Recap: Reasoning Over Time

- **Stationary Markov models**
  \[ X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_4 \rightarrow \cdots \]
  
  \[ P(X_1) \quad P(X|X_{-1}) \]

- **Hidden Markov models**
  \[ X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_4 \rightarrow \cdots \]
  \[ P(X|E) \]

<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>

Filtering

Filtering is the inference process of finding a distribution over \( X_T \) given \( e_1 \) through \( e_T \): \( P(X_T | e_1 \cdots e_T) \)

- We first compute \( P(X_t | e_1) \) : \( P(x_t|e_1) \propto P(x_t) \cdot P(e_1|x_t) \)
- For each \( t \) from 2 to \( T \), we have \( P(X_t | e_1 \cdots e_{t-1}) \)
- **Elapsed time: compute** \( P(X_t | e_1 \cdots e_{t-1}) \)
  \[ P(x_t | e_1 \cdots e_{t-1}) = \sum_{x_{t-1}} P(x_{t-1} | e_1 \cdots e_{t-1}) \cdot P(x_t | x_{t-1}) \]

- **Observe:** compute \( P(X_t | e_1 \cdots e_t) = P(X_t | e_1) \)
  \[ P(x_t | e_1 \cdots e_t) \propto P(x_t | e_1 \cdots e_{t-1}) \cdot P(e_t | x_t) \]

HMM Conditional Independence

- HMMs have two important independence properties:
  - Past independent of future given the state of the present
  \[ X_2 \perp X_4 \mid X_3 \]
  \[ X_2 \perp X_4 \mid X_3 \]
  \[ X_2 \perp X_4 \mid X_3 \]
  \[ X_2 \perp X_4 \mid X_3 \]

Filtering: Umbrella Example

**Elapsed time: compute** \( P(X_t | e_1 \cdots e_t) \)
\[ P(x_t | e_1 \cdots e_{t-1}) = \sum_{x_{t-1}} P(x_{t-1} | e_1 \cdots e_{t-1}) \cdot P(x_t | x_{t-1}) \]

**Observe:** compute \( P(X_t | e_1 \cdots e_t) = P(X_t | e_1) \)
\[ P(x_t | e_1 \cdots e_t) \propto P(x_t | e_1 \cdots e_{t-1}) \cdot P(e_t | x_t) \]

<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>

Belief: \(-P(rain), P(sun)\)

- Prior on \( X_1 \): \( P(X_1) <0.5, 0.5> \)
- Observe \( E_1 = \) umbrella: \( P(X_1 | E_1 = \text{umbrella}) <0.82, 0.18> \)
- Observe \( E_1 = \) umbrella: \( P(X_2 | E_1 = \text{umbrella}) <0.63, 0.37> \)
- Elapse time: \( P(X_2 | E_1 = \text{umbrella}) <0.88, 0.12> \)
### The Forward Algorithm

- We are given evidence at each time and want to know
  \[ B_t(X) = P(X|e_{1:t}) \]
- We can derive the following updates
  \[ P(x_t|e_{1:t}) \propto P(x_t, e_{1:t}) \]
  \[ = \sum_{x_{t-1}} P(x_{t-1}, x_t, e_{1:t}) \]
  \[ = \sum_{x_{t-1}} P(x_{t-1}, e_{1:t-1}) P(x_t|x_{t-1}) P(e_t|x_t) \]
  \[ = P(e_t|x_t) \sum_{x_{t-1}} P(x_t|x_{t-1}) P(x_{t-1}, e_{1:t-1}) \]

### Online Belief Updates

- Every time step, we start with current \( P(X|\text{evidence}) \)
- 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:
  \[ P(x_t|e_{1:t}) \propto P(x_t|e_{1:t-1}) \cdot P(e_t|x_t) \]
- The forward algorithm does both at once (and doesn’t normalize)
- Problem: space is \(|X|^2|X|^2|X|^2|X|^2\) and time is \(|X|^2|X|^2|X|^2|X|^2\) per time step

### 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
- This is how robot localization works in practice

### 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 prob
  - 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:
  - 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 equivalent to renormalizing the distribution
- Now the update is complete for this time step, continue with the next one
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 a main technique

[Demo]

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 \( t-1 \)

\[
\begin{align*}
G_1^a & \quad E_1^a \\
G_1^b & \quad E_1^b \\
G_2^a & \quad E_2^a \\
G_2^b & \quad E_2^b \\
G_3^a & \quad E_3^a \\
G_3^b & \quad E_3^b
\end{align*}
\]

- Discrete valued dynamic Bayes nets are also HMMs

[Demo]

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: \( G_1^a = (3,3) \quad G_1^b = (5,3) \)
- Elapse time: Sample a successor for each particle
  - Example successor: \( G_2^a = (2,3) \quad G_2^b = (6,3) \)
- Observe: Weight each entire sample by the likelihood of the evidence conditioned on the sample
  - Likelihood: \( P(E_t^a | G_t^a) \times P(E_t^b | G_t^b) \)
- Resample: Select prior samples (tuples of values) in proportion to their likelihood

[Demo]

P4: Ghostbusters 2.0 (beta)

- 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

[Demo]

Exact Inference in DBNs

- Variable elimination applies to dynamic DBNs
- Procedure: “unroll” the network for \( T \) time steps, then eliminate variables until \( P(X_T | e_1:T) \) is computed

\[
\begin{align*}
G_1^a & \quad E_1^a \\
G_1^b & \quad E_1^b \\
G_2^a & \quad E_2^a \\
G_2^b & \quad E_2^b \\
G_3^a & \quad E_3^a \\
G_3^b & \quad E_3^b
\end{align*}
\]

- Online belief updates: Eliminate all variables from the previous time step; store factors for current time only

[Demo]

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

[DEMONS]