CSE 473: Artificial Intelligence

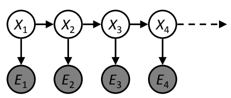
Hidden Markov Models

Dieter Fox --- University of Washington (Presented by Peter Henry)

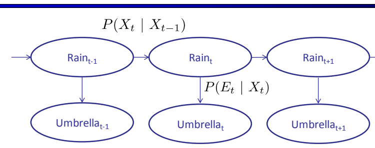
Hidden Markov Models

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: $P(X_t \mid X_{t-1})$
  - Emissions: $P(E_t \mid X_t)$

R_t	R_{t+1}	$P(R_{t+1} \mid R_t)$
+r	+r	0.7
+r	-r	0.3
-r	+r	0.3
-r	-r	0.7

R_t	U_t	$P(U_t \mid R_t)$
+r	+u	0.9
+r	-u	0.1
-r	+u	0.2
-r	-u	0.8

Ghostbusters HMM

- $P(X_1)$ = 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.

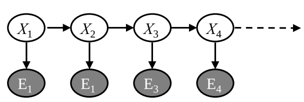
1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

$P(X_1)$

1/6	1/6	1/2
0	1/6	0
0	0	0

$P(X'|X=\langle 1,2 \rangle)$

Etc…

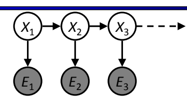


P(E|X)

P(red | 3)	P(orange | 3)	P(yellow | 3)	P(green | 3)
0.05	0.15	0.5	0.3

Etc… (must specify for other distances)

Joint Distribution of an HMM



- Joint distribution:

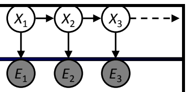
$$P(X_1, E_1, X_2, E_2, X_3, E_3) = P(X_1)P(E_1|X_1)P(X_2|X_1)P(E_2|X_2)P(X_3|X_2)P(E_3|X_3)$$

- 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?

Chain Rule and HMMs



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

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

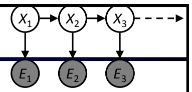
- *Assuming* that

$$X_2 \perp\!\!\!\perp E_1 \mid X_1, \quad E_2 \perp\!\!\!\perp X_1, E_1 \mid X_2, \quad X_3 \perp\!\!\!\perp X_1, E_1, E_2 \mid X_2, \quad E_3 \perp\!\!\!\perp X_1, E_1, X_2, E_2 \mid X_3$$

gives us the expression posited on the previous slide:

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

Chain Rule and HMMs



- From the chain rule, *every* joint distribution over $X_1, E_1, \ldots, X_T, E_T$ can be written as:

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

- *Assuming* that for all *t*:
  - State independent of all past states and all past evidence given the previous state, i.e.:

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

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

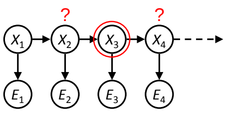
$$E_t \perp\!\!\!\perp X_1, E_1, \ldots, X_{t-2}, E_{t-2}, X_{t-1}, E_{t-1} \mid X_t$$

gives us the expression posited on the earlier slide:

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

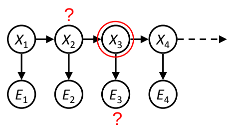
Conditional Independence

- HMMs have two important independence properties:
  - Markov hidden process: future depends on past via the present



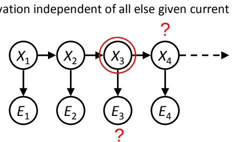
Conditional Independence

- 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



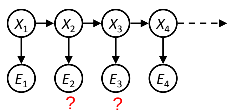
Conditional Independence

- 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

- 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
  - evidence $E_{1:n} = e_{1:n}$
- 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

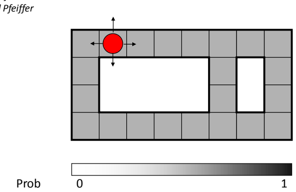
$$x^*_{1:n} = \mathrm{argmax}_{x_{1:n}}\, P(x_{1:n}|e_{1:n})$$

Filtering / Monitoring

- Filtering, or monitoring, is the task of tracking the distribution $B_t(X) = P_t(X_t \mid e_1, \ldots, e_t)$ (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 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

Example from Michael Pfeiffer

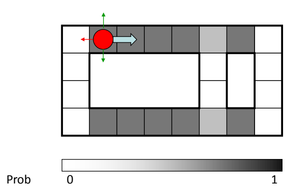


t=0

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.

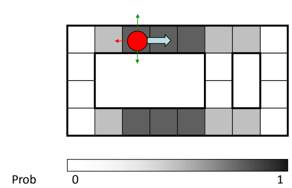
Example: Robot Localization



t=1

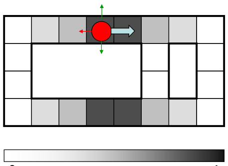
Lighter grey: was possible to get the reading, but less likely b/c required 1 mistake

Example: Robot Localization



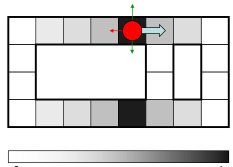
t=2

Example: Robot Localization



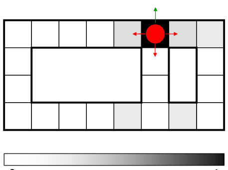
t=3

Example: Robot Localization



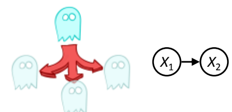
t=4

Example: Robot Localization



t=5

Inference: Base Cases



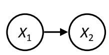
$$P(X_1|e_1)$$

$$P(x_1|e_1) = P(x_1, e_1)/P(e_1)$$
$$\propto_{X_1} P(x_1, e_1)$$
$$= P(x_1)P(e_1|x_1)$$

$$P(X_2)$$

$$P(x_2) = \sum_{x_1} P(x_1, x_2)$$
$$= \sum_{x_1} P(x_1)P(x_2|x_1)$$

Passage of Time



- Assume we have current belief P(X | evidence to date)

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

- Then, after one time step passes:

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

- Or compactly:

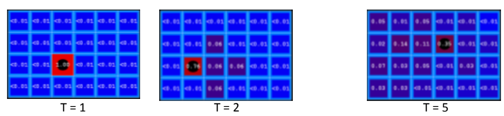
$$B'(X_{t+1}) = \sum_{x_t} P(X'|x_t)B(x_t)$$

- 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

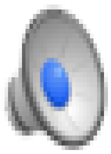
Example: Passage of Time

- As time passes, uncertainty "accumulates"

(Transition model: ghosts usually go clockwise)



Video of Passage of Time (Transition Model)



Observation

- Assume we have current belief P(X | previous evidence):

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

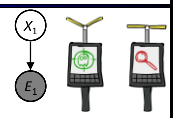
- Then, after evidence comes in:

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

- Or, compactly:

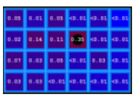
$$B(X_{t+1}) \propto_{X_{t+1}} P(e_{t+1}|X_{t+1})B'(X_{t+1})$$

- Basic idea: beliefs "reweighted" by likelihood of evidence
- Unlike passage of time, we have to renormalize

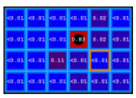


Example: Observation

- As we get observations, beliefs get reweighted, uncertainty "decreases"



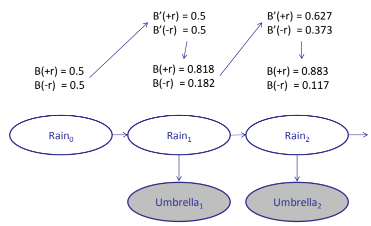
Before observation



After observation

$$B(X) \propto P(e|X)B'(X)$$

Example: Weather HMM



R_t	R_{t+1}	$P(R_{t+1}\|R_t)$
+r	+r	0.7
+r	-r	0.3
-r	+r	0.3
-r	-r	0.7

R_t	U_t	$P(U_t\|R_t)$
+r	+u	0.9
+r	-u	0.1
-r	+u	0.2
-r	-u	0.8

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

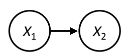
$$P(x_t|e_{1:t}) \propto_X P(x_t, e_{1:t})$$
$$= \sum_{x_{t-1}} P(x_{t-1}, x_t, e_{1:t})$$
$$= \sum_{x_{t-1}} P(x_{t-1}, e_{1:t-1})P(x_t|x_{t-1})P(e_t|x_t)$$
$$= P(e_t|x_t) \sum_{x_{t-1}} P(x_t|x_{t-1})P(x_{t-1}, e_{1:t-1})$$

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

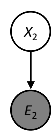
- Every time step, we start with current P(X | evidence)
- We update for time:

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



- We update for evidence:

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

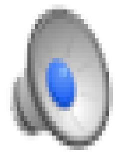


- The forward algorithm does both at once (and doesn't normalize)
- Potential issue: space is |X| and time is $|X|^2$ per time step

Pacman – Sonar (P4)

[Demo: Pacman – Sonar – No Beliefs(L14D1)]

Video of Demo Pacman – Sonar (with beliefs)



HMM Computations (Reminder)

- Given
  - parameters
  - evidence $E_{1:n} = e_{1:n}$
- 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
    $x^*_{1:n} = \text{argmax}_{x_{1:n}} P(x_{1:n}|e_{1:n})$

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+1*

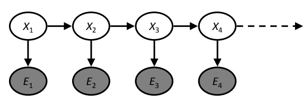
$$\begin{aligned} P(X_t|e_{1:n}) &= \alpha\, P(X_t|e_{1:t}) P(e_{t+1:n}|X_t, e_{1:t}) \\ &= \alpha\, P(X_t|e_{1:t}) P(e_{t+1:n}|X_t) \\ &= \alpha\, \mathbf{f}_{1:t} \times \mathbf{b}_{t+1:n} \end{aligned}$$

45

Most Likely Explanation

HMMs: MLE Queries

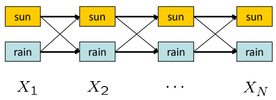
- HMMs defined by
  - States X
  - Observations E
  - Initial distribution: $P(X_1)$
  - Transitions: $P(X|X_{-1})$
  - Emissions: $P(E|X)$



- New query: most likely explanation: $\arg\max_{x_{1:t}} P(x_{1:t}|e_{1:t})$
- New method: the Viterbi algorithm

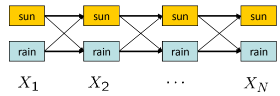
State Trellis

- 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)

$$f_t[x_t] = P(x_t, e_{1:t})$$

$$= P(e_t|x_t) \sum_{x_{t-1}} P(x_t|x_{t-1}) f_{t-1}[x_{t-1}]$$

Viterbi Algorithm (Max)

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

$$= P(e_t|x_t) \max_{x_{t-1}} P(x_t|x_{t-1}) m_{t-1}[x_{t-1}]$$

Most Probably Explanation (Sequence)

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