Hidden Markov Models

- Markov chains not so useful for most agents
- Eventually you don’t know anything anymore
- Need observations to update your beliefs
- Hidden Markov models (HMMs)
- Underlying Markov chain over states \(S\)
- You observe outputs (effects) at each time step
- As a Bayes’ net:

![Diagram of Hidden Markov Model](image)

Example

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

Ghostbusters HMM

- \(P(K) = \text{uniform}\)
- \(P(K|X) = \text{ghosts usually move clockwise, but sometimes move in a random direction or stay put}\)
- \(P(E|X) = \text{same sensor model as before: red means close, green means far away}\)

HMM Computations

- Given
  - parameters
  - evidence \(E_{1:n}\)
- Inference problems include:
  - Filtering, find \(P(X_t|e_1:e_t)\) for all \(t\)
  - Smoothing, find \(P(X_t|e_1)\) for all \(t\)
  - Most probable explanation, find \(e_1:n = \arg\max_{e_1:n} P(e_1:n|X_{1:n})\)
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

Real HMM Examples

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

Conditional Independence

- 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

Conditional Independence

- Quiz: does this mean that observations are independent given no evidence?
  - No, correlated by the hidden state
Filtering / Monitoring

- Filtering, or monitoring, is the task of tracking the distribution $B(X)$ (the belief state) over time.
- We start with $B(X)$ in an initial setting, usually uniform.
- As time passes, or we get observations, we update $B(X)$.
- The Kalman filter (one method – real-valued values)
  - Invented in the 60's as a method of trajectory estimation for the Apollo program.

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.

$t=2$

$t=3$

$t=4$
**Example: Robot Localization**

![Robot Localization Diagram]

**Inference Recap: Simple Cases**

\[
P(X_1 | e_1) = \frac{P(X_1, e_1)}{P(e_1)}
\]

\[
P(X_2)
\]

**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:
  \[
P(x_t | e_{1:t-1}) \propto P(x_t | e_{1:t-1}) \cdot P(e_t | x_t)
\]
- The forward algorithm does both at once (and doesn’t normalize)
- Problem: space is \(|X|\) and time is \(|X|^t\) per time step

**Passage of Time**

- 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:
  \[
P(X_{t+1} | e_{1:t+1}) = \sum_x P(X_{t+1}, e_{1:t+1}) \cdot B(x)
\]
- Or, compactly:
  \[
  B(X') = \sum_x P(X' | e) \cdot B(x)
\]
- Basic idea: beliefs get "pushed" through the transitions
- With the "B" notation, we have to be careful about what time step the belief is about, and what evidence it includes

**Example: Passage of Time**

- As time passes, uncertainty “accumulates”

![Passage of Time Diagram]

**Observation**

- Assume we have current belief \( P(X | \text{previous evidence}) \):
  \( B'(X_{t+1}) = P(X_{t+1} | e_{1:t+1}) \)
- Then:
  \[
P(X_{t+1} | e_{1:t+1}) \propto P(e_{t+1} | X_{t+1}) \cdot P(X_{t+1} | e_{1:t})
\]
- Or:
  \[
  B(X_{t+1}) \propto P(e | X) \cdot B'(X_{t+1})
\]
- 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”

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

### The Forward Algorithm

- We want to know: \(B_t(X) = P(X_t|e_{1:t})\)
- We can derive the following updates:

\[
P(x_t|e_{1:t}) \propto \prod_{t=1}^T P(x_t|e_t)
\]

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

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

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

- To get \(B_t(X)\) compute each entry and normalize

### Example: Run the Filter

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

### Example HMM

- True
  - Rain_0
  - Rain_1
  - Rain_2
- False
  - Umbrella_0
  - Umbrella_1
  - Umbrella_2

### Summary: Filtering

- Filtering is the inference process of finding a distribution over \(X_t\) given \(e_t\) through \(e_T\):
  - \(P(X_t|e_{1:t})\)
  - We first compute \(P(X_1|e_1)\):
  - For each \(t\) from 2 to \(T\), we have \(P(X_t|e_{1:t-1})\)
  - 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:

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

### Intel Multi-Sensor Board

- Courtesy of T. Choudhury, G. Borriello

- Dieter Fox, University of Washington
New device

Dieter Fox, University of Washington

New device

Dieter Fox, University of Washington

Sensor board: Data Stream

Courtesy of T. Choudhury, G. Borriello

Activity Model

[UAI-06, ISER-06]

Environment

indoor, outdoor, vehicle

Activity

walk, run, stop, up/downstairs, drive, elevator, cover

Boosted classifier outputs

[Choudhury et al., UCAI-05]

After Action Review Evaluation (DARPA/NIST)
Best Explanation Queries

- Query: most likely seq:
  \[ \arg \max_{x_{1:t}} P(x_{1:t}|e_{1:t}) \]

State Path Trellis

- Each arc represents some transition
- Each arc has weight \( P(x_t|e_{t-1}) P(e_t|x_t) \)
- Each path is a sequence of states
- The product of weights on a path is the seq’s probability
- Can think of the Forward (and now Viterbi) algorithms as computing sums of all paths (best paths) in this graph

Viterbi Algorithm

\[
x_t^* = \arg \max_{\pi_{1:T}} P(x_{1:T}|e_{1:T}) = \arg \max_{\pi_{1:T}} P(x_{1:T}, e_{1:T})
\]

\[
m_t[x_t] = \max_{\pi_{1:t}} P(x_{1:t-1}, x_t, e_{1:t})
\]

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

\[= P(e_t|x_t) \max_{\pi_{1:t-1}} P(x_{1:t-1}, e_{1:t-1}) m_{t-1}[x_{t-1}] \]

Example

Recap: Reasoning Over Time

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

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>rain</td>
<td>umbrella</td>
<td>0.9</td>
<td></td>
</tr>
<tr>
<td>rain</td>
<td>no umbrella</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>sun</td>
<td>umbrella</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>sun</td>
<td>no umbrella</td>
<td>0.8</td>
<td></td>
</tr>
</tbody>
</table>

- Hidden Markov models