Announcements

- If you want to discuss anything post-midterm, sign up for a **1:1 session** (link in Ed)
- **Project 4 checkpoint** due **today** (Nov 1) at 11:59pm PT
  - **Rest of Project 4** due **next Tuesday** (Nov 8) at 11:59pm PT
- **Homework 8** due **this Friday** (Nov 4) at 11:59pm PT
- **Discussion attendance credit** has been updated
- **No lecture on Thanksgiving week** (Nov 22). Enjoy a break!
CS 188: Artificial Intelligence
HMMs, Particle Filters, and Applications

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’s Topics

- Exact Inference in Hidden Markov Models (HMMs)
- Approximate Inference in HMMs via Particle Filtering
- Applications in Robot Localization and Mapping
- Brief overview of Dynamic Bayes Nets
Recap: Reasoning Over Time

- **Markov models**

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

- **Hidden Markov models**

  \[ P(E|X) \]

<table>
<thead>
<tr>
<th>X</th>
<th>E</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>rain</td>
<td>umbrella</td>
<td>0.9</td>
</tr>
<tr>
<td>rain</td>
<td>no umbrella</td>
<td>0.1</td>
</tr>
<tr>
<td>sun</td>
<td>umbrella</td>
<td>0.2</td>
</tr>
<tr>
<td>sun</td>
<td>no umbrella</td>
<td>0.8</td>
</tr>
</tbody>
</table>
HMM Inference: Find State Given Evidence

- We are given evidence at each time and want to know

\[ B_t(X) = P(X_t|e_1:t) \]

- Idea: start with \( P(X_1) \) and derive \( B_t \) in terms of \( B_{t-1} \)
  - equivalently, derive \( B_{t+1} \) in terms of \( B_t \)
Two Steps: Passage of Time + Observation

\[ B(X_t) = P(X_t | e_{1:t}) \quad B'(X_{t+1}) \]
Passage of Time: Base Case

\[ P(X_2) \]

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

\[ = \sum_{x_1} P(x_1)P(x_2|x_1) \]
Passage of Time: General Case

- Assume we have current belief \( P(X \mid \text{evidence to date}) \) and transition prob.

\[
B(X_t) = P(X_t \mid e_{1:t}) \quad P(X_{t+1} \mid x_t)
\]

- Then, after one time step passes:

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

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

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

- 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

\[
\text{Or compactly:} \\
B'(X_{t+1}) = \sum_{x_t} P(X_{t+1} \mid x_t) B(x_t)
\]
Example: Passage of Time

- As time passes, uncertainty “accumulates”

(Transition model: ghosts usually go clockwise)
Two Steps: Passage of Time + Observation

\[ B(X_t) = P(X_t|e_{1:t}) \quad B'(X_{t+1}) \]

\[ B(X_{t+1}) \]
Observation: Base Case

\[ P(X_1|e_1) \]

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

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

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

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

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 “reweighted” by likelihood of evidence

Unlike passage of time, we have to renormalize.
Example: Observation

- As we get observations, beliefs get reweighted, uncertainty “decreases”

\[ B(X) \propto P(e|X)B'(X) \]
Two Steps: Passage of Time + Observation

\[ B(X_t) = P(X_t|e_{1:t}) \quad B'(X_{t+1}) \]

\[ B(X_{t+1}) \]
Online Belief Updates

- Every time step, we start with current $P(X \mid \text{evidence})$
- We update for time:

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

- We update for evidence:

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

- This is our updated belief $B_t(X) = P(X_t|e_{1:t})$
- The forward algorithm does both at once (and doesn’t normalize)
The Forward Algorithm

- We are given evidence at each time and want to know

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

- We can derive the following updates

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

We can normalize as we go if we want to have \( P(x|e) \) at each time step, or just once at the end...

[Demo: Ghostbusters Exact Filtering (L15D2)]
Video of Ghostbusters Filtering
Online Belief Updates

- Every time step, we start with current $P(X \mid \text{evidence})$
- We update for time:
  \[
P(x_t|e_{1:t-1}) = \sum_{x_{t-1}} P(x_{t-1}|e_{1:t-1}) \cdot P(x_t|x_{t-1})
  \]
- We update for evidence:
  \[
P(x_t|e_{1:t}) \propto P(x_t|e_{1:t-1}) \cdot P(e_t|x_t)
  \]
- This is our updated belief $B_t(X) = P(X_t|e_{1:t})$
- The forward algorithm does both at once (and doesn’t normalize)
Today’s Topics

- Exact Inference in Hidden Markov Models (HMMs)
- Approximate Inference in HMMs via Particle Filtering
- Applications in Robot Localization and Mapping
- Brief overview of Dynamic Bayes Nets
Particle Filtering
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, $N \ll |X|$
- 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: Passage of 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)
- 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 down-weighted (in fact they now sum to (N times) an approximation of P(e))
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 showing the process of particle filtering including elapse, weight, and resample steps.](Demos: ghostbusters particle filtering (L15D3,4,5))
Video of Demo – One Particle
Video of Demo – Huge Number of Particles
Today’s Topics

- Exact Inference in Hidden Markov Models (HMMs)
- Approximate Inference in HMMs via Particle Filtering
- Applications in Robot Localization and Mapping
- Brief overview of Dynamic Bayes Nets
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

DP-SLAM, Ron Parr

[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]
Today’s Topics

- Exact **Inference** in Hidden Markov Models (HMMs)
- Approximate Inference in HMMs via **Particle Filtering**
- **Applications** in Robot Localization and Mapping
- Brief overview of **Dynamic Bayes Nets**
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
We’re done with Part II: Uncertainty!

We’ve seen how AI methods for:

- Representing uncertainty structure via Bayes Nets and multiple ways of doing inference
- Incorporating decision-making with uncertainty via Decision Nets
- Exploiting special structure of sequences / time via Markov Models and Hidden Markov Models and exact and approximate inference

Next up: Part III: Machine Learning!