EECS 445 Discussion 12: Bayesian Networks 2 and Hidden Markov Models

1 More on Bayesian Networks

1.1 Discovering Dependence through D-separation

Suppose we are interested in whether a pair of nodes X_1, X_2 are conditionally independent given a third node X_3, or marginally independent given no nodes. We perform the following procedure on the graph (this is called *moralizing*):

1. Keep only the "ancestral" graph of the variables of interest. In other words, keep the variables of interest themselves in the graph, and keep all nodes that have a directed path to those nodes (ancestors). Remove all other nodes from the graph.
2. Moralize the parents: add undirected edges between any two nodes in the ancestral graph that share a child. If a child has more than two parents, add edges between all of the parents.
3. Change the remaining edges in the graph to be undirected.
4. We are now ready to read off independence properties from the graph:
   - If there is no path from X_1 to X_2, then they are marginally independent (assuming those two were the only nodes of interest when doing the procedure). We write: $X_1 \perp\!\!\!\perp X_2$
   - If there are no paths from X_1 to X_2 or all such paths go through X_3, then X_1 and X_2 are conditionally independent given X_3. We write: $X_1 \perp\!\!\!\perp X_2 | X_3$.
     Note: If we were to check if $X_1 \perp\!\!\!\perp X_2 | \{X_3, X_4\}$, then we would need to verify that all paths from X_1 to X_2 go through any of the variables in $\{X_3, X_4\}$.

Lets use an example to illustrate the concept: Consider the following graph below:

Discussion Question 1. *Are E and F marginally independent? Are they conditionally independent given A? Given B? Given C? Given C AND D?*

Note that for these sets of questions, the final graph we arrive at doing the above steps will be the same because A, B, C, D are all ancestors of E and F! Thus we can use the below moralized graph to answer all of these.

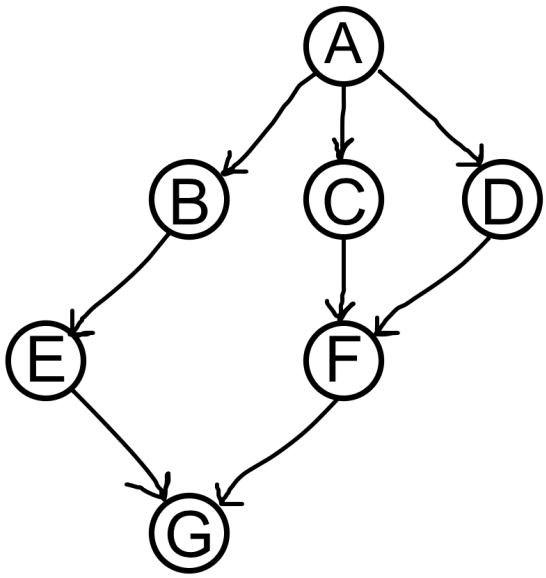
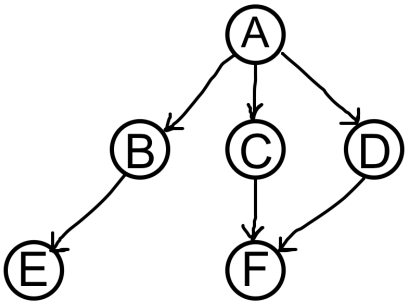
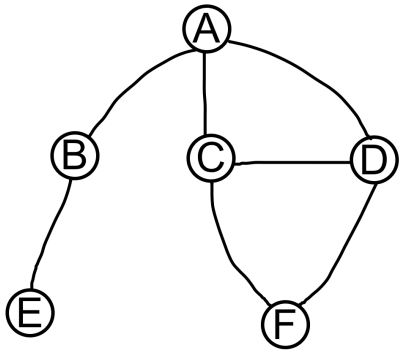


Figure 1: A Bayesian Network



(a) Ancestral Graph



(b) Moralized Graph

Figure 2: Deriving Conditional Independence Properties

Solution: E and F are not independent, because there are paths between them. E and F are conditionally independent given B and given A because all paths in the final graph from E to F go through both B and A.
E and F are conditionally independent given C AND D, because all paths between E and F go through C or through D.
However, E and F are conditionally dependent given C or given D, because not all paths between E and F go through C, nor do they all go through D.

Discussion Question 2. *Let's do a computational example. Recall the radio report/alarm graph from*

lecture, together with its probability distribution tables.

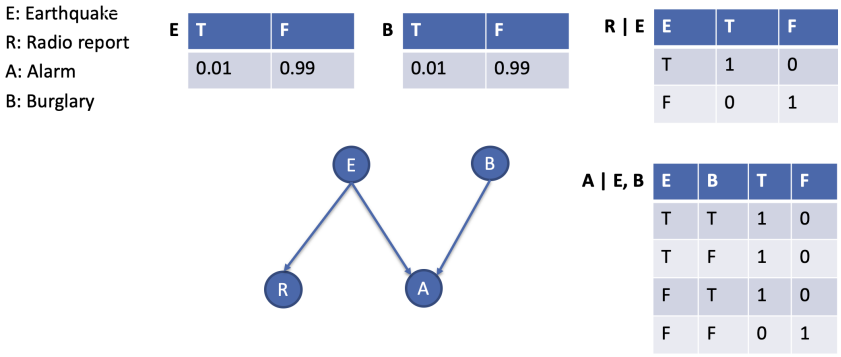


Figure 3: Generative probability model over 4 variables

Compute the probability that a burglary happened given that the alarm went off? Does the result match your intuition? Explain.

Solution:

$$
\begin{aligned}
P(B=T|A=T) &= \frac{P(B=T,A=T)}{P(A=T)} = \\
&= \frac{P(B=T)P(A=T|B=T)}{P(A=T)} = \\
&= \frac{P(B=T)\sum_{e\in\{T,F\}} P(A=T,E=e|B=T)}{\sum_{b\in\{T,F\}} P(B=b)\sum_{e\in\{T,F\}} P(A=T,E=e|B=b)} = \\
&= \frac{P(B=T)\sum_{e\in\{T,F\}} P(E=e)P(A=T|E=e,B=T)}{\sum_{b\in\{T,F\}} P(B=b)\sum_{e\in\{T,F\}} P(E=e)P(A=T|E=e,B=b)} = \\
&= \frac{0.01(0.01\times 1+0.99\times 1)}{0.01(0.01\times 1+0.99\times 1)+0.99(0.01\times 1+0.99\times 0)} = \\
&= \frac{0.01}{0.01+0.0099} \approx 0.5025
\end{aligned}
$$

Due to the symmetry of this example, we expect that $P(B=T|A=T) = P(E=T|A=T)$. However, we notice that $P(B=T|A=T) > 0.5$. This is due to the fact that B and E are not mutually exclusive, and there is a slight chance that both events occur simultaneously.

Discussion Question 3. *Suppose we learned that the radio report also occurred. How does the probability that a burglary occurred change? In other words, what is* $P(B=T|R=T,A=T)$*?*

Solution:

$$
\begin{aligned}
P(B=T|R=T,A=T) &= \frac{P(B=T,R=T,A=T)}{P(R=T,A=T)} = \\
&= \frac{\sum_e P(B=T,R=T,A=T,E=e)}{\sum_e \sum_b P(A=T|E=e,B=b)P(R=T|E=e)P(E=e)P(B=b)} = \\
&= \frac{\sum_e P(A=T|E=e,B=T)P(R=T|E=e)P(E=e)P(B=T)}{\sum_e P(R=T|E=e)P(E=e)\sum_b P(A=T|E=e,B=b)P(B=b)} = \\
&= \frac{1\times 1\times 0.01\times 0.01 + 1\times 0\times 0.99\times 0.01}{0.01\times 1(1\times 0.01+1\times 0.99)+0.99\times 0(1\times 0.01+0\times 0.99)} = \\
&= \frac{0.0001}{0.01} = \\
&= 0.01 = \\
&= P(B=T)
\end{aligned}
$$

Notice that knowing that $R=T$ tells us that $E=T$ because

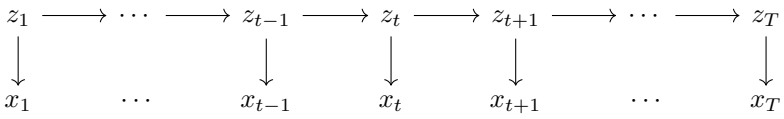
$$P(E=T|R=T,A=T) = P(E=T|R=T) = \frac{P(R=T|E=T)P(E=T)}{P(R=T)} = \frac{0.01}{0.01} = 1$$

Knowing $E=T$ completely determines A, effectively removing all additional information that we had gained about B before.

2 Hidden Markov Models

Hidden Markov Models are a particular type of graphical model in which we assume that the data we observe come from a sequence of hidden states. Each observation is associated with a state; each state is independent of all other states and observations given the previous state. Each observation is independent of all other observations and states given the associated state.

An HMM can be represented by the following model:



Underlying hidden "states" are denoted in the top row, while observations are emitted from these hidden states. We typically know the observations in the bottom row and want to estimate the corresponding states in the top row.

2.1 The Setup

We formally define an HMM as a tuple $\langle H, O, \theta \rangle$ where:

- H is the set of possible hidden states, $|H| = M$
- O is the set of possible observations, $|O| = N$
- $\theta = [A, B, \pi]$

- A is an $M \times M$ matrix where $A(h_i, h_j) = P(z_{t+1} = h_j \mid z_t = h_i)$ is the probability of transitioning from state h_i to h_j
- B is an $M \times N$ matrix where $B(h_i, o_\ell) = P(x_t = o_\ell \mid z_t = h_i)$ is the probability of emitting symbol o_ℓ from state h_i.
- π is an $M \times 1$ vector where $\pi(h_i) = P(z_1 = h_i)$ is the probability of starting in state h_i.

Given this, the probability of a sequence of observations and states in an HMM is:

$$P(x_1, \ldots, x_T, z_1, \ldots, z_T; \theta) = \pi(z_1) \prod_{t=1}^{T-1} A(z_t, z_{t+1}) \prod_{t=1}^{T} B(z_t, x_t)$$

2.2 Example: Setting up HMMs

Suppose you send a robot to Mars. Unfortunately, it gets stuck in a canyon while landing and most of its sensors break. You know the canyon has 3 areas. Areas 1 and 3 are sunny and hot, while Area 2 is cold. You decide to plan a rescue mission for the robot from Area 3, knowing the following things about the robot:

1. Every hour, it tries to move forward by one area (i.e. from Area 1 to Area 2, or Area 2 to Area 3). It succeeds with probability 0.75 and fails with probability 0.25. If it fails, it stays where it is. If it is in Area 3, it always stays there (and waits to be rescued).
2. The temperature sensor still works. Every hour, we get a binary reading telling us whether the robot's current environment is hot or cold.
3. We have no idea where the robot initially got stuck.

Discussion Question 4. *Construct an HMM for this problem: define a transition matrix A, an observation matrix B, and an initial state distribution* π_0

Solution: We'll start with the transition matrix. Remember that each row corresponds to the current state, and each column corresponds to the next state. We'll use 3 states, each corresponding to an area.

1. If the robot is in Area 1, it stays where it is with probability 0.25, moves to Area 2 with probability 0.75, and can't move to Area 3.
2. Similarly, if the robot is in Area 2, it stays where it is with probability 0.25, can't move back to Area 1, and moves to Area 3 with probability 0.75.
3. If the robot is in Area 3, it always stays in Area 3.

Each item above gives us one row of A. Putting it all together, we obtain:

$$A = \begin{array}{c} \\ \mathbf{1} \\ \mathbf{2} \\ \mathbf{3} \end{array} \begin{array}{c} \begin{array}{ccc} \mathbf{1} & \mathbf{2} & \mathbf{3} \end{array} \\ \begin{bmatrix} 0.25 & 0.75 & 0 \\ 0 & 0.25 & 0.75 \\ 0 & 0 & 1 \end{bmatrix} \end{array}$$

Next, let's look at the observation matrix. There are two possible observations, hot and cold. Areas 1 and 3 always produce "hot" readings while Area 2 always produces a "cold" reading:

$$B = \begin{array}{c} \\ \mathbf{1} \\ \mathbf{2} \\ \mathbf{3} \end{array} \begin{array}{c} \begin{array}{cc} \mathbf{hot} & \mathbf{cold} \end{array} \\ \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \end{array}$$

Since we have no idea where the robot starts, our initial state distribution will be uniform:

$$\pi_0 = \begin{matrix}\mathbf{1}\\ \mathbf{2}\\ \mathbf{3}\end{matrix}\begin{bmatrix}\frac{1}{3}\\ \frac{1}{3}\\ \frac{1}{3}\end{bmatrix}$$

3 The Viterbi Algorithm

Suppose that we knew the model parameters θ for and HMM and observed an output sequence $x_1, \ldots, x_T$. We want to know which sequence of hidden states $z_1, \ldots, z_T$ was most likely to generate this output sequence.

$$\arg \max_{z_1 \ldots, z_T} p(x_1, \ldots, x_T, z_1, \ldots, z_T; \theta)$$

We could try brute forcing through all possible sequences, but there are M^T possible such sequences! That approach is computationally intractable. Luckily, there is an efficient $O(TM^2)$ dynamic programming algorithm called the **Viterbi Algorithm** that solves this problem for us.

Define:

$$r(z_1, \ldots, z_k) = \prod_{t=1}^{k} A(z_{t-1}, z_t) \prod_{t=1}^{k} B(z_t, x_t) \text{ where } A(z_0, z_1) = \pi(z_1)$$

$$S(k, v) = \text{ the set of all sequences of length } k \text{ that end in } z_k = v$$

$$\Psi(k, v) = \max_{S(k,v)} r(z_1, \ldots, z_k), \text{ most probable sequence of length } k \text{ that ends with } z_k = v$$

$\Psi(k, v)$ has the following recursive property:

$$\Psi(k, v) = \max_{u \in H} \Big\{ \Psi(k-1, u) A(u, v) B(v, x_k) \Big\}$$

This is saying that the most likely sequence of length k that ends with z_k is simply the sequence which maximizes the probability of the best sequence of length $k-1$ ending in some state u times the probability of transitioning from u to v times the probability of emitting symbol x_k in state v.
We define our basecase as:

$$\Psi(1, v) = \pi(v) B(v, x_1)$$

We can then calculate the values for $\Psi(k, v)$ by iterating through k and v and storing previous values in table. Note that there are MT entries in this table for $k = 1, \ldots, T$ and $v = 1, \ldots, T$. Calculating each entry takes a maximum over every possible state $u \in H$, which leads to the final complexity of $O(TM^2)$.

3.1 HMM/Viterbi: Example 1

Suppose we have the following HMM:

- $H = \{H = healthy, S = sick\}$
- $O = \{W = wheezing, c = coughing, si = smiling\}$
- $A =$

	H	S
H	0.5	0.5
S	0.2	0.8

Note: in our definition $A(H, S) = 0.8$ is the probability of transitioning from H to S.

- $B =$

	W	C	si
H	0.1	0.2	0.7
S	0.4	0.5	0.1

- $\pi =$

H	0.2
S	0.8

Suppose we observe the sequence w, si.

Discussion Question 5. *What is the most likely sequence of hidden states? You can solve this using brute force*

Solution: Let's consider all possible states that it could take on. The possible sequences of hidden states are:
HH
HS
SH
SS

The probability of sequence HH is:
$\mathrm{P}(HH) = \pi(H) * B(H, w) * A(H, H) * B(H, si)$
$\mathrm{P}(HH) = 0.2 * 0.1 * 0.5 * 0.7 = 0.007$
The probability of sequence HS is:
$\mathrm{P}(HS) = \pi(H) * B(H, w) * A(H, S) * B(S, si)$
$\mathrm{P}(HS) = 0.2 * 0.1 * 0.5 * 0.1 = 0.001$
The probability of sequence SH is:
$\mathrm{P}(SH) = \pi(S) * B(S, w) * A(H, H) * B(H, si)$
$\mathrm{P}(SH) = 0.8 * 0.4 * 0.2 * 0.7 = 0.0448$
The probability of sequence SS is:
$\mathrm{P}(SS) = \pi(S) * B(S, w) * A(S, S) * B(S, si)$
$\mathrm{P}(SS) = 0.8 * 0.4 * 0.8 * 0.1 = 0.0256$
Thus sequence SH is the most likely sequence of hidden states.

3.2 Viterbi Algorithm: Example 2

Suppose we have the following HMM:

- $H = \{X, Y, Z\}$
- $O = \{0, 1\}$
- $A =$

	X	Y
X	0.3	0.7
Y	0.2	0.8

  Note: in our definition $A(X, Y) = 0.7$ is the probability of transitioning from X to Y.

- $B =$

	0	1
X	0.3	0.7
Y	0.5	0.5

- $\pi =$

X	0.6
Y	0.4

Suppose we observe the sequence 001. The Viterbi algorithm has been run and produced the following table:

	$t=1$	$t=2$	$t=3$
X	0.18	0.0162	0.0112
Y	0.2	0.08	0.032

Unfortunately, we have not recorded which previous states maximized the next states in the table! We will need to manually backtrack using the formulas:

$$\hat{z}_T = \arg\max_v \big\{\Psi(T, v)\big\}$$

$$\hat{z}_i = \arg\max_v \big\{\Psi(i, v)A(v, \hat{z}_{i+1})\big\}, \forall i < T$$

Discussion Question 6. *Given the above table, what is the most likely sequence of hidden states?*

Solution: Answer: We can use the above definitions to find that the most likely sequence of states is YYY.

$$\hat{z}_3 = \arg\max\big\{\Psi(3, X), \Psi(3, Y)\big\} = \arg\max\big\{0.0112, 0.032\big\} = Y$$

$$\hat{z}_2 = \arg\max\big\{\Psi(2, X)A(X, Y), \Psi(2, Y)A(Y, Y)\big\}$$

$$\hat{z}_2 = \arg\max\big\{0.0162 * 0.7, 0.08 * 0.8\big\} = \arg\max\big\{0.01134, 0.064\big\} = Y$$

$$\hat{z}_1 = \arg\max\big\{\Psi(1, X)A(X, Y), \Psi(1, Y)A(Y, Y)\big\}$$

$$\hat{z}_1 = \arg\max\big\{0.18 * 0.7, 0.2 * 0.8\big\} = \arg\max\big\{0.126, 0.16\big\} = Y$$

Note that we can backtrack in the way we did above, but it requires $O(TM)$ steps to backtrack: $O(M)$ steps for each $\arg\max$, we do an $\arg\max$ $O(T)$ times. Instead, we should keep a back pointer which tells us in the table which previous state maximized the current state! That prevents us from having to do any additional calculations.