Hidden Markov Models

- Markov chains not so useful for most agents
  - Need observations to update your beliefs
- Hidden Markov models (HMMs)
  - Underlying Markov chain over states $X$
  - You observe outputs (effects) at each time step
  - As a Bayes's net (or more generally, a graphical model):

Example: Weather HMM

An HMM is defined by:
- Initial distribution: $P(X_1)$
- Transitions: $P(X_t | X_{t-1})$
- Emissions: $P(E_t | X_t)$

Ghostbusters HMM

- $P(X_t) =$ uniform
- $P(X'|X) =$ ghosts usually move clockwise, but sometimes move in a random direction or stay put
- $P(E|X) =$ same sensor model as before: red means close, green means far away.

Joint Distribution of an HMM

- Joint distribution:
  $$P(X_1, E_1, X_2, E_2, X_3, E_3, \ldots) = P(X_1)P(E_1|X_1)P(X_2|X_1)P(E_2|X_2)P(X_3|X_2)P(E_3|X_2)$$
- More generally:
  $$P(X_1, E_1, \ldots, X_T, E_T) = P(X_1)P(E_1|X_1) \prod_{t=2}^{T} P(X_t|X_{t-1})P(E_t|X_t)$$
- Questions to be resolved:
  - Does this indeed define a joint distribution?
  - Can every joint distribution be factored this way, or are we making some assumptions about the joint distribution by using this factorisation?
From the chain rule, every joint distribution over $X_1, E_1, X_2, E_2, X_3, E_3$ can be written as:

$$P(X_1, E_1, X_2, E_2, X_3, E_3) = P(X_1)P(E_1|X_1)P(X_2|X_1, E_1)P(E_2|X_1, E_1, X_2)$$

$$P(X_2|X_1, E_1, X_2, E_2)P(E_3|X_1, E_1, X_2, E_2)$$

Assuming that

$X_3 \perp E_1 \mid X_1, E_2 \perp X_1 \mid X_2, X_3 \perp X_1, E_1, E_2 \mid X_2, E_3 \perp X_1, E_1, X_2, E_2 \mid X_3$ gives us the expression posited on the previous slide:

$$P(X_1, E_1, X_2, E_2, X_3, E_3) = P(X_1)P(E_1|X_1)P(X_2|X_1, E_1)P(E_2|X_1, E_1, X_2)P(X_3|X_2, E_2)P(E_3|X_2, E_2)$$

HMMs have two important independence properties:

- Markov hidden process: future depends on past via the present
- Current observation independent of all else given current state

Quiz: does this mean that evidence variables are guaranteed to be independent?

[No, they are correlated by the hidden state(s)]
**Real HMM Examples**

- **Speech recognition HMMs:**
  - Observations are acoustic signals (continuous valued)
  - States are specific positions in specific words (so, tens of thousands)

- **Machine translation HMMs:**
  - Observations are words (tens of thousands)
  - States are translation options

- **Robot tracking:**
  - Observations are range readings (continuous)
  - States are positions on a map (continuous)

**HMM Computations**

- Given parameters and evidence $E_{1:t}$

- Inference problems include:
  - Filtering, find $P(X_t | e_{1:t})$ for all $t$
  - Smoothing, find $P(X_t | e_{1:n})$ for all $t$
  - Most probable explanation, find $x^*_{1:n} = \arg \max_{x_{1:n}} P(x_{1:n} | e_{1:n})$

**Filtering / Monitoring**

- Filtering, or monitoring, is the task of tracking the distribution $B_t(X) = P(X_t | e_1, ..., e_t)$ (the belief state) over time

- We start with $B_t(X)$ in an initial setting, usually uniform

- As time passes, or we get observations, we update $B(X)$

- The Kalman filter was invented in the 60's and first implemented as a method of trajectory estimation for the Apollo program
  - Kalman filter is a type of HMM with continuous values

**Example: Robot Localization**

$t=0$

Sensor model: can read in which directions there is a wall, never more than 1 mistake

Motion model: may not execute action with small prob.

$t=1$

Lighter grey: was possible to get the reading, but less likely b/c required 1 mistake

$t=2$
Passage of Time

- Assume we have current belief $P(X \mid \text{evidence to date})$
  
  
  - $B(X_t) = P(X_t \mid \text{evidence to date})$
  
  - Then, after one time step passes:
    
    $P(X_{t+1} \mid \text{evidence to date}) = \sum_{x_{t+1} \in X} P(X_{t+1} \mid x_t) P(x_t \mid \text{evidence to date})$
    
    - Or compactly:
      
      $B'(X_{t+1}) = \sum_{x_t} P(X_t \mid \text{evidence to date}) B(x_t)$

- 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 $\text{evidence to date}$ includes.

Example: Passage of Time

- As time passes, uncertainty "accumulates" (Transition model: ghosts usually go clockwise)
Video of Passage of Time (Transition Model)

Observation
- Assume we have current belief \( P(X | \text{previous evidence}) \):
  \[ B'(X_{t+1}) = P(X_{t+1} | e_{t+1}) \]
- Then, after evidence comes in:
  \[ P(X_{t+1} | e_{t+1}) = \frac{P(X_{t+1}, e_{t+1} | e_{t+1})}{P(e_{t+1})} = \frac{P(e_{t+1} | e_{t+1}, X_{t+1}) P(X_{t+1} | e_{t+1})}{P(e_{t+1})} = P(X_{t+1} | e_{t+1}) B(X_{t+1}) \]
- Or, compactly:
  \[ B(X_{t+1}) \propto_{X_{t+1}} P(e_{t+1} | X_{t+1}) B'(X_{t+1}) \]

Example: Observation
- As we get observations, beliefs get reweighted, uncertainty "decreases"

Example: Weather HMM

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

Online Belief Updates
- Every time step, we start with current P(X | evidence)
- We update for time:
  \[ P(x_t | e_{1:t-1}) = \sum_{e_{t-1}} P(e_{t-1} | e_{1:t-1}) \cdot P(x_t | x_{t-1}) \]
- We update for evidence:
  \[ P(x_t | e_{t+1}) \propto P(x_t | e_{t+1} | e_{t}) \cdot P(e_{t+1} | x_t) \]
- The forward algorithm does both at once (and doesn't normalize)
- Potential issue: space is |X| and time is |X|^2 per time step
HMM Computations (Reminder)

- Given
  - parameters
  - evidence $E_{1:n} \equiv e_{1:n}$

- Inference problems include:
  - Filtering, find $P(X_t|e_{1:t})$ for all $t$
  - Smoothing, find $P(X_t|e_{1:n})$ for all $t$
  - Most probable explanation, find $x^*_{1:n} = \arg\max_{x_{1:n}} P(x_{1:n}|e_{1:n})$

Smoothing

- Smoothing is the process of using all evidence better individual estimates for a hidden state (or all hidden states)
- Idea: run FORWARD algorithm up until $t$, and a similar BACKWARD algorithm from the final timestep $n$ down to $t$:

$$P(x_t|e_{1:n}) = \alpha P(x_t|e_{1:t}) P(e_{t+1:n}|x_t, e_{1:t})$$
$$= \alpha P(x_t|e_{1:t}) P(e_{t+1:n}|x_t)$$
$$= \alpha \Gamma_{t+1:t} b_{x_{t+1:n}}$$

Most Likely Explanation

HMMs: MLE Queries

- HMMs defined by
  - States $X$
  - Observations $E$
  - Initial distribution: $P(X_1)$
  - Transitions: $P(X|X_{t-1})$
  - Emissions: $P(E|X)$

- New query: most likely explanation: $\arg\max_{x_{1:n}} P(x_{1:n}|e_{1:n})$
- 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)
  \[
  f_t(x_t) = P(x_t, e_1:t) = \sum_{x_{t-1}} P(x_{t-1}, e_1:t) = P(x_t) \sum_{x_{t-1}} P(e_1:t|x_{t-1}, x_t)
  \]

- Viterbi Algorithm (Max)
  \[
  m_t(x_t) = \max_{x_{t-1}} P(x_{t-1}, e_1:t) = P(x_t) \max_{x_{t-1}} P(e_1:t|x_{t-1}, x_t)
  \]

**Most Probably Explanation (Sequence)**

- **Viterbi algorithm:** very similar to filtering algorithm (FORWARD)
- Essentially: replace “sum” with “max”, keep back pointers

<table>
<thead>
<tr>
<th>State Space Paths</th>
<th>Umbrella</th>
<th>Most Likely Paths</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rain 1</td>
<td>true</td>
<td>.8182</td>
</tr>
<tr>
<td>Rain 2</td>
<td>true</td>
<td>.5155</td>
</tr>
<tr>
<td>Rain 3</td>
<td>true</td>
<td>.0361</td>
</tr>
<tr>
<td>Rain 4</td>
<td>true</td>
<td>.0334</td>
</tr>
<tr>
<td>Rain 5</td>
<td>true</td>
<td>.0210</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>m_{1,1}</th>
<th>m_{1,2}</th>
<th>m_{1,3}</th>
<th>m_{1,4}</th>
<th>m_{1,5}</th>
</tr>
</thead>
<tbody>
<tr>
<td>.1819</td>
<td>.0491</td>
<td>.1237</td>
<td>.0173</td>
<td>.0024</td>
</tr>
</tbody>
</table>