_{\mathrm{ML}}(x) = \frac{\mathrm{count}(x)}{\text{total samples}}$$

- In Bayesian statistics, we think of the parameters as just another random variable, with its own distribution

$$\begin{aligned}\theta_{MAP} &= \arg\max_\theta P(\theta|\mathbf{X})\\ &= \arg\max_\theta P(\mathbf{X}|\theta)P(\theta)/P(\mathbf{X}) \quad \Rightarrow \quad ????\\ &= \arg\max_\theta P(\mathbf{X}|\theta)P(\theta)\end{aligned}$$

Estimation: Laplace Smoothing

- Laplace's estimate:
  - Pretend you saw every outcome once more than you actually did

H H T

$$P_{LAP}(x) = \frac{c(x)+1}{\sum_x [c(x)+1]}$$

$$= \frac{c(x)+1}{N+|X|}$$

$$P_{ML}(X) =$$

$$P_{LAP}(X) =$$

  - Can derive this estimate with *Dirichlet priors* (see cs281a)

Estimation: Laplace Smoothing

- Laplace's estimate (extended):
  - Pretend you saw every outcome k extra times

H H T

$$P_{LAP,k}(x) = \frac{c(x)+k}{N+k|X|}$$

  - What's Laplace with k = 0?
  - k is the strength of the prior

$$P_{LAP,0}(X) =$$

$$P_{LAP,1}(X) =$$

$$P_{LAP,100}(X) =$$

- Laplace for conditionals:
  - Smooth each condition independently:

$$P_{LAP,k}(x|y) = \frac{c(x,y)+k}{c(y)+k|X|}$$