Announcements

- Written 6 due tonight
- Project 4 up!
  - Due 4/15 – start early!
- Course contest update
  - Planning to post by Friday night
P4: Ghostbusters

- **Plot**: Pacman's grandfather, Grandpac, learned to hunt ghosts for sport.

- He was blinded by his power, but could hear the ghosts' banging and clanging.

- **Transition Model**: All ghosts move randomly, but are sometimes biased

- **Emission Model**: Pacman knows a "noisy" distance to each ghost

---

Today

- **Dynamic Bayes Nets (DBNs)**
  - [sometimes called temporal Bayes nets]

- **HMMs**: Most likely explanation queries

- **Speech recognition**
  - A massive HMM!
  - Details of this section not required

- **Start machine learning**
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

- Variable elimination applies to dynamic Bayes nets.
- Procedure: “unroll” the network for $T$ time steps, then eliminate variables until $P(X_T|e_{1:T})$ is computed.
- Online belief updates: Eliminate all variables from the previous time step; store factors for current time only.
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: $G_1^a = (3,3)$  $G_1^b = (5,3)$
- **Elapse time**: Sample a successor for each particle
  - Example successor: $G_2^a = (2,3)$  $G_2^b = (6,3)$
- **Observe**: Weight each entire sample by the likelihood of the evidence conditioned on the sample
  - Likelihood: $P(E_1^a | G_1^a) \times P(E_1^b | G_1^b)$
- **Resample**: Select prior samples (tuples of values) in proportion to their likelihood

SLAM

- SLAM = Simultaneous Localization And Mapping
  - We do not know the map or our location
  - Our belief state is over maps and positions!
  - Main techniques: Kalman filtering (Gaussian HMMs) and particle methods

- [DEMOS]
  - [intel-lab-raw-odo.wmv, intel-lab-scan-matching.wmv, visionSlam_heliOffice.wmv]
Today

- Dynamic Bayes Nets (DBNs)
  - [sometimes called temporal Bayes nets]

- HMMs: Most likely explanation queries

- Speech recognition
  - A massive HMM!
  - Details of this section not required

- Start machine learning

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"
  "It is impossible for journalists to enter Tibetan areas"

- Information extraction
  - Web search, question answering
  - Text classification, spam filtering, etc…
HMMs: MLE Queries

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

- Query: most likely explanation:
  $$\arg \max_{x_{1:t}} P(x_{1:t}|e_{1:t})$$

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

\[
x_{1:T}^* = \arg\max_{x_{1:T}} P(x_{1:T}|e_{1:T}) = \arg\max_{x_{1:T}} P(x_{1:T}, e_{1:T})
\]

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

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

\[
= P(e_t|x_t)\max_{x_{t-1}} P(x_t|x_{t-1})\max_{x_{1:t-2}} P(x_{1:t-1}, e_{1:t-1})
\]

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

Example

\[
P_1(\text{rain | true}) \cdot P(e_1|\text{rain | true}) \cdot 0.515 = 0.8 \cdot 0.182 \cdot 0.0491 = 0.0361
\]

\[
P_1(\text{rain | false}) \cdot P(e_1|\text{rain | false}) \cdot 0.515 = 0.8 \cdot 0.918 \cdot 0.1233 = 0.0734
\]

\[
P_1(\text{true | false}) \cdot P(e_1|\text{true | false}) \cdot 0.0210 = 0.8 \cdot 0.0173 \cdot 0.0210 = 0.0018
\]
Today

- Dynamic Bayes Nets (DBNs)
  - [sometimes called temporal Bayes nets]

- HMMs: Most likely explanation queries

- *Speech recognition*
  - A massive HMM!
  - Details of this section not required

- Start machine learning

Digitizing Speech

Thanks to Bryan Plllorn for this slide!
Speech in an Hour

- Speech input is an acoustic wave form

```
s p ee ch l a b
```

“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)

```
s p ee ch l a b
```

- Fourier transform of wave displayed as a spectrogram
  - darkness indicates energy at each frequency
Adding 100 Hz + 1000 Hz Waves

Spectrum

Frequency components (100 and 1000 Hz) on x-axis
Part of [ae] from “lab”

- Note complex wave repeating nine times in figure
- Plus smaller waves which repeats 4 times for every large pattern
- Large wave has frequency of 250 Hz (9 times in .036 seconds)
- Small wave roughly 4 times this, or roughly 1000 Hz
- Two little tiny waves on top of peak of 1000 Hz waves

Back to Spectra

- Spectrum represents these freq components
- Computed by Fourier transform, algorithm which separates out each frequency component of wave.

- x-axis shows frequency, y-axis shows magnitude (in decibels, a log measure of amplitude)
- Peaks at 930 Hz, 1860 Hz, and 3020 Hz.
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.

![Diagram of vocal tract](image)

Length 17.5 cm.

Figure from W. Barry Speech Science slides
Acoustic Feature Sequence

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

- 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 $X$
HMMs for Speech

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
End of Part II!

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

Parameter Estimation

- Estimating the distribution of a random variable
- Elicitation: ask a human!
  - Usually need domain experts, and sophisticated ways of eliciting probabilities (e.g. betting games)
  - Trouble calibrating
- Empirically: use training data
  - For each outcome \( x \), look at the empirical rate of that value:
    \[
    P_{ML}(x) = \frac{\text{count}(x)}{\text{total samples}}
    \]
    
    \[
    P_{ML}(r) = \frac{1}{3}
    \]

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