CSE 250A FINAL EXAM
FALL QUARTER 2021

OUT: `Sun Dec 05 @ noon PST (Canvas)`
DUE: `Mon Dec 06 @ noon PST (Gradescope)`

Question	Points
Academic integrity statement	*
1. d-separation	5
2. Inference in a small network	10
3. Inference in an extended network	5
4. Inference in a bipartite network	14
5. Trimmed logistic regression	17

Question	Points
6. EM algorithm	10
7. Inference in HMMs	7
8. (Reverse) Viterbi algorithm	13
9. Policy improvement	8
10.Temporal difference learning	11
Total:	**100**

Do not wait until the last possible moment to upload your completed exam. There will be a ten-minute grace period to allow for last-minute glitches. After this grace period, one point will be deducted for each minute of lateness. Gradescope will stop accepting exams at 1:00 pm.

The exam is expected to take one full afternoon, but you may work as long as needed until the deadline.
Be sure to sign and submit the statement of academic integrity with your completed exam.
Exams without signed statements will not be graded.

It is recommended, but not required, to print the exam and submit your answers in the space provided. You do not need to typeset your solutions. Your writing should be legible, and your final answers should be clearly indicated.

There are no programming questions on the exam. You are allowed to check your arithmetic with programs in Python, Matlab, R, etc. But this should not be necessary, and you may need to show intermediate steps for full credit.

The later parts of problems often do not depend on the results of earlier ones; therefore it is a good strategy to attempt all parts of every problem. An asterisk may indicate a less familiar calculation. If you do not see the solution quickly, you may wish to return to these parts later.

If something is unclear, state the assumptions that seem most natural to you and proceed under those assumptions. Out of fairness, we may decline to answers questions that are posted on Piazza or sent to us via email. Do not post any public questions on Piazza during the course of the exam.

Academic Integrity Statement

This take-home, open-book exam represents work that I have completed on my own. In particular, during the twenty-four hour period of the exam, I hereby affirm[1] the following:

(i) I have not communicated with other students in the class about these problems.

(ii) I have not consulted other knowledgeable researchers or acquaintances.

(iii) I have not contracted for help on any parts of the exam over the Internet.

I understand that any of the above actions, if proven, will result in a failing grade on the exam. In addition, I understand the following:

(i) I am obligated to report any violations of academic integrity by others that come to my attention during the exam.

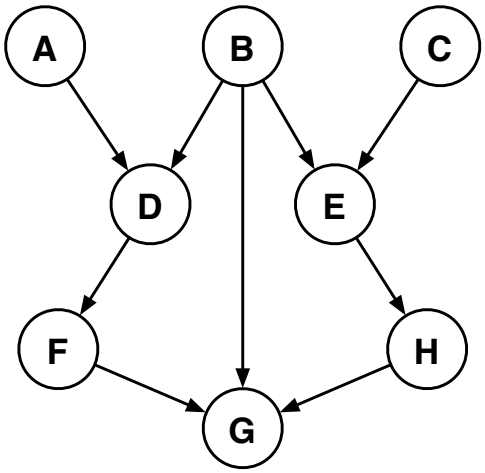
(ii) All other students in the course (past and present) are under the same obligation.

Signature **Date**

[1] **If you do not have a printer, you may simply copy, sign, and date the following statement with your solutions:** *I affirm the statement of academic integrity on page 2 of the exam.*

1. d-separation (5 pts)

For the belief network shown below, indicate whether the following statements of marginal or conditional independence are **true (T)** or **false (F)**. No further explanation is required.



_____ $P(A) = P(A|E)$

_____ $P(A|C,G) = P(A|G)$

_____ $P(A|D,F,G) = P(A|D,G)$

_____ $P(A,B,C|D,E) = P(A|B,D)\,P(B|C,D,E)\,P(C|D,E)$

_____ $P(D,G|F) = P(D|F)\,P(G|F)$

_____ $P(D,E|B) = P(D|B)\,P(E|B)$

_____ $P(F|C) = P(F)$

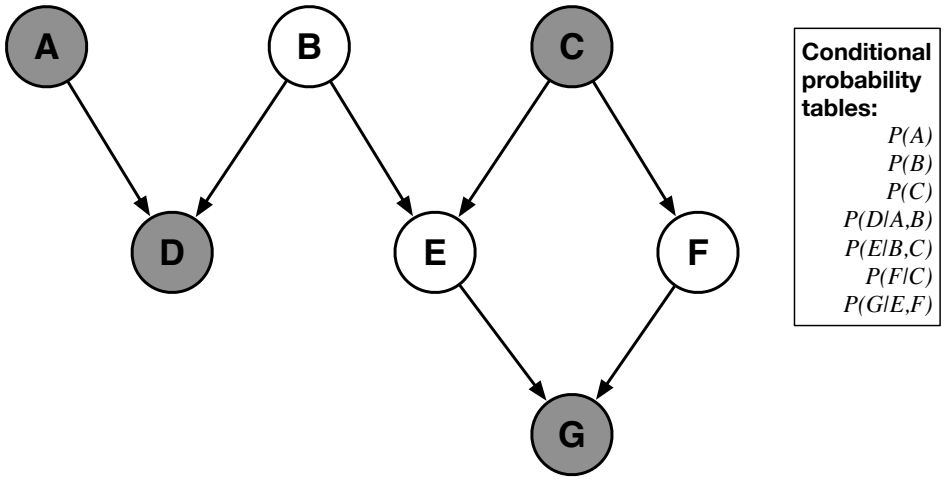
_____ $P(F|C,E) = P(F|E)$

_____ $P(F,H|B) = P(F|B)\,P(H|B)$

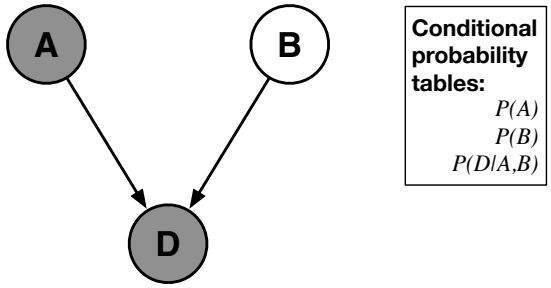
_____ $P(F,H|B,G) = P(F|B,G)\,P(H|B,G)$

2. Inference in a small network

Consider the belief network shown below, whose shaded nodes are observed. In this problem you will be guided through an efficient computation of the posterior probability $P(F|A, C, D, G)$. You are expected to perform these computations *efficiently*—that is, by exploiting the structure of the DAG and not marginalizing over more variables than necessary.

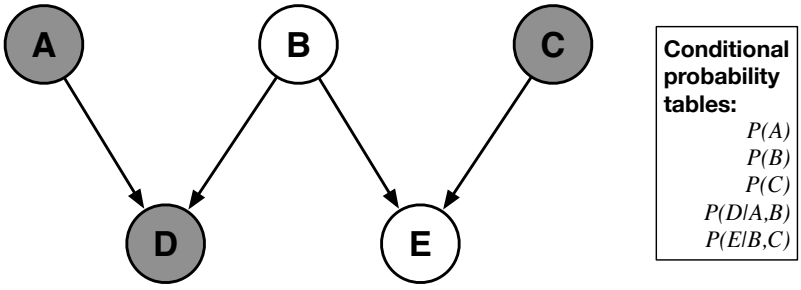


(Continued on next page...)



(a) **Node B** (3 pts)

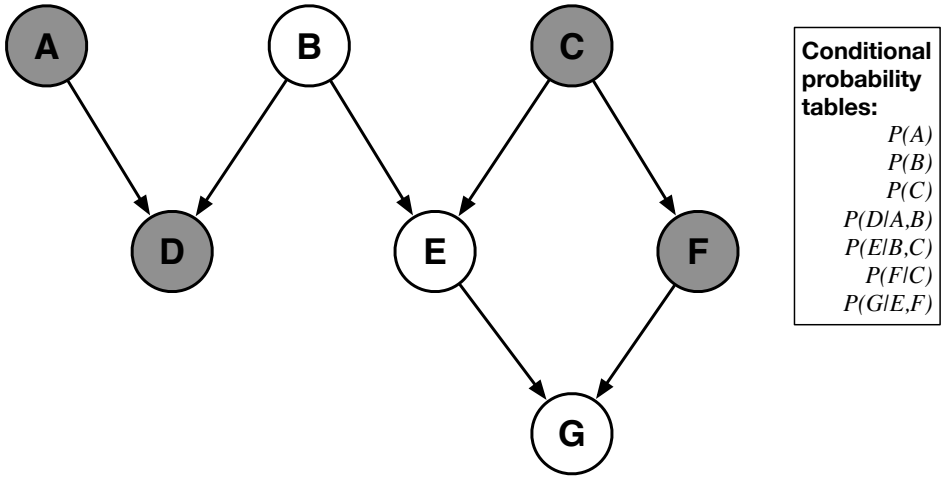
Consider just the part of the belief network shown above. Show how to compute the posterior probability $P(B|A, D)$ in terms of the conditional probability tables (CPTs) for these nodes. *Briefly justify each step of your solution.*



(b) **Node E** (2 pts)

Consider just the part of the belief network shown above. Show how to compute the posterior probability $P(E|A, C, D)$ in terms of your answer from part (a) and the CPTs of the belief network. *Briefly justify each step of your solution.*

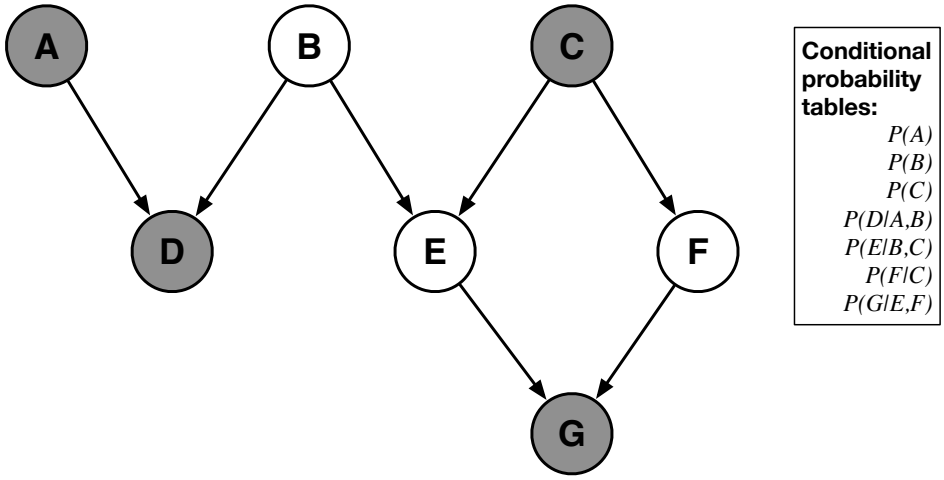
Note: for this problem you may assume that $P(B|A, D)$ from part (a) is given. Thus you can answer this question even if you skipped or missed part (a).



(c) **Node G** (2 pts)

Consider just the part of the belief network shown above. Show how to compute the conditional probability $P(G|A, C, D, F)$ in terms of your answer from part (b) and the CPTs of the belief network. *Briefly justify each step of your solution.*

Note: for this problem you may assume that $P(E|A, C, D)$ from part (b) is given. Thus you can answer this question even if you skipped or missed part (b).



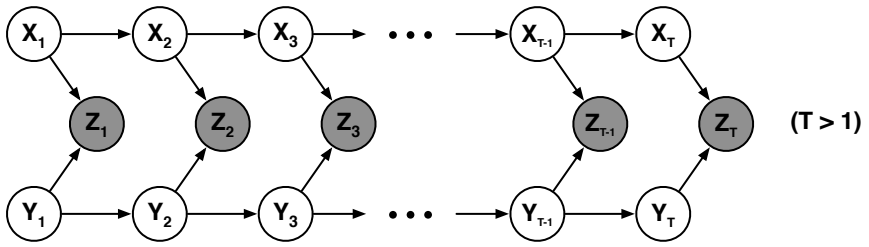
(d) **Node F** (3 pts)

Show how to compute the posterior probability $P(F|A, C, D, G)$ in terms of your answer from parts (a,b,c) and the CPTs of the belief network. *Briefly justify each step of your solution.*

Note: for this problem you may assume that $P(G|A, C, D, F)$ from part (c) is given. Thus you can answer this question even if you skipped or missed part (c).

3. Inference in an extended network

Consider the belief network of discrete random variables shown below, where $X_t \in \{1, 2, ..., n\}$ and $Y_t \in \{1, 2, ..., n\}$ form parallel chains, and $Z_t \in \{1, 2, ..., m\}$ is the child of X_t and Y_t at time t.



(a) **Polytree?** (1 pt)

Is this belief network a polytree? Justify your answer by stating the definition of a polytree, then explaining how the network either satisfies or violates your definition.

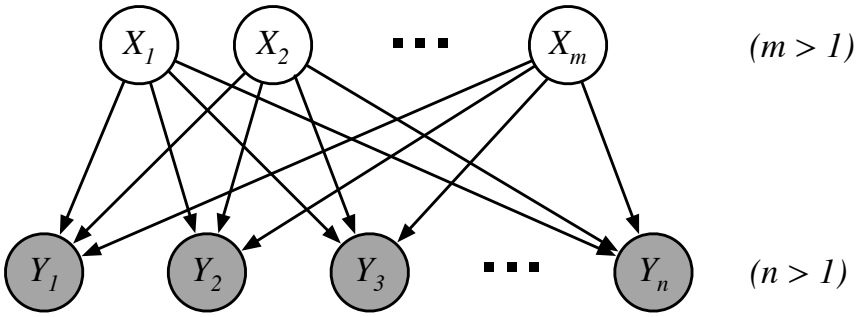
(b) **Inference** (4 pts)

Consider how to efficiently compute the marginal probability $P(Z_1, Z_2, \ldots, Z_T)$.
What is the computational complexity of this inference?

Answer this question by *clustering* the nodes in this DAG (wisely) to obtain a more familiar belief network. Then, based on this clustering, give a precise answer, noting how the complexity of inference depends on the cardinalities n and m, as well as the sequence length T. (For instance, is the dependence constant, linear, polynomial, or exponential?)

Note: you are **not** asked nor expected to derive an algorithm for this inference, only to deduce its computational complexity.

4. Inference in a bipartite network



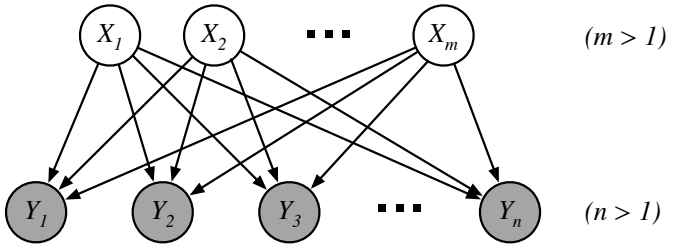
(a) **Polytree?** (1 pt)

Consider the above belief network of binary random variables. Is the network a polytree? Explain (in one sentence) why or why not.

(b) **Marginal probability** (3 pts)

Show how to compute the marginal probability $P(y_1, y_2, \ldots, y_n)$ in terms of the conditional probability tables (CPTs) of the belief network—that is, in terms of the prior probabilities $P(x_i)$ and the conditional probabilities $P(y_j|x_1, x_2, \ldots, x_m)$.

4. Inference in a bipartite network (con't)



(c) **Scaling** (2 pts)

How does your computation of the marginal probability $P(y_1, y_2, \ldots, y_n)$ from part (b) scale in terms of the numbers of root notes (m) and second-layer nodes (n)? Be precise in your answer (i.e., linearly, polynomially, exponentially), and briefly justify your reasoning.

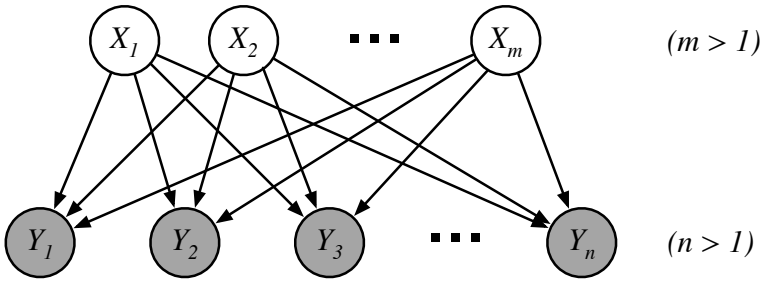
(d) **Normalization** (2 pts)

As shorthand, let $\mu_i = P(X_i = 1)$ denote the prior probability of each root node to be equal to one. The distribution over root nodes in the network can be written as:

$$P(x_1, x_2, \ldots, x_m) \;=\; \prod_{i=1}^{m} \mu_i^{x_i} (1-\mu_i)^{1-x_i}.$$

Prove that this distribution is normalized—namely, that the sum $\sum_{x_1,x_2,\ldots,x_m} P(x_1, x_2, \ldots, x_m)$ is equal to unity. This calculation, though seemingly uninteresting, is designed to prepare you for part (f), where you will prove a nontrivial result about noisy-OR networks.

4. Inference in a bipartite network (con't)



(e) **Noisy-OR** (2 pts)

Suppose that noisy-OR conditional probability tables are used at the nodes in the second layer of the network. In particular, let

$$P(Y_j=1|x_1,x_2,\ldots,x_m) \;=\; 1-\prod_{i=1}^{m}(1-\rho_{ij})^{x_i},$$

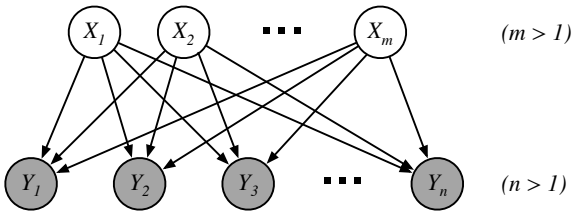
where ρ_{ij} is the noisy-OR parameter attached to the edge in the belief network that connects node X_i to node Y_j. Consider the marginal probability

$$P(Y_1=0,Y_2=0,\ldots,Y_n=0)$$

of all-zero observations. By specializing your result[2] from part (b) to noisy-OR networks, express this marginal probability in terms of the parameters $\mu_i = P(X_i=1)$ and ρ_{ij}.

[2] *Hint:* your answer will involve a sum over values of the network's unobserved nodes. You do **not** need to perform this sum in this part of the question.

4. Inference in a bipartite network (con't)



(f*) **Efficient inference** (4 pts)

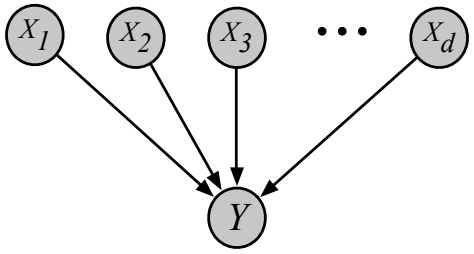
Polynomial-time inference is generally not possible in loopy belief networks. In this noisy-OR network, however, it is possible to compute the marginal probability

$$P(Y_1 = 0, Y_2 = 0, \ldots, Y_n = 0)$$

of *all-zero* observations in time linear in the number of edges of the network—that is, in time $O(mn)$. To do so, fully simplify your expression from part (e), now explicitly performing the sum over the network's unobserved nodes.

Hint: consider how you proved the result in part (d), which also involved a seemingly exponential sum over the same nodes.

5. Trimmed logistic regression



Consider the belief network shown above, with real-valued variables $X_i \in \Re$ and binary variables $Y \in \{0, 1\}$. Suppose that the conditional probability table (CPT) for node Y has the form:

$$P\,(Y=1|\vec{x}) \;=\; \gamma\sigma(\vec{w}\cdot\vec{x})$$

where $\sigma(z) = [1 + e^{-z}]^{-1}$ is the usual sigmoid function, $\vec{w} \in \Re^d$ is a real-valued weight vector, and γ *is an additional model parameter between* 0 *and* 1. Note that $P\,(Y=1|\vec{x})$ in this model is *trimmed* to a maximum value of γ.

(a) **Log likelihood** (2 pts)

Consider a fully observed data set of i.i.d. examples $\{\vec{x}_t, y_t\}_{t=1}^T$ where for each example $\vec{x}_t \in \Re^d$ and $y_t \in \{0, 1\}$. Compute the log (conditional) likelihood,

$$\mathcal{L}(\vec{w}, \gamma) \;=\; \sum_{t=1}^{T} \log P(y_t|\vec{x}_t)\,,$$

in terms of the parameter γ and the weight vector $\vec{w}$. You will want to simplify your expression as much as possible for the next part of this question.

(b) **Gradient** (4 pts)

As shorthand in this problem, let $p_t = P(Y=1|\vec{x}_t)$. The gradient $\frac{\partial \mathcal{L}}{\partial \vec{w}}$ of the log-likelihood in part (a) is given by one of the following expressions:

$$\text{(i)} \quad \frac{\partial \mathcal{L}}{\partial \vec{w}} = \sum_{t=1}^{T} (y_t - p_t)\, \vec{x}_t,$$

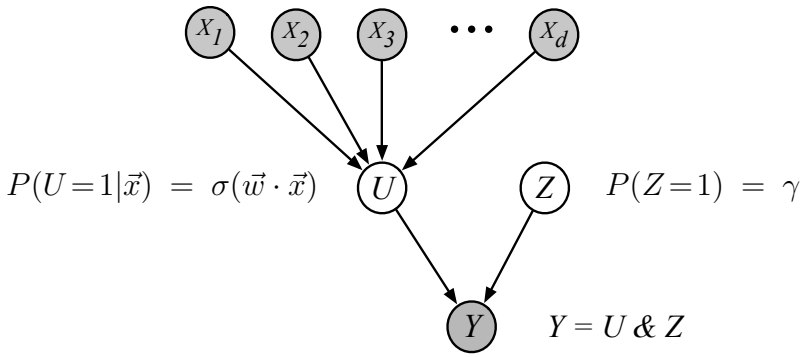
$$\text{(ii)} \quad \frac{\partial \mathcal{L}}{\partial \vec{w}} = \sum_{t=1}^{T} \left(\frac{y_t - p_t}{1 - p_t} \right) \vec{x}_t,$$

$$\text{(iii)} \quad \frac{\partial \mathcal{L}}{\partial \vec{w}} = \sum_{t=1}^{T} \sigma(-\vec{w} \cdot \vec{x}_t) \left(\frac{y_t - p_t}{1 - p_t} \right) \vec{x}_t.$$

Compute this gradient, and show that your final answer matches one of these expressions. *You must show your work for credit.*

(c) **Latent variable model** (4 pts)

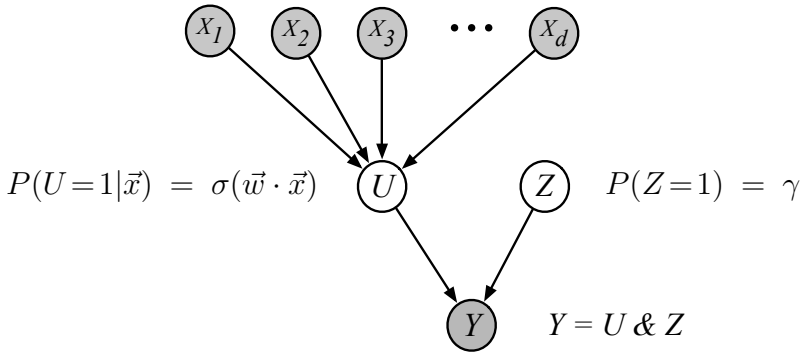
One can derive an EM update for the parameter γ by reformulating this belief network as a latent variable model. The rest of the problem guides you through this process.



Consider the belief network shown above, with binary hidden variables U and Z. Suppose also that node Y in this model is *determined* by the logical-AND of its parents:

$$P(Y = 1|U, Z) = \begin{cases} 1 & \text{if } U = Z = 1, \\ 0 & \text{otherwise} \end{cases}$$

Compute $P(Y = 1|\vec{x})$ in this belief network, and show that it takes the same exact form as the model in part (a). *Justify your steps briefly to receive full credit.*



(d) **Posterior for positive examples** (1 pt)

What is the posterior probability $P(Z=1|\vec{x}, Y=1)$ for positively labeled examples in this hidden variable model? *Hint:* no calculation required! (But do justify your answer.)

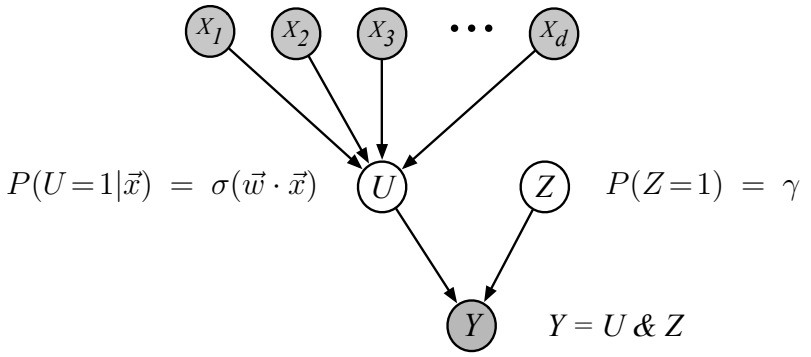
(e*) **Posterior for negative examples** (5 pts)

Show how to compute the posterior probability for negatively labeled examples,

$$P(Z=1|\vec{x}, Y=0),$$

in terms of the input vector $\vec{x}$, the weight vector $\vec{w}$, and the parameter γ. *Justify your steps briefly to receive full credit.*

(f) **EM update** (1 pt)



The EM algorithm maximizes the likelihood in part (a) by iteratively computing the posteriors $P(Z=1|\vec{x}_t, y_t)$ and using them to update the parameter γ. Assuming these probabilities have been computed for each example $(\vec{x}_t, y_t)$, complete the EM update for γ below:

$\gamma \longleftarrow$ _____

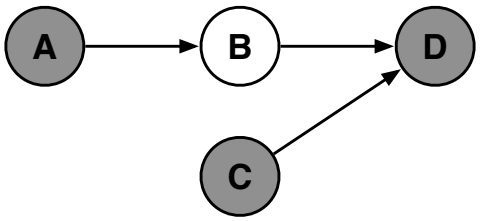
Note: you can answer this question correctly even if you skipped parts (d) and (e).

6. EM algorithm

(a) **Posterior probability** (3 pts)

Consider the belief network shown below, where B is a hidden node and A, C, and D are observed nodes. Show how to compute the posterior probability $P(B|A, C, D)$ in terms of the conditional probability tables (CPTs) of the belief network.

Justify your steps briefly to receive full credit.

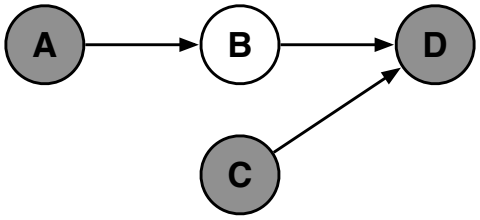


(b) **Log-likelihood** (3 pts)

Consider a data set of T partially labeled examples $\{a_t, c_t, d_t\}_{t=1}^{T}$ over the observed nodes of the network. The log (conditional) likelihood of the data set is given by:

$$\mathcal{L} = \sum_t \log P(D = d_t | A = a_t, C = c_t)$$

Compute this expression in terms of the CPTs of the belief network.
Justify your steps briefly to receive full credit.



(c) **EM algorithm** (3 pts)

Consider the EM algorithm that updates the CPTs to maximize the likelihood of the data set in part (b). Complete the numerator and denominator in the below expressions for updating the conditional probability tables $P(B=b|A=a)$ and $P(D=d|B=b,C=c)$.

$$P(B=b|A=a) \leftarrow \frac{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}$$

$$P(D=d|B=b,C=c) \leftarrow \frac{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}$$

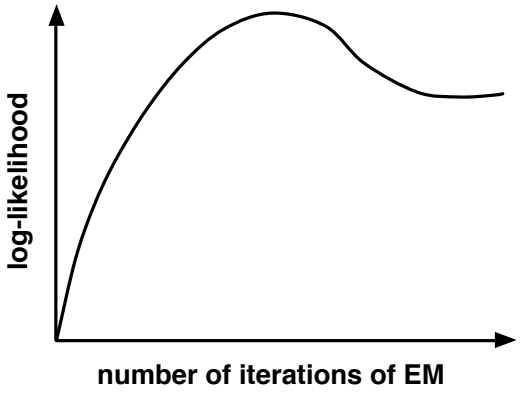
Suggested notation. Use $P(b|a_t,c_t,d_t)$ as shorthand for $P(B=b|A=a_t,C=c_t,D=d_t)$, the posterior probability computed in part (a). Also, use functions such as $I(a,a_t)$, where:

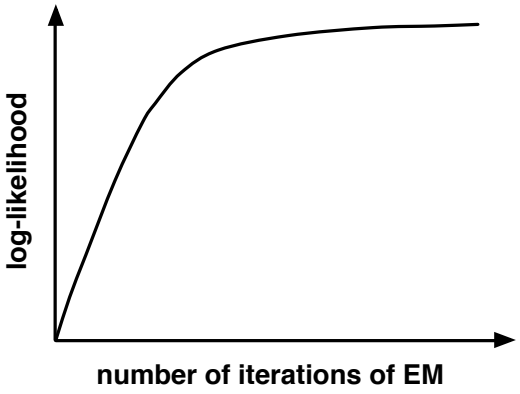
$$I(a,a_t) = \begin{cases} 1 \text{ if } a = a_t, \\ 0 \text{ if } a \neq a_t. \end{cases}$$

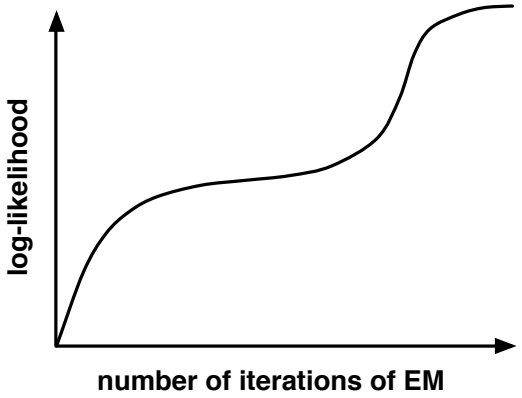
Simplify your answers as much as possible, and indicate clearly your final answers if they do not appear in the above space.

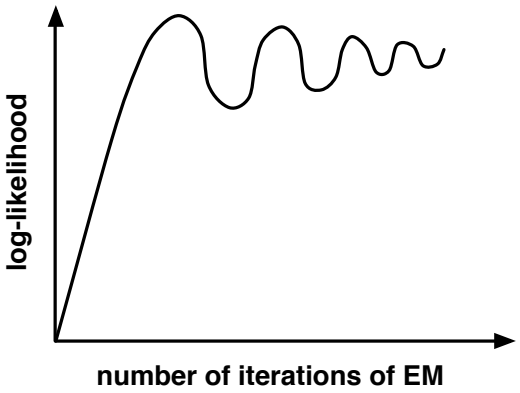
(d) **Convergence** (1 pt)

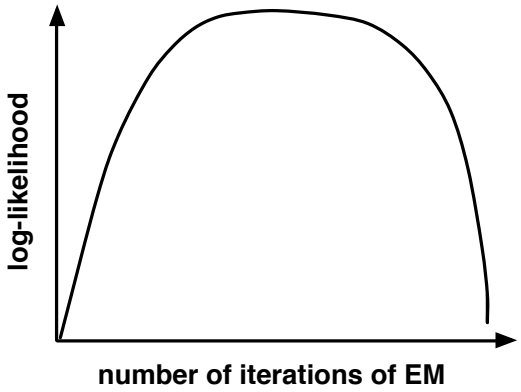
The EM algorithm typically converges to a local maximum of the log-likelihood. Circle the plots below that might be obtained from a correct implementation of the EM algorithm.









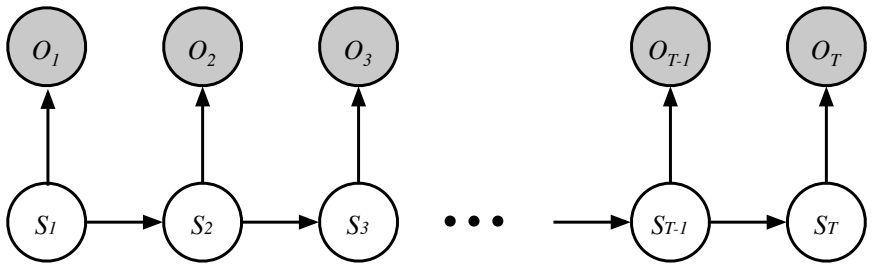


7. Inference in HMMs

Consider a discrete hidden Markov model (HMM) with the belief network shown below. Let $S_t \in \{1, 2, ..., n\}$ and $O_t \in \{1, 2, ..., m\}$ denote, respectively, the hidden state and observation at time t. Also, as usual, let

$$\begin{aligned} \pi_i &= P(S_1 = i), \\ a_{ij} &= P(S_{t+1} = j | S_t = i), \\ b_{ik} &= P(O_t = k | S_t = i), \end{aligned}$$

denote the initial distribution over hidden states, the transition matrix, and the emission matrix. In your answers you may also use $b_i(k)$ to denote the matrix element b_{ik}.



(Continued on next page)

(a) **Posterior transition matrix** (3 pts)

The forward-backward algorithm in discrete HMMs computes the probabilities:

$$\begin{aligned} \alpha_{it} &= P(o_1, o_2, \ldots, o_t, S_t = i), \\ \beta_{it} &= P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i). \end{aligned}$$

In terms of these probabilities (which you may assume to be given) and the parameters (a_{ij}, b_{ik}, π_i) of the HMM, show how to compute the posterior probability

$$P(S_{t+1} = j | S_t = i, o_1, o_2, \ldots, o_T)$$

for times $1 \leq t < T$. Simplify your answer as much as possible. Show your work for full credit, **briefly** justifying each step in your derivation.

(b) **First and last states** (4 pts)

In the last part of this problem, you computed the posterior transition matrices with matrix elements:

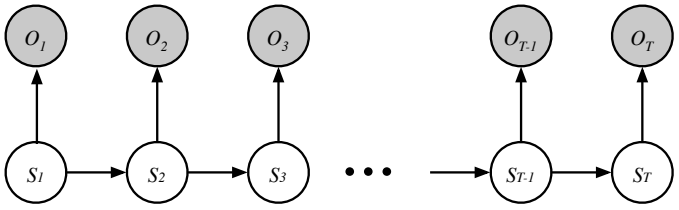
$$U_{ij}^{(t)} = P(S_{t+1}=j|S_t=i, o_1, o_2, \ldots, o_T).$$

In terms of these $n \times n$ matrices $U^{(t)}$, which you may assume to be given, consider how to efficiently compute the posterior probability $P(S_T=j|S_1=i, o_1, o_2, \ldots, o_T)$. In particular, prove (by induction or otherwise) that

$$P(S_T=j|S_1=i, o_1, o_2, \ldots, o_T) \;=\; \left[U^{(1)}U^{(2)}U^{(3)} \ldots U^{(T-1)}\right]_{ij},$$

where the right hand side denotes the ij^{th} element of the matrix product inside the brackets. **Note:** *you can answer this question even if you missed part (a).*

8. (Reverse) Viterbi algorithm



$$\begin{aligned} \pi_i &= P(S_1 = i), \\ a_{ij} &= P(S_{t+1} = j | S_t = i), \\ b_{ik} &= P(O_t = k | S_t = i), \end{aligned}$$

Consider a particular sequence of observations $(o_1, o_2, \ldots, o_T)$ for the discrete HMM shown above. In class we showed how to compute the log-probability of the most likely sequence of hidden states, from times 1 to t, that ends in state i:

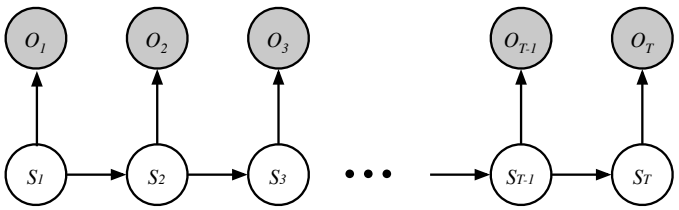
$$\ell^*_{it} = \max_{s_1,\ldots,s_{t-1}} \Big[\log P(s_1, \ldots, s_{t-1}, S_t = i, o_1, \ldots, o_t) \Big].$$

In this problem you will consider a related matrix. Let r^*_{it} denote the log-probability of the most likely sequence of hidden states, **from times $t+1$ to T, that begins in state i at time t**:

$$r^*_{it} = \max_{s_{t+1},\ldots,s_T} \Big[\log P(s_{t+1}, \ldots, s_T, o_{t+1}, \ldots, o_T | S_t = i) \Big].$$

It is also possible to derive the most likely sequence of hidden states, from times 1 to T, from the matrix r^*_{it}. This problem guides you through this derivation.

(go to next page)



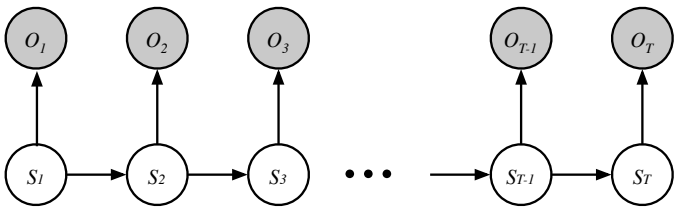
$$\begin{aligned} \pi_i &= P(S_1 = i), \\ a_{ij} &= P(S_{t+1} = j | S_t = i), \\ b_{ik} &= P(O_t = k | S_t = i), \end{aligned}$$

(a) **Base case** (2 pts)

We filled in the matrix ℓ^*_{it} from left to right, starting at time 1 and finishing at time T. We'll fill in the matrix r^*_{it} from right to left. Consider the $(T-1)^{\text{th}}$ column given by:

$$r^*_{i,T-1} = \max_j \Big[\log P(S_T = j, o_T | S_{T-1} = i) \Big].$$

Show how to compute these matrix elements at time $T-1$ in terms of the parameters a_{ij} and b_{ik} of the HMM, as well as the final observation o_T.



$$\begin{aligned} \pi_i &= P(S_1 = i), \\ a_{ij} &= P(S_{t+1} = j | S_t = i), \\ b_{ik} &= P(O_t = k | S_t = i), \end{aligned}$$

(b) **Backward pass** (4 pts)

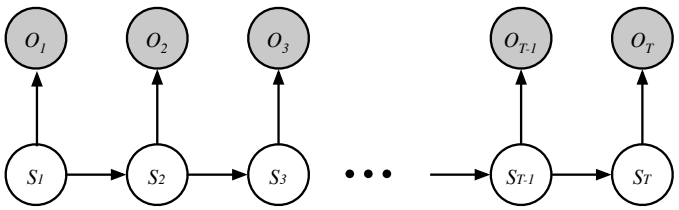
Next consider how to compute the elements r^*_{it} in the t^{th} column of the matrix from those in the $(t+1)^{\text{th}}$ column. Once again, these elements are defined as:

$$r^*_{it} = \max_{s_{t+1},\ldots,s_T} \Big[\log P(s_{t+1}, \ldots, s_T, o_{t+1}, \ldots, o_T | S_t = i) \Big].$$

Derive a simple recursion for these elements in terms of the (already computed) elements at time $t+1$, the parameters a_{ij} and b_{ik} of the HMM, and the observation o_{t+1}. *A helpful first step has been provided.*

$$r^*_{it} = \max_{s_{t+2},\ldots,s_T} \max_j \Big[\log P(S_{t+1} = j, s_{t+2}, \ldots, s_T, o_{t+1}, \ldots, o_T | S_t = i) \Big]$$

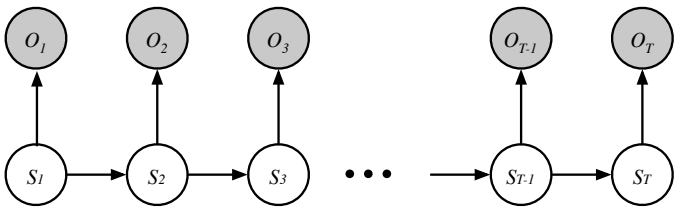
$$=$$



$$\begin{aligned} \pi_i &= P(S_1 = i), \\ a_{ij} &= P(S_{t+1} = j | S_t = i), \\ b_{ik} &= P(O_t = k | S_t = i), \end{aligned}$$

(c) **Sanity check** (1 pt)

Suppose that for the rightmost column of the matrix we define $r^*_{iT} = 0$ for all states i. Show in this case that your answer to part (b) reproduces your answer to part (a).



$$\begin{aligned} \pi_i &= P(S_1 = i), \\ a_{ij} &= P(S_{t+1} = j | S_t = i), \\ b_{ik} &= P(O_t = k | S_t = i), \end{aligned}$$

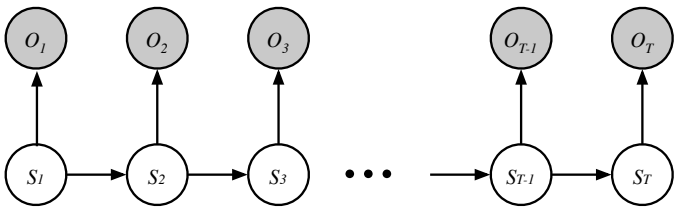
(d) **Initial state** (4 pts)

Suppose that you have computed the matrix elements r^*_{it} for all times $1 \leq t \leq T$. In particular, consider the elements of the first column:

$$r^*_{i1} = \max_{s_2,\ldots,s_T} \Big[\log P(s_2, \ldots, s_T, o_2, \ldots, o_T | S_1 = i) \Big].$$

Show how to compute the most likely hidden state s^*_1 at time $t = 1$ from these elements. (Be careful!) Your answer should also involve the parameters π_i and b_{ik} of the HMM, as well as the observation o_1 at time $t = 1$. *A helpful first step has been provided.*

$$\begin{aligned} s^*_1 &= \operatorname*{argmax}_i \max_{s_2,\ldots,s_T} \Big[\log P(S_1 = i, s_2, \ldots, s_T, o_1, \ldots, o_T) \Big] \\ &= \end{aligned}$$



$$\begin{aligned} \pi_i &= P(S_1 = i), \\ a_{ij} &= P(S_{t+1} = j | S_t = i), \\ b_{ik} &= P(O_t = k | S_t = i), \end{aligned}$$

(e) **Forward pass** (2 pts)

Show how to compute the most likely hidden states s_t^* at all times t given your results from parts (b-d). In particular, fill in the empty brackets below to complete the final line of pseudocode:

for $t = 2$ **to** T

$$i = s_{t-1}^*$$

$$s_t^* = \text{argmax}_j \Big[\qquad \Big]$$

9. Policy improvement

For this problem, consider the Markov decision process (MDP) with three states $s \in \{1, 2, 3\}$, two actions $a \in \{\uparrow, \downarrow\}$, discount factor $\gamma = \frac{3}{4}$, and these rewards and transition matrices:

s	$R(s)$
1	12
2	-3
3	6

s	s'	$P(s'\|s, a = \uparrow)$
1	1	$\frac{1}{3}$
1	2	$\frac{2}{3}$
1	3	0
2	1	0
2	2	$\frac{1}{3}$
2	3	$\frac{2}{3}$
3	1	$\frac{2}{3}$
3	2	0
3	3	$\frac{1}{3}$

s	s'	$P(s'\|s, a = \downarrow)$
1	1	$\frac{1}{3}$
1	2	0
1	3	$\frac{2}{3}$
2	1	$\frac{2}{3}$
2	2	$\frac{1}{3}$
2	3	0
3	1	0
3	2	$\frac{2}{3}$
3	3	$\frac{1}{3}$

(a) **Policy evaluation** (4 pts)

Compute the state-value function $V^\pi(s)$ for the policy π that chooses action $a = \uparrow$ in each state. Complete the table with your final answers, but also *show your work (on the following page) to receive full credit.*

Hint #1: one entry in the table is already provided (so you don't need to invert a 3x3 matrix).
Hint #2: the missing entries in the table are integers.

s	$\pi(s)$	$V^\pi(s)$
1	$\uparrow$	**24**
2	$\uparrow$	
3	$\uparrow$	

(a) **Policy evaluation** (con't)

(b) **Greedy policy** (4 pts)

Compute the greedy policy $\pi'(s)$ in states $s=1$ and $s=2$ with respect to the policy $\pi(s)$ and state-value function $V^{\pi}(s)$ *that is shown below*. Complete the table with your final answer, but *show your work to receive full credit*.

Note that this is **not** the same policy and state-value function that you examined in part (a); thus, you can answer this question correctly even if you missed part (a).

s	$\pi(s)$	$V^{\pi}(s)$	$\pi'(s)$
1	$\uparrow$	24	
2	$\downarrow$	12	
3	$\downarrow$	16	$\uparrow$

10. Temporal difference learning

Consider how to estimate the expected value $\mathrm{E}[X]$ of a scalar random variable X from a sequence of independently, identically distributed (i.i.d.) observations $(x_1, x_2, x_3, \ldots)$. A simple, online estimate can be computed by the method of temporal differences (TD). In particular, let μ_t denote the estimate of $\mathrm{E}[X]$ after t observations. The temporal difference update rule is

$$\mu_{t+1} \;=\; \mu_t + \alpha_{t+1}(x_{t+1} - \mu_t),$$

where the learning rates $\alpha_t \in [0, 1]$ must be chosen appropriately to ensure convergence as $t \to \infty$. In this problem, you will work out an example of this procedure.

(a) **Gaussian?** (1 pt)

Suppose that X is a Gaussian random variable. Does it follow that μ_t is Gaussian distributed? Briefly justify your answer.

(b) **Conditions for convergence** (1 pt)

In lecture we stated two necessary conditions on the growth of the learning rates α_t for convergence; i.e., so that $\lim_{t\to\infty} \mu_t = \mathrm{E}[X]$. Restate these conditions.

(c) **Constant learning rate** (1 pt)

Consider the choice of a *small but constant* learning rate: $\alpha_t = \alpha$ for all t, where $0 < \alpha < 1$. Does this choice of learning rate satisfy your conditions in part (b)? If not, state which condition from part (b) is violated.

(d) **Estimate after T updates** (3 pts)

Suppose that the TD estimate is initialized as $\mu_0 = 0$. The TD update rule for a *constant learning rate* α is given by

$$\mu_{t+1} \;=\; \mu_t + \alpha(x_{t+1} - \mu_t).$$

For this constant learning rate, prove (by induction or otherwise) that after T updates of the learning rule, we estimate:

$$\mu_T \;=\; \alpha \sum_{t=1}^{T} (1-\alpha)^{T-t} x_t.$$

(e) **Asymptotic mean** (2 pts)

In part (d) of this problem, you showed that for constant learning rate α, the TD estimate at time T is given by

$$\mu_T \;=\; \alpha \sum_{t=1}^{T} (1-\alpha)^{T-t} x_t.$$

Starting from this result, show that for constant learning rate the expected value of the TD estimate converges to the expected value of X; in other words:

$$\lim_{T \to \infty} \mathrm{E}[\mu_T] = \mathrm{E}[X].$$

Note: *you can answer this question even if you were unable to answer the previous one.*

(f*) **Asymptotic variance** (3 pts)

In part (d) of this problem, you showed that for constant learning rate α, the TD estimate at time T is given by

$$\mu_T \;=\; \alpha \sum_{t=1}^{T} (1-\alpha)^{T-t} x_t.$$

Suppose that the random variable X has variance $\sigma^2 = \mathrm{E}[(X - \mathrm{E}[X])^2]$. Starting from the above result, show that the expected variance of the TD estimate in this case converges to:

$$\lim_{T\to\infty} \mathrm{E}[(\mu_T - \mathrm{E}[\mu_T])^2] = \left(\frac{\alpha}{2-\alpha}\right)\sigma^2.$$

Note that the right hand side does not vanish: the asymptotic variance is finite. This observation should be consistent with your answer in part (c) stating whether or not the TD estimate converges with a small but constant learning rate.