L05 – Discrete Hidden Markov Models

AIMA 14.1–14.3; Barber 23.2

What you should know after this lecture

- What a hidden Markov model is and what it is good for
- What a recursive “filter” is, in this context
- How to solve inference problems in an HMM using sum-product and max-product

Aggregating information over time

- If the world is static, and you get different pieces of evidence over time, you can simply aggregate them via conditioning.
- But what if the world state could be changing over time?

Note that it doesn't matter what order the information arrives in!
Let $u(b, o)$ be a function that takes a belief $b = p(S)$ and an observation o, and returns a new belief $b' = p(S|o)$. Prove that $u(u(b, o_1), o_2) = u(u(b, o_2), o_1)$.
<u>Don't peek at next slide until you've done this!</u>

Belief update when state does not change

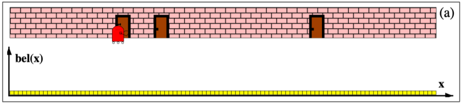
Prove that $u(u(b, o_1), o_2) = u(u(b, o_2), o_1)$.

$$\begin{aligned}
u(b, o_1)(s) &= \frac{p(o_1 \mid s)b(s)}{\sum_{s'} p(o_1 \mid s')b(s')} \\
u(u(b, o_1), o_2)(s) &= \frac{p(o_2 \mid s)u(b, o_1)(s)}{\sum_{s''} p(o_2 \mid s'')u(b, o_1)(s'')} \\
&= \frac{p(o_2 \mid s)p(o_1 \mid s)b(s)}{\sum_{s'} p(o_1 \mid s')b(s') \sum_{s''} p(o_2 \mid s'')u(b, o_1)(s'')} \\
&= \frac{p(o_2 \mid s)p(o_1 \mid s)b(s)}{\sum_{s'} p(o_1 \mid s')b(s') \frac{\sum_{s''} p(o_2|s'')p(o_1|s'')b(s'')}{\sum_{s'''} p(o_1|s''')b(s''')}} \\
&= \frac{p(o_2 \mid s)p(o_1 \mid s)b(s)}{\sum_{s''} p(o_2 \mid s'')p(o_1 \mid s'')b(s'')}
\end{aligned}$$

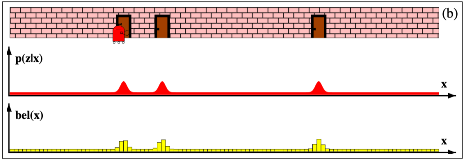
This is clearly the same as $u(u(b, o_2), o_1)$.

Robot Localization

- Robot initially has no idea where it is.



- Robot has a door detector. Intuitively (to us!) this gives the robot three possible locations it might be at.



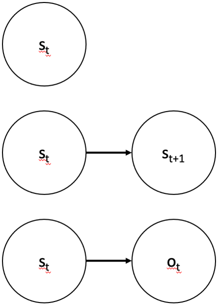
(Figure due to Thrun, Burgard and Fox, 2003, Probabilistic Robotics.)

- Concerns:
  - How do we represent those three locations? Is it really only three locations the robot can be at?
  - What happens when the robot moves?

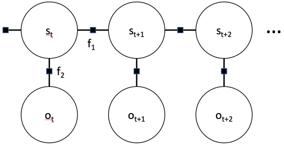
Robot Localization

- Let's say the state at each time t is s_t, where s_t is a random variable over the domain of locations (e.g., discrete locations in the hallway).
- Let's say the observations at each time step are o_t over the domain of {Door, No-Door}
- We need some way of linking the states at time s_t and s_{t+1} and the states to the observations. Our door detector might sometimes fail, and it'll fail in proportion to how close (or far) we are from the door. Also, when we try and move from state s_t to s_{t+1}, sometimes we'll stop short, sometimes we'll overshoot.
- Let's represent s_t and o_t as random variables, and assume that we know $P(s_{t+1}|s_t)$, $P(o_t|s_t)$ and a prior $P(s_0)$.

Resulting Graphical Model



Each of these subgraphs are the result of our assumed model of the world



When we merge common variables, we get this resulting factor graph.

Hidden Markov Models

- Idea first due to Baum (1966), used throughout communications field, summarized by Rabiner (1989)
- Defined by a model λ consisting of a tuple $\lambda = (\mathcal{S}, \mathcal{O}, A, B, \pi)$ that describe a discrete dynamical system as follows:
  - $\mathcal{S}$ is the set of hidden discrete states $\mathcal{S} = \{s(1), s(2), \ldots, s(n)\}$
  - $\mathcal{O}$ is the set of discrete observations $O = \{o(1), o(2), \ldots, o(m)\}$
  - $\mathbf{A} = \{A_{ij}\}$ is the dynamics or "transition" model, $A_{ij} = P(S_t = s(j) \mid S_{t-1} = s(i))$
  - $\mathbf{B} = \{B_{ik}\}$ is the measurement or "sensor" model, $B_{ik} = P(O_t = o(k) \mid S_t = s(i))$
  - π is the initial state distribution

Barber uses h for states, ν for observations. AIMA uses $\mathbf{X}$ for states, $\mathbf{e}$ for observations, and never uses a specific symbol for the transition or observation models. We are using s for states and o for observations to be consistent with the MDP and POMDP lectures coming up.

Dependencies in Hidden Markov Models

- Note that **A**, **B** and π are potentials of the form $\phi(\cdot)$ that we saw in last lecture.
- HMM defined by:
  - $S_{0:t-1} \perp\!\!\!\perp S_{t+1:T} \mid S_t$
  - $O_t \perp\!\!\!\perp (S_{0:t-1}, S_{t+1:T}) \mid S_t$
  - We have conditional distributions $P(O_t|S_t)$ and $P(S_{t+1}|S_t)$
  - First assumption known as the “Markov” assumption.

Three Questions

Three questions we might ask:

- Given the observation sequence $\mathbf{o}_{0:T} = \{o_0, o_1, \ldots, o_T\}$, how do we efficiently compute $P(S_T|\mathbf{o}_{0:T})$, the probability of the observation sequence given the model?
- Given the observation sequence $\mathbf{o}_{0:T} = \{o_0, o_1, \ldots, o_T\}$, how do we efficiently compute $P(o_{0:T})$, the probability of the observation sequence given the model?
- Given the observation sequence $\mathbf{o}_{0:T} = \{o_0, o_1, \ldots, o_T\}$, how do we find a corresponding state sequence $\mathbf{s}*_{0:T} = \{s_0^*, s_1^*, \ldots, s_T^*\}$ which is optimal in some meaningful sense (i.e., best "explains" the observations)?

Rabiner did not include our first question, but did include an important additional question about how to estimate A and B from data.

Filtering with Bayes Filter (Forward Algorithm)

Compute $P(s_T \mid o_{0:T})$

- Use sum-product, with S_T as the root of the tree.
- Conditioning on evidence $o_0, \dots, o_T$ effectively selects out a specific column of B for each time step.

Table multiplication and summing form:

$$P(S_0 \mid o_0) \propto \mu_{S_0 \to A^{01}} = \pi \cdot B_{o_0}$$

$$P(S_t \mid o_{0:t}) \propto \mu_{S_t \to A^{t,t+1}} = \sum_{S_{t-1}} \mu_{S_{t-1} \to A^{t-1,t}} \cdot A \cdot B_{o_t}$$

Matrix multiply form, where $\otimes$ is elementwise multiply and concatenation is matrix multiply:

$$P(S_0 \mid o_0) = \alpha_0 \propto \pi \otimes B_{o_0}$$

$$P(S_t \mid o_{0:t}) = \alpha_t \propto (\alpha_{t-1} A) \otimes B_{o_t}$$

Smoothing

More generally, compute, for all t, $P(s_t \mid o_{0:T})$
Forward-Backward algorithm = sum-product

- Do forward pass, computing α from left
- Do backward pass, computing β from the right

$$\begin{aligned}\beta_t &= \mu_{A^{t,t+1} \to S_t} \propto P(S_t \mid o_{t+1:T}) \\ &= \sum_{S_{t+1}} A \cdot B_{o_{t+1}} \cdot \mu_{A^{t+1,t+2} \to S_{t+1}} \\ &= \sum_{S_{t+1}} A \cdot B_{o_{t+1}} \cdot \beta_{t+1} \\ \beta_T &= \mathbf{1}\end{aligned}$$

- $P(S_t \mid o_{0:T}) \propto \alpha_t \beta_t$

Likelihood of observation sequence

The values α_T are equivalent to $P(o_{0:T}, S_T)$, so

$$P(o_{0:T}) = \sum_s P(o_{0:T}, S_T = s) = \sum_s \alpha_T[s]$$

Derivation:

$$\begin{aligned}
P(o_{0:T}) &= \sum_{s_{0:T}} P(o_{0:T} \mid s_{0:T}) P(s_{0:T}) \\
&= \sum_{s_{0:T}} \left(\prod_{t=0}^{T} P(o_t \mid s_t) \right) P(s_{0:T}) \\
&= \sum_{s_{0:T}} \left(\left(\prod_{t=2}^{T} P(o_t \mid s_t) \right) P(s_{2:T} \mid s_1) \Big(P(s_1 \mid s_0) P(o_1 \mid s_1) \Big) P(o_0 \mid s_0) P(s_0) \right) \\
&= \sum_{s_{0:T}} B_{s_T,o_T} \cdot A_{s_{T-1},s_T} \dots B_{s_1,o_1} \cdot A_{s_0,s_1} \cdot B_{s_0,o_0} \cdot \pi(s_0) \\
&= B_{s_T,o_T} \cdot \sum_{s_T} A_{s_{T-1},s_T} \dots \underbrace{B_{s_1,o_1} \cdot \underset{\substack{\downarrow \\ \text{Marginalizing out } s_0}}{\sum_{s_0}} A_{s_0,s_1} \cdot \underbrace{B_{s_0,o_0} \cdot \pi(s_0)}_{\alpha_0}}_{\alpha_1}
\end{aligned}$$

$$\Rightarrow \alpha_{t+1} = \left[\sum_{j=0}^{|S|} \alpha_t A_{i,j} \right] \cdot B_{i,o_{t+1}} \quad \text{And} \quad P(o_{0:T}) = \sum \alpha_T$$

Maximum likelihood state sequence

Compute $s^*_{0:T} = \mathsf{argmax}_{s_{0:T}} P(s_{0:T} \mid o_{0:T})$ using max-product algorithm!

- Forward pass to compute

$$\delta_t = \max_{s_{0:t}} P(s_{0:t} \mid o_{0:t}) = \max_{S_{t-1}} A \cdot \delta_{t-1} \cdot B_{o_t}$$

- Remember best s_{t-1} for each s

$$\psi_t(s) = \operatorname*{argmax}_{S_{t-1}} A \cdot \delta_{t-1} \cdot B_{o_t}$$

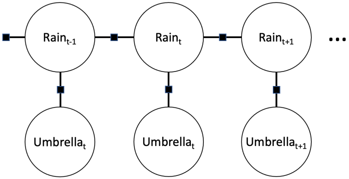
- Backtrace:

$$s^*_T = \operatorname*{argmax}_{S_T} \delta_T$$

$$s^*_t = \psi_{t+1}(s^*_{t+1})$$

Also called the Viterbi algorithm

Viterbi Decoding Example (from AIMA)

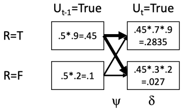


R_{t-1}	$P(R_{t-1})$
T	.5
F	.5

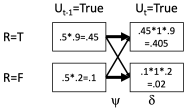
R_{t-1}	$P(R_t=T)$
T	.7
F	.3

R_t	$P(U_t=T)$
T	.9
F	.2

One step of Viterbi decoding would be:



If we set the $P(R_t = T|R_{t-1} = T) = 1$ and $P(R_t = F|R_{t-1} = F) = 1$, we get a slightly different graph:



- On the right, it might be the case that the $R_t = F$ really is the most likely state. How might this happen?

Next time

- Kalman filtering