Recap: Reasoning Over Time

• Markov models

\[ X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_4 \rightarrow \ldots \]

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

• Hidden Markov models

\[ X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_4 \rightarrow \ldots \]

\[ E_1 \rightarrow E_2 \rightarrow E_3 \rightarrow E_4 \]

\[ P(E|X) \]

<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>
Passage of Time

- Assume we have current belief $P(X \mid \text{evidence to date})$

  $$B(X_t) = P(X_t \mid e_{1:t})$$

- Then, after one time step passes:

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

- Or, compactly:

  $$B'(X_{t+1}) = \sum_{x_t} P(X' \mid x) B(x_t)$$

- 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”

\[
B'(X'|x) = \sum_x P(X'|x)B(x)
\]

Transition model: ghosts usually go clockwise
Example: Observation

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

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

$\text{True} \quad 0.500 \quad 0.500 \quad 0.627$

$\text{False} \quad 0.500 \quad 0.818 \quad 0.373$

$\text{Rain}_0 \quad \text{Rain}_1 \quad \text{Rain}_2$

$\text{Umbrella}_1 \quad \text{Umbrella}_2$
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

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

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

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

\[
P(x_t \mid e_{1:t-1}) = \sum_{x_{t-1}} P(x_{t-1} \mid e_{1:t-1}) \cdot P(x_t \mid x_{t-1})
\]

• We update for evidence:

\[
P(x_t \mid e_{1:t}) \propto_X P(x_t \mid e_{1:t-1}) \cdot P(e_t \mid x_t)
\]

• The forward algorithm does both at once (and doesn’t normalize)
• Problem: space is \( |X| \) and time is \( |X|^2 \) per time step
• Voice Recognition:
http://www.youtube.com/watch?v=d9gDcHBmr3I
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:** $\langle P(\text{rain}), P(\text{sun}) \rangle$

- $P(X_1) = \langle 0.5, 0.5 \rangle$  
  *Prior on $X_1$*

- $P(X_1 | E_1 = \text{umbrella}) = \langle 0.82, 0.18 \rangle$  
  *Observe*

- $P(X_2 | E_1 = \text{umbrella}) = \langle 0.63, 0.37 \rangle$  
  *Elapse time*

- $P(X_2 | E_1 = \text{umb}, E_2 = \text{umb}) = \langle 0.88, 0.12 \rangle$  
  *Observe*
Particle Filtering

- Sometimes $|X|$ is too big to use exact inference
  - $|X|$ may be too big to even store $B(X)$
  - E.g. $X$ is continuous
  - $|X|^2$ may be too big to do updates

- 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
Representation: Particles

• 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 $x$ will have $P(x) = 0!$
  – More particles, more accuracy

• For now, all particles have a weight of 1

Particles:
- (3,3)
- (2,3)
- (3,3)
- (3,2)
- (3,3)
- (3,2)
- (2,1)
- (3,3)
- (3,3)
- (2,1)
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 probs
  – Here, most samples move clockwise, but some move in another direction or stay in place

• This captures the passage of time
  – If we have enough samples, close to the exact values before and after (consistent)
Particle Filtering: Observe

- Slightly trickier:
  - Don’t do rejection sampling (why not?)
  - We don’t sample the observation, we fix it
  - This is similar to likelihood weighting, so we downweight our samples based on the evidence

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

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

- Note that, as before, the probabilities don’t sum to one, since most have been downweighted (in fact they sum to 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 analogous to renormalizing the distribution

- Now the update is complete for this time step, continue with the next one

Old Particles:
- (3,3) w=0.1
- (2,1) w=0.9
- (2,1) w=0.9
- (3,1) w=0.4
- (3,2) w=0.3
- (2,2) w=0.4
- (1,1) w=0.4
- (3,1) w=0.4
- (2,1) w=0.9
- (3,2) w=0.3

New Particles:
- (2,1) w=1
- (2,1) w=1
- (2,1) w=1
- (3,2) w=1
- (2,2) w=1
- (2,1) w=1
- (1,1) w=1
- (3,1) w=1
- (2,1) w=1
- (1,1) w=1
Robot Localization

• 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 often used

http://www.youtube.com/watch?v=kqJpuMNHF_g&feature=related

http://www.youtube.com/watch?v=INLja6Ya3Ig&feature=related
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

![Noisy distance prob](chart)