CSE 40171: Artificial Intelligence

Reasoning Over Time: Hidden Markov Models
Homework #5 has been released. It is due at 11:59PM on 10/29.
CSE Seminar: 11/1
Amy Reibman
Purdue University

3:30pm
DBART 138

Video Quality in an Age of Universal Cameras
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
An HMM is defined by:

- Initial distribution: \( P(X_1) \)
- Transitions: \( P(X_t \mid X_{t-1}) \)
- Emissions: \( P(E_t \mid X_t) \)

\[
\begin{array}{c|c|c|c|c|c}
R_t & R_{t+1} & P(R_{t+1} \mid R_t) & R_t & U_t & P(U_t \mid R_t) \\
\hline
+r & +r & 0.7 & +r & +u & 0.9 \\
+r & -r & 0.3 & +r & -u & 0.1 \\
-r & +r & 0.3 & -r & +u & 0.2 \\
-r & -r & 0.7 & -r & -u & 0.8 \\
\end{array}
\]
Joint Distribution of an HMM

Joint Distribution:

\[ P(X_1, E_1, X_2, E_2, X_3, E_3) = P(X_1)P(E_1|X_1)P(X_2|X_1)P(E_2|X_2)P(X_3|X_2)P(E_3|X_3) \]

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 factorization?
From the chain rule, every joint distribution over $X_1, E_1, X_2, E_2, X_3, E_3$ can be written as:

$$P(X_1, E_1, X_2, E_2, X_3, E_3) = P(X_1)P(E_1|X_1)P(X_2|X_1, E_1)P(E_2|X_1, E_1, X_2)P(X_3|X_1, E_1, X_2, E_2)P(E_3|X_1, E_1, X_2, E_2, X_3)$$
Chain Rule and HMMs

Assuming that:

\[ X_2 \perp E_1 \mid X_1, \quad E_2 \perp X_1, E_1 \mid X_2, \quad X_3 \perp X_1, E_1, E_2 \mid X_2, \quad E_3 \perp X_1, E_1, X_2, E_2 \mid X_3 \]

Gives us the expression we’ve seen previously:

\[
P(X_1, E_1, X_2, E_2, X_3, E_3) = P(X_1)P(E_1 \mid X_1)P(X_2 \mid X_1)P(E_2 \mid X_2)P(X_3 \mid X_2)P(E_3 \mid X_3)
\]
From the chain rule, every joint distribution over $X_1, E_1, \ldots, X_T, E_T$ can be written as:

$$P(X_1, E_1, \ldots, X_T, E_T) = P(X_1)P(E_1|X_1) \prod_{t=2}^{T} P(X_t|X_1, E_1, \ldots, X_{t-1}, E_{t-1})P(E_t|X_1, E_1, \ldots, X_{t-1}, E_{t-1}, X_t)$$
Chain Rule and HMMs

Assuming that for all $t$:

State independent of all past states and all past evidence given the previous state, i.e.:

$$X_t \perp X_1, E_1, \ldots, X_{t-2}, E_{t-2}, E_{t-1} \mid X_{t-1}$$

Evidence is independent of all past states and all past evidence given the current state, i.e.:

$$E_t \perp X_1, E_1, \ldots, X_{t-2}, E_{t-2}, X_{t-1}, E_{t-1} \mid X_t$$

Gives us the the expression we’ve seen previously:

$$P(X_1, E_1, \ldots, X_T, E_T) = P(X_1)P(E_1|X_1) \prod_{t=2}^{T} P(X_t|X_{t-1})P(E_t|X_t)$$
Implied Conditional Independencies

Many implied conditional independencies, e.g.,

\[ E_1 \perp X_2, E_2, X_3, E_3 \mid X_1 \]

To prove them:

- Approach 1: follow similar (algebraic) approach to what we did in the Markov models lecture
- Approach 2: directly from the graph structure

Intuition: If path between \( U \) and \( V \) goes through \( W \), then \( U \perp V \mid W \)
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)
Filtering / Monitoring

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

- We start with $B_1(X)$ in an initial setting, usually uniform

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

- The Kalman filter was invented in the 1960s and first implemented 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.
Example: Robot Localization

$t = 1$

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

Slide credit: Dan Klein and Pieter Abbeel, UC Berkeley CS 188
Example: Robot Localization

$$t = 2$$

Slide credit: Dan Klein and Pieter Abbeel, UC Berkeley CS 188
Example: Robot Localization

$\text{Prob} \quad 0 \quad 1$

$t = 3$

Slide credit: Dan Klein and Pieter Abbeel, UC Berkeley CS 188
Example: Robot Localization

$t = 4$
Example: Robot Localization

\[ t = 5 \]
Inference: Base Cases

\[ P(X_1|e_1) \]

\[ P(x_1|e_1) = P(x_1, e_1)/P(e_1) \]

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

\[ = P(x_1)P(e_1|x_1) \]
Inference: Base Cases

\[ P(X_2) \]

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

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

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:

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

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

$$= \sum_{x_t} P(X_{t+1}|x_t) P(x_t|e_{1:t})$$
Passage of Time

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

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

Or compactly:

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

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”

(Transition model: ghosts usually go clockwise)
Observation

Assume we have current belief $P(X|\text{previous evidence})$:

$$B'(X_{t+1}) = P(X_{t+1}|e_{1:t})$$

Then, after evidence comes in:

$$P(X_{t+1}|e_{1:t+1}) = \frac{P(X_{t+1}, e_{t+1}|e_{1:t})/P(e_{t+1}|e_{1:t})}{\alpha_{X_{t+1}} P(X_{t+1}, e_{t+1}|e_{1:t})}$$

$$= P(e_{t+1}|e_{1:t}, X_{t+1}) P(X_{t+1}|e_{1:t})$$

$$= P(e_{t+1}|X_{t+1}) P(X_{t+1}|e_{1:t})$$
Assume we have current belief $P(X | \text{previous evidence})$:

$$B'(X_{t+1}) = P(X_{t+1} | e_{1:t})$$

Or, compactly:

$$B(X_{t+1}) \propto_{X_{t+1}} P(e_{t+1} | X_{t+1}) B'(X_{t+1})$$

- Basic idea: beliefs “rewighted” by likelihood of evidence
- Unlike passage of time, we have to renormalize
Example: Observation

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

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

\[ P(R_{t+1} | R_t) \]

\[
\begin{align*}
R_t & \quad R_{t+1} & \quad P(R_{t+1} | R_t) \\
+r & \quad +r & \quad 0.7 \\
+r & \quad -r & \quad 0.3 \\
-r & \quad +r & \quad 0.3 \\
-r & \quad -r & \quad 0.7 \\
\end{align*}
\]

\[
\begin{align*}
R_t & \quad U_t & \quad P(U_t | R_t) \\
+r & \quad +u & \quad 0.9 \\
+r & \quad -u & \quad 0.1 \\
-r & \quad +u & \quad 0.2 \\
-r & \quad -u & \quad 0.8 \\
\end{align*}
\]

Slide credit: Dan Klein and Pieter Abbeel, UC Berkeley CS 188
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

\[
\begin{align*}
P(x_t|e_{1:t}) \propto_X & P(x_t, e_{1:t}) \\
& = \sum_{x_{t-1}} P(x_{t-1}, x_t, e_{1:t}) \\
& = \sum_{x_{t-1}} P(x_{t-1}, e_{1:t-1})P(x_t|x_{t-1})P(e_t|x_t) \\
& = P(e_t|x_t) \sum_{x_{t-1}} P(x_t|x_{t-1})P(x_{t-1}, e_{1:t-1})
\end{align*}
\]

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:

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

We update for evidence:

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

The forward algorithm does both at once (and doesn’t normalize)
Demo: Python Implementation

```python
states = ('rainy', 'sunny')
symbols = ('walk', 'shop', 'clean')

start_prob = {
    'rainy': 0.5,
    'sunny': 0.5
}

trans_prob = {
    'rainy': { 'rainy': 0.7, 'sunny': 0.3 },
    'sunny': { 'rainy': 0.4, 'sunny': 0.6 }
}

emit_prob = {
    'rainy': { 'walk': 0.1, 'shop': 0.4, 'clean': 0.5 },
    'sunny': { 'walk': 0.6, 'shop': 0.3, 'clean': 0.1 }
}

model = hmm.Model(states, symbols, start_prob, trans_prob, emit_prob)
```

https://github.com/jason2506/PythonHMM