Announcements

- **Written Assessment 1 Grades Released**
  - Regrade requests close today (7/20) at 11:59pm

- **Written Assessment 2 is out!**
  - Due **Friday 7/24 at 11:59pm**

- **Homework 5**
  - Due **Friday 7/24 at 11:59pm**

- **Project 4**
  - Due **Friday 7/31 at 11:59pm**
CS 188: Artificial Intelligence

HMMs, Particle Filters, and Applications

Instructor: Nikita Kitaev
University of California, Berkeley

[These slides were created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley. All CS188 materials are available at http://ai.berkeley.edu.]
Today

- **HMMs**
  - Particle filters
  - Demos!
  - Most-likely-explanation queries

- **Applications:**
  - Robot localization / mapping
  - Speech recognition (later)
Recap: Reasoning Over Time

- **Markov models**

  \[ P(X_1) \quad P(X|X_{-1}) \]

- **Hidden Markov models**

  \[
  \begin{array}{c}
  X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_4 \\
  E_1 \rightarrow E_2 \rightarrow E_3 \rightarrow E_4
  \end{array}
  \]

\[
\begin{array}{cccc}
X & E & P \\
\hline
\text{rain} & \text{umbrella} & 0.9 \\
\text{rain} & \text{no umbrella} & 0.1 \\
\text{sun} & \text{umbrella} & 0.2 \\
\text{sun} & \text{no umbrella} & 0.8 \\
\end{array}
\]
Video of Demo Ghostbusters Markov Model (Reminder)
Video of Ghostbusters Exact Filtering (Reminder)
**Inference: Base Cases**

\[
P(X_1|e_1)
\]

\[
P(x_1|e_1) = \frac{P(x_1, e_1)}{P(e_1)}
\]

\[
\propto_{X_1} P(x_1, e_1)
\]

\[
= P(x_1)P(e_1|x_1)
\]

\[
P(X_2)
\]

\[
P(x_2) = \sum_{x_1} P(x_1, x_2)
\]

\[
= \sum_{x_1} P(x_1)P(x_2|x_1)
\]
Inference: Base Cases

\[ P(X_2) \]

\[ P(x_2) = \sum_{x_1} P(x_1, x_2) \]

\[ = \sum_{x_1} P(x_1)P(x_2|x_1) \]
Assume we have current belief \( P(X | \text{evidence to date}) \)

\[ B(X_t) = P(X_t | e_{1:t}) \]

Then, after one time step passes:

\[
P(X_{t+1} | e_{1:t}) = \sum_{x_t} P(X_{t+1}, x_t | e_{1:t})
\]

\[
= \sum_{x_t} P(X_{t+1} | x_t, e_{1:t}) P(x_t | e_{1:t})
\]

\[
= \sum_{x_t} P(X_{t+1} | x_t) P(x_t | e_{1:t})
\]

Basic idea: beliefs get “pushed” through the transitions

- With the “B” notation, we have to be careful about what time step \( t \) the belief is about, and what evidence it includes
Example: Passage of Time

- As time passes, uncertainty “accumulates”

(Transition model: ghosts usually go clockwise)
Inference: Base Cases

\[ P(X_1 | e_1) \]

\[ P(x_1 | e_1) = P(x_1, e_1) / P(e_1) \]

\[ \propto_{X_1} P(x_1, e_1) \]

\[ = P(x_1) P(e_1 | x_1) \]
Assume we have current belief $P(X \mid \text{previous evidence})$:

$$B'(X_{t+1}) = P(X_{t+1} \mid e_{1:t})$$

Then, after evidence comes in:

$$P(X_{t+1} \mid e_{1:t+1}) = P(X_{t+1}, e_{t+1} \mid e_{1:t}) / P(e_{t+1} \mid e_{1:t})$$

$$\propto_{X_{t+1}} P(X_{t+1}, e_{t+1} \mid e_{1:t})$$

$$= P(e_{t+1} \mid e_{1:t}, X_{t+1}) P(X_{t+1} \mid e_{1:t})$$

$$= P(e_{t+1} \mid X_{t+1}) P(X_{t+1} \mid e_{1:t})$$

Or, compactly:

$$B(X_{t+1}) \propto_{X_{t+1}} P(e_{t+1} \mid X_{t+1}) B'(X_{t+1})$$

Basic idea: beliefs “rewighted” by likelihood of evidence

Unlike passage of time, we have to renormalize
As we get observations, beliefs get reweighted, uncertainty “decreases”

\[ B(X) \propto P(e|X)B'(X) \]
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: $\langle P(\text{rain}), P(\text{sun}) \rangle$

\[
\begin{align*}
P(X_1) & \quad <0.5, 0.5>  & \text{Prior on } X_1 \\
P(X_1 | E_1 = \text{umbrella}) & \quad <0.82, 0.18>  & \text{Observe} \\
P(X_2 | E_1 = \text{umbrella}) & \quad <0.63, 0.37>  & \text{Elapse time} \\
P(X_2 | E_1 = \text{umb}, E_2 = \text{umb}) & \quad <0.88, 0.12>  & \text{Observe}
\end{align*}
\]

[Demo: Ghostbusters Exact Filtering (L15D2)]
Particle Filtering

- Filtering: approximate solution

- Sometimes $|X|$ is too big to use exact inference
  - $|X|$ may be too big to even store $B(X)$
  - E.g. $X$ is continuous

- Solution: approximate inference
  - Track samples of $X$, not all values
  - Samples are called particles
  - Time per step is linear in the number of samples
  - But: number needed may be large
  - In memory: list of particles, not states

- This is how robot localization works in practice

- Particle is just new name for sample
Our representation of $P(X)$ is now a list of $N$ particles (samples)
- Generally, $N << |X|$
- Storing map from $X$ to counts would defeat the point

$P(x)$ approximated by number of particles with value $x$
- So, many $x$ may have $P(x) = 0!$
- More particles, more accuracy

For now, all particles have a weight of 1
Particle Filtering: Elapse Time

- Each particle is moved by sampling its next position from the transition model
  \[ x' = \text{sample}(P(X'|x)) \]
  - This is like prior sampling – samples’ frequencies reflect the transition probabilities
  - Here, most samples move clockwise, but some move in another direction or stay in place
- This captures the passage of time
  - If enough samples, close to exact values before and after (consistent)
Particle Filtering: Observe

- Slightly trickier:
  - Don’t sample observation, fix it
  - Similar to likelihood weighting, downweight samples based on the evidence

\[ w(x) = P(e|x) \]

\[ B(X) \propto P(e|X)B'(X) \]

- As before, the probabilities don’t sum to one, since all have been downweighted (in fact they now sum to \((N \text{ times})\) an approximation of \(P(e)\))
Particle Filtering: Resample

- Rather than tracking weighted samples, we resample.

- \( N \) times, we choose from our weighted sample distribution (i.e. draw with replacement).

- This is equivalent to renormalizing the distribution.

- Now the update is complete for this time step, continue with the next one.
Recap: Particle Filtering

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

![Diagram of Particle Filtering Process]

- **Elapse**: Update the particles based on the system dynamics.
- **Weight**: Assign weights to each particle based on the likelihood of the observation.
- **Resample**: Select new particles with replacement, proportional to their weights.

Particles:

1. (3,3)
2. (2,3)
3. (3,3)
4. (3,2)
5. (3,3)
6. (3,3)
7. (3,3)
8. (3,3)
9. (2,3)
10. (1,2)
11. (3,3)
12. (3,3)
13. (3,2)
14. (2,3)
15. (3,2)
16. (3,3)
17. (2,2)
18. (3,2)
19. (3,3)
20. (3,2)

Weights:

- (3,2) w=.9
- (2,3) w=.2
- (3,2) w=.9
- (3,1) w=.1
- (3,3) w=.4
- (3,2) w=.9
- (2,3) w=.2
- (3,2) w=.9
- (2,2) w=.4
- (3,2) w=.4

New Particles:

- (3,2)
- (2,2)
- (3,2)
- (2,3)
- (3,3)
- (3,2)
- (1,3)
- (2,3)
- (3,2)
- (2,2)
- (3,2)
- (3,2)

[Demos: ghostbusters particle filtering (L15D3,4,5)]
Video of Demo – Moderate Number of Particles
Video of Demo – One Particle
Video of Demo – Huge Number of Particles
In robot localization:

- We know the map, but not the robot’s position
- Observations may be vectors of range finder readings
- State space and readings are typically continuous (works basically like a very fine grid) and so we cannot store $B(X)$
- Particle filtering is a main technique
Particle Filter Localization (Sonar)

Global localization with sonar sensors

[Video: global-sonar-uw-annotated.avi]
Particle Filter Localization (Laser)
Robot Mapping

- **SLAM: Simultaneous Localization And Mapping**
  - We do not know the map or our location
  - State consists of position AND map!
  - Main techniques: Kalman filtering (Gaussian HMMs) and particle methods

[Demo: PARTICLES-SLAM-mapping1-new.avi]
Particle Filter SLAM – Video 1

[Demo: PARTICLES-SLAM-mapping1-new.avi]
Particle Filter SLAM – Video 2

[Demo: PARTICLES-SLAM-fastslam.avi]
Dynamic Bayes Nets
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 \( t-1 \)

- Dynamic Bayes nets are a generalization of HMMs

[Demo: pacman sonar ghost DBN model (L15D6)]
Pacman – Sonar (P4)

[Demo: Pacman – Sonar – No Beliefs(L14D1)]
Video of Demo Pacman Sonar Ghost DBN Model
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: $P(E_1^a \mid G_1^a) \times P(E_1^b \mid G_1^b)$

- **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)**

\[
m_t[x_t] = \max_{x_{1:t-1}} P(x_{1:t-1}, x_t, e_{1:t})
= P(e_t|x_t) \max_{x_{t-1}} P(x_t|x_{t-1}) m_{t-1}[x_{t-1}]
\]
Speech Recognition
Digitizing Speech
Speech in an Hour

- Speech input is an acoustic waveform

![Acoustic waveform diagram](http://www.psyc.leeds.ac.uk/research/cogn/speech/tutorial/)

“l” to “a” transition:

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)

- 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
Figure: Mark Liberman

Spectrum Shapes
Vowel [i] sung at successively higher pitches

Graphs: Ratrete 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 to form the state space $X$
States in a Word
Transitions with a Bigram Model

Training Counts

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

198015222 the first
194623024 the same
168504105 the following
158562063 the world
...
14112454 the door
-----------------------------------
23135851162 the *

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 $e_{1:T}$?

$$x^*_1:T = \arg \max_{x_{1:T}} P(x_{1:T} | e_{1:T}) = \arg \max_{x_{1:T}} P(x_{1:T}, e_{1:T})$$

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