Today

- Particle filter recap
- Speech recognition using HMMs
Particle Filtering Review

- An approximate technique for filtering: \( P( X_t \mid e_1, \ldots, e_t) \)
- Idea: always keep \( N \) guesses (samples) of the value of \( X_t \)
- Initial samples, or particles, are drawn from the prior \( P(X_1) \)
- Three operations:
  1) Elapse time: draw a sample for \( X_{t+1} \) from each particle using \( P(X_{t+1} \mid x_t) \)
  2) Observe: weight all particles by the likelihood of the evidence \( e_t \)
  3) Resample: sample new particles in proportion to those weights

Resampling Step Details

- Each particle is already weighted by the evidence likelihood: \( P(e_t \mid x_t) \)
- We randomly choose particles in proportion to those weights
- Probability of choosing a particle is proportional to its weight
- Each new particle is chosen independently, with replacement
- The probability of selecting a set of new particles is the product of probabilities for each one

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 particles

Hidden Markov Models

- An HMM is
  - Initial distribution: \( P(X_1) \)
  - Transitions: \( P(X_t \mid X_{t-1}) \)
  - Emissions: \( P(E_t \mid X_t) \)
- Query: most likely seq: \[ \arg\max_{x_{1:T}} P(x_{1:T} \mid e_{1:T}) \]

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 \( P(x_t \mid x_{t-1})P(e_t \mid x_t) \)
- 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

\[ x_{1:T}^* = \arg\max_{x_{1:T}} P(x_{1:T} \mid e_{1:T}) = \arg\max_{x_{1:T}} P(x_{1:T}, e_{1:T}) \]
\[ m_t[x_t] = \max_{x_{t-1}} P(x_{1:t-1}, x_t, e_{1:T}) \]
\[ = \max_{x_{1:t-1}} P(x_{1:t-1}, e_{1:t-1}) P(x_t \mid x_{t-1}) P(e_t \mid x_t) \]
\[ = P(e_t \mid x_t) \max_{x_{t-1}} P(x_{1:t-1}) m_{t-1}[x_{t-1}] \]
\[ = P(e_t \mid x_t) \max_{x_{t-1}} P(x_t \mid x_{t-1}) m_{t-1}[x_{t-1}] \]
Example

Digitizing Speech

Speech in an Hour

Spectral Analysis

Adding 100 Hz + 1000 Hz Waves

Speech input is an acoustic wave form

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

Spectral Analysis

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

Spectrum

Frequency components (100 and 1000 Hz) on x-axis

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

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
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.