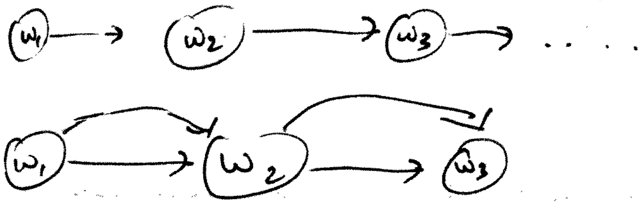
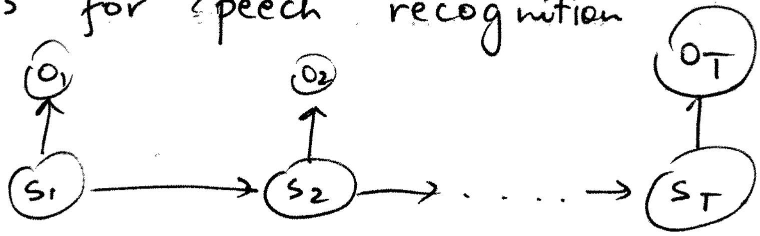
Compact expressions of complex worlds

1) noisy-OR
2) naive Bayes
3) Markov models of language

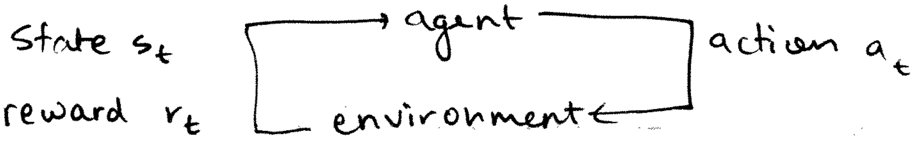
W_ℓ = ℓth word in sentence



4) HMMs for speech recognition

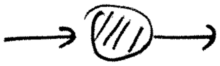


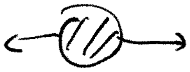
5) MDPs for planning

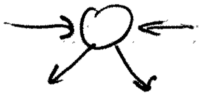


Efficient algorithms

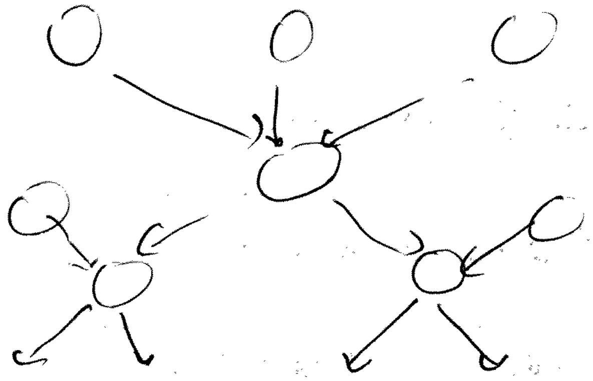
1) Conditional independence tests via d-separation

(i) 

(ii) 

(iii) 

2) polytree algorithm for inference



3) EM algorithm for ML estimation

update $P(X_i = x \mid pa_i = \pi) \longleftarrow$

$$\frac{\sum_t P(X_i = x, pa_i = \pi \mid V^{(t)})}{\sum_t P(pa_i = \pi \mid V^{(t)})}$$

guarantee = monotonic convergence

$$\mathcal{L} = \sum_t \log P(V^{(t)})$$

never decreases,

generally increases at each iteration..

4) HMM's

- computing likelihood $P(O_1 O_2 \ldots O_T)$
- decode argmax $P(S_1 S_2 \ldots S_T \mid O_1 O_2 \ldots O_T)$
- belief updating - learning

5) Algorithms in MDPs

- Policy iteration

$$\pi_0 \xrightarrow{\text{evaluate}} \begin{matrix} V^{\pi_0}(s) \\ Q^{\pi_0}(s,a) \end{matrix} \xrightarrow[\text{greedy}]{\text{improve}} \pi_1 \longrightarrow \dots$$

- Value iteration

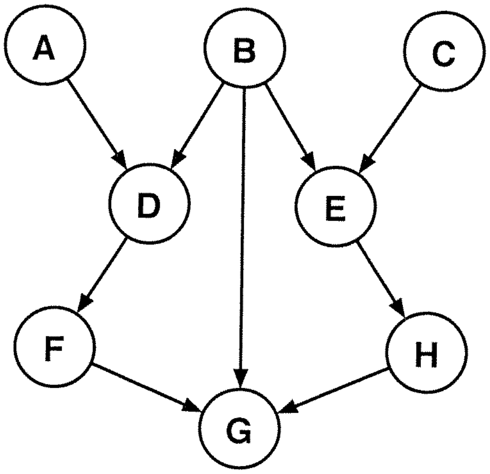
$$V_{k+1}(s) \leftarrow R(s) + \gamma \max_a \sum_{s'} P(s'|s,a) V_k(s')$$

$Q^\pi(s,a)$ = expected return if take action a at $t=0$, in state s, then follow policy π.

$$= E^\pi \left[\sum_{t=0}^{\infty} \gamma^t R(s_t) \mid s_0 = s, a_0 = a \right]$$

2. Conditional independence (10 pts)

For the belief network shown below, indicate whether the following statements of conditional independence are **true (T)** or **false (F)**.



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

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

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

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

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

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

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

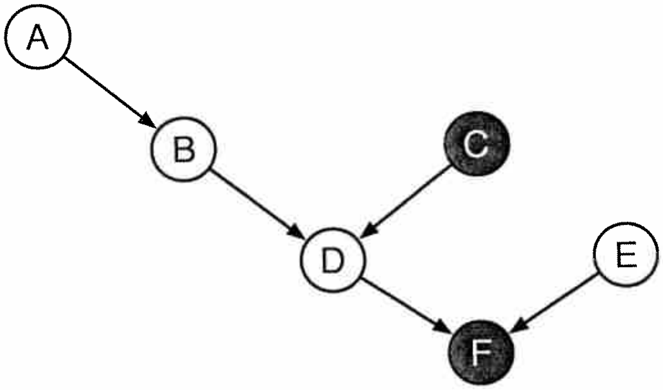
______ $P(F|C,E) = P(F|E)$

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

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

(*) $P(A,B,C|D,E) = P(C|D,E)\,P(B|C,D,E)\,\underbrace{P(A|B,C,D,E)}_{\overset{?}{=}\ P(A|B,D)}$

3. Polytree inference



Inference in a polytree scales linearly in the number of nodes and the sizes of their conditional probability tables (CPTs). For the belief network shown above, consider how to *efficiently* compute the posterior probability $P(E|C, F)$. This can be done in four consecutive steps in which the later steps rely on the results from earlier ones.

Complete the procedure below for this inference by showing how to compute the necessary result at each step. For full credit, make each step as efficient as possible. Your answers should be expressed in terms of the CPTs of the belief network as well as the results of previous steps. **Hint:** *at each step, you'll want to exploit what you've just computed in the previous one.*

(a) **Marginal** (2 pts)

$P(B) = \sum_a P(A=a, B)$ Marginalization (M)

$= \sum_a P(A=a)\, P(B|A=a)$ product rule (PR)

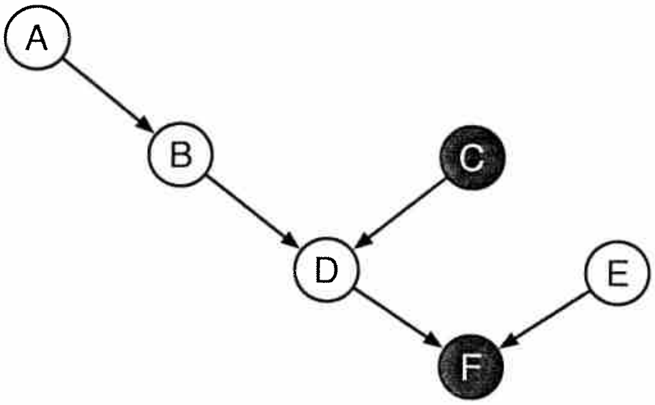
(b) **Conditional** (3 pts)

$P(D|C) = \sum_b P(B=b, D|C)$ (M)

$= \sum_b P(B=b|\cancel{C})\, P(D|B=b, C)$ (PR) (CI)

$= \sum_b P(B=b)\, P(D|B=b, C)$ //

3. Polytree inference (con't)



(c) **Conditional** (3 pts)

$P(F|C,E)$

Marginalization over D
prod rule
CI

$$= \sum_d \underbrace{P(D=d|C)}_{(b)} \underbrace{P(F|D=d,E)}_{CPT}$$

(d) **Posterior** (5 pts)

$$P(E|C,F) = \frac{\overbrace{P(F|E,C)}^{(c)} P(E|C)}{P(F|C)}$$ Bayes rule

$$= \frac{P(F|E,C)\ P(E)}{P(F|C)}$$ CI

$$= \cancel{P(E)} \quad \frac{P(F|E,C)\ P(E)}{\sum_e P(F|E=e,C)\ P(E=e)}$$ normalization.

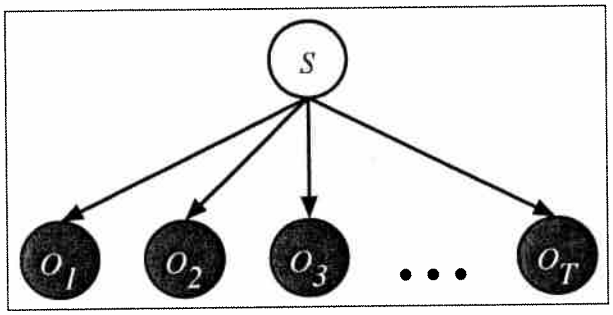
3. Belief updating

(a) **Naive Bayes model** (3 pts)

Consider the belief network shown below for the discrete random variables $S \in \{1, 2, \ldots, n\}$ and $O_t \in \{1, 2, \ldots, m\}$. Also, let the CPTs of the network be parameterized by:

$$\begin{aligned} \pi_i &= P(S{=}i), \\ b_{ik} &= P(O_t{=}k|S{=}i), \end{aligned}$$

where the same CPT is used for all the child nodes. In terms of these parameters, show how to compute the posterior probability $P(S{=}i|o_1, o_2, \ldots, o_T)$. You may use the notation $b_i(o_t)$ as shorthand for $P(o_t|S{=}i)$.



$$P(S=i \mid O_1, O_2 \ldots O_T)$$

$$= \frac{P(O_1, O_2 \ldots O_T \mid S=i)\, P(S=i)}{P(O_1 \ldots O_T)} \quad \text{Bayes}$$

Product rule, CI

$$= \frac{P(S=i) \times \left(P(O_1|S=i)\, P(O_2|S=i) \ldots P(O_T|S=i) \right)}{P(O_1 \ldots O_T)}$$

$$= \frac{\pi_i \prod_{t=1}^{T} P(O_t|S=i)}{\sum_{j=1}^{n} \pi_j \prod_{t=1}^{T} P(O_t|S=j)}$$

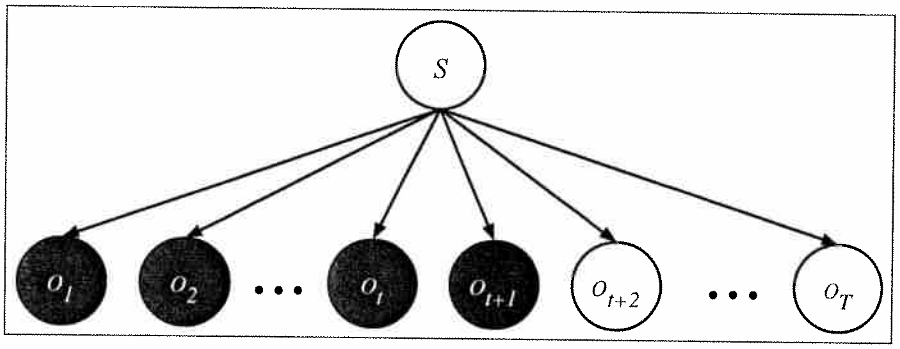
normalization

(b) **Belief updating** (5 pts)

Consider the same belief network as before. In this problem, you will derive an incremental update rule for the scenario where the observations arrive one at a time. Let

$$q_{it} = P(S{=}i|o_1, o_2, \ldots, o_t)$$

denote the posterior probability over the hidden variable S based on the first t observations. Derive a recursion for these probabilities at time $t+1$ in terms of those at time t. (Your answer should also involve the parameters of the belief network and the observation o_{t+1} at time $t+1$.)



Note: the incremental update should require only $O(n)$ operations, **not** $O(nt)$ or $O(nT)$ operations as presumably did your answer to part (a).

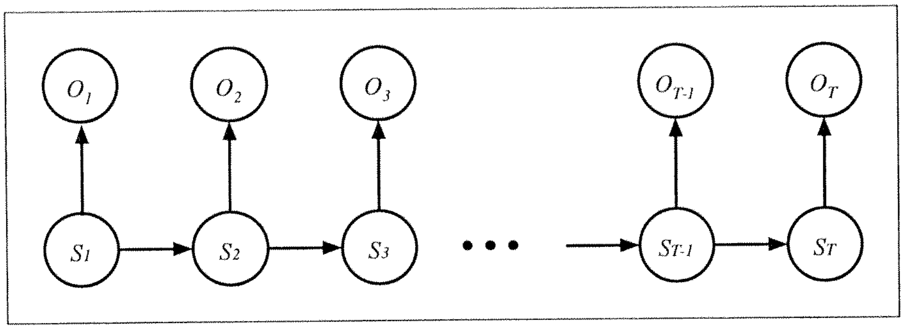
$$q_{i,t} = P(s=i|o_1, .. o_t)$$

$$q_{i,t+1} = P(s=i|o_1, o_2, \ldots, o_{t+1})$$

$$= \frac{P(o_{t+1}|s=i, o_1, \ldots o_t)\, P(s=i|o_1, o_2 \ldots o_t)}{P(o_{t+1}|o_1\, o_2 \ldots o_t)} \quad \text{Bayes rule}$$

$$= \frac{P(o_{t+1}|s=i)\, P(s=i|o_1 .. o_t)}{P(o_{t+1}|o_1 \ldots o_t)} \quad \text{CI}$$

$$= \frac{b_i(o_{t+1})\, q_{it}}{\sum_j b_j(o_{t+1})\, q_{jt}} \quad \begin{array}{l}\text{substitution} \\ \text{normalization.}\end{array}$$



(c) **Comparison to HMM** (2 pts)

Now consider a discrete HMM where as usual $S_t \in \{1, 2, ..., n\}$ denotes the hidden state at time t, $O_t \in \{1, 2, ..., m\}$ denotes the observation at time t, and the parameters

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

denote the transition and emission matrices. Also let $q_{it} = P(S_t = i|o_1, o_2, \ldots, o_t)$ denote the posterior over the hidden state S_t at time t after seeing t observations. In an HMM, the equation for belief updating is given by:

$$q_{j,t+1} = \frac{\sum_i q_{it}a_{ij}b_j(o_{t+1})}{\sum_{i'j'} q_{i't}a_{i'j'}b_{j'}(o_{t+1})}.$$

Suppose we consider the trivial HMM in which the hidden state *never* changes. This can be enforced by setting the transition matrix a_{ij} equal to the identity matrix: that is,

$$a_{ij} = I(i, j),$$

where $I(i, j)$ is the usual indicator function. For this special case, does the above formula for belief updating reduce to your answer in part (b)? If yes, show that it does; if not, explain why it doesn't.

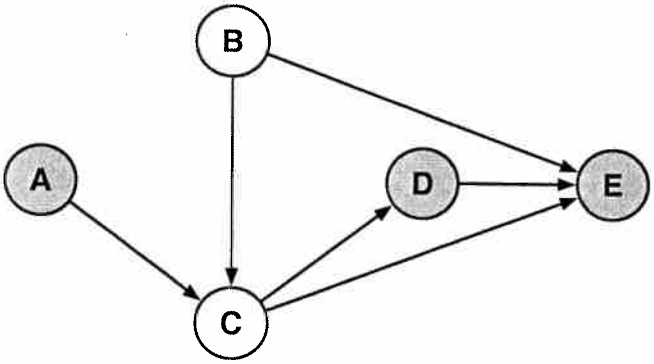
$$q_{j,t+1} = \frac{\sum_i q_{it}\, a_{ij}\, b_j(o_{t+1})}{\sum_{i',j'} q_{i't}\, a_{i'j}\, b_{j'}(o_{t+1})}$$

$$= \frac{\sum_{i=1}^{n} q_{it}\, I(i,j)\, b_j(o_{t+1})}{[\qquad\qquad]}$$

$$= \frac{q_{jt}\, b_j(o_{t+1})}{\sum_{j'} q_{j't}\, b_{j'}(o_{t+1})}$$

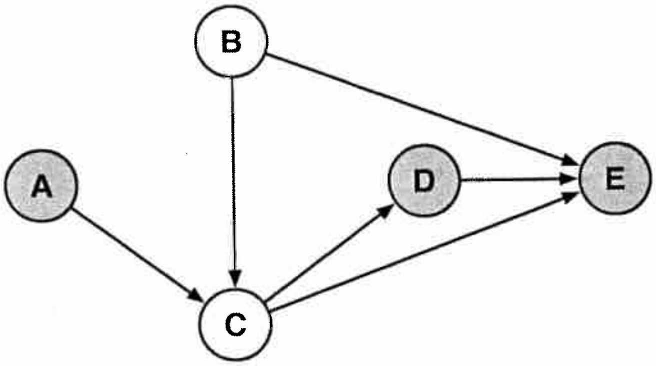
1. EM algorithm

Consider the belief network, shown below, with conditional probability tables $P(A)$, $P(B)$, $P(C|A, B)$, $P(D|C)$, and $P(E|B, C, D)$. Suppose that values of the nodes B and C are unobserved (hidden), while the values of the shaded nodes A, D, and E are observed.



(a) **Posterior probability** (4 pts)

Show how to compute the posterior probability $P(B = b, C = c|A = a, D = d, E = e)$ in terms of the belief network's conditional probability tables. You may use shorthand notation to simplify your answer: i.e., $P(b, c|a, d, e) = P(B{=}b, C{=}c|A{=}a, D{=}d, E{=}e)$.



(b) **Posterior probability** (2 pts)

Compute the posterior probabilities $P(b|a, d, e)$ and $P(c|a, d, e)$ in terms of your answer from part (a); in other words, for this problem you may assume that the posterior probability $P(b, c|a, d, e)$ is given.

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

Consider a data set of T partially labeled examples $\{a_t, d_t, e_t\}_{t=1}^{T}$ over the nodes A, D, and E of the belief network. The log-likelihood of the data is given by:

$$\mathcal{L} = \sum_{t=1}^{T} \log P(A=a_t, D=d_t, E=e_t)$$

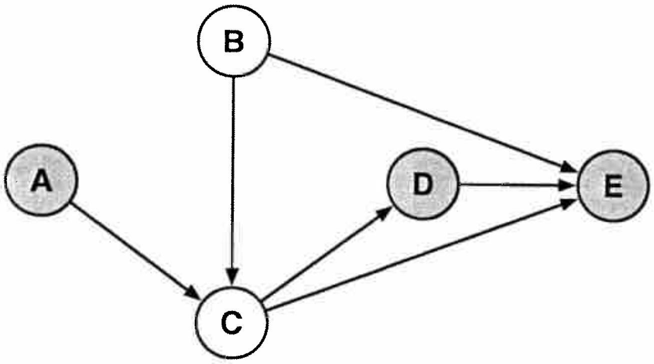
Compute this log-likelihood in terms of the CPTs of the belief network.

Marginalization

$$= \sum_{t} \log \sum_{b,c} P(a_t, B=b, C=c, d_t, e_t)$$

$$= \sum_{t} \log \sum_{b,c} P(a_t)\, P(b)\, P(c|a_t, b)\, P(d_t|c)\, P(e_t|b, c, d_t)\Big]$$

PR & CI



(d) **EM algorithm** (8 pts)

Consider the EM algorithm that updates the CPTs to maximize the log-likelihood of the data set in part (c). Complete the numerator and denominator in the expressions for the EM updates shown below and on the following pages.

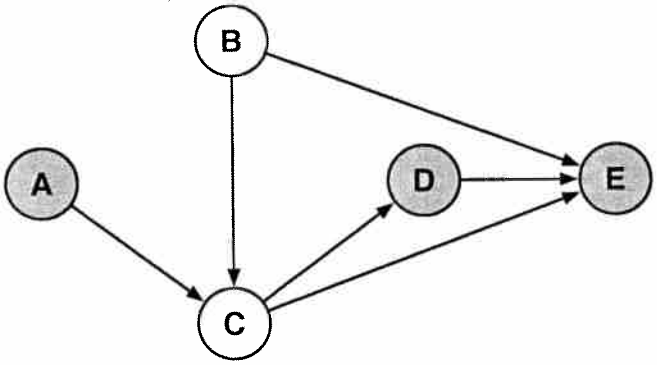
Suggested notation. Use shorthand such as $P(b, c|a_t, d_t, e_t)$ for the posterior probabilities computed in parts (a) and (b). Also, use indicator functions such as $I(a, a_t)$, where:

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

Simplify your answers as much as possible, and put your *final* answers in the spaces provided below.

$$P(B{=}b) \leftarrow \frac{\sum_{t=1}^{T} P(b|a_t, d_t, e_t)}{T}$$

$$P(C{=}c|A{=}a, B{=}b) \leftarrow \frac{\sum_{t=1}^{T} I(a, a_t)\, P(b, c|a_t, d_t, e_t)}{\sum_{t=1}^{T} I(a, a_t)\, P(b|a_t, d_t, e_t)}$$



(d) **EM algorithm** (continued)

$P(D\!=\!d|C\!=\!c) \;\leftarrow\;$ ________________________________

$$P(E\!=\!e|B\!=\!b, C\!=\!c, D\!=\!d) \;\leftarrow\; \frac{\sum_t I(d, d_t) I(e, e_t) P(b, c | a_t, d_t, e_t)}{\sum_t I(d, d_t) P(c, b | a_t, d_t, e_t)}$$

(e) **Multiple choice** (3 pts)

Consider the EM algorithm for belief networks of discrete nodes with arbitrarily specified CPTs. Indicate the *best* answer to each statement in the space provided.

______ **In practice, the EM algorithm for these belief networks will generally:**

(a) converge to a *global maximum* of the log-likelihood.
[✓ (b)] converge to a *local maximum* of the log-likelihood.
(c) converge to a *global minimum* of the log-likelihood.
(d) converge to a *local minimum* of the log-likelihood.
(e) not converge.

______ **The EM algorithm for these belief networks is guaranteed to converge:**

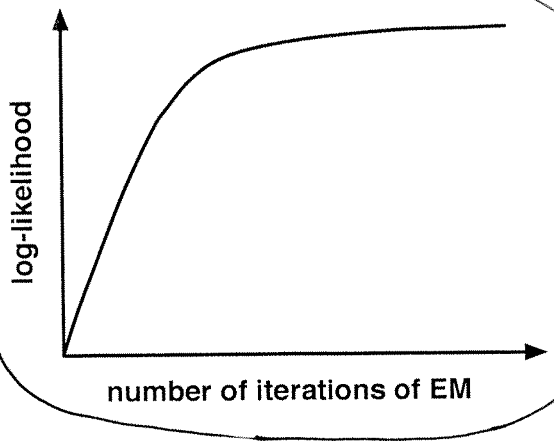
(a) within T iterations, where T is the number of examples.
(b) only when the parameters are initialized by a domain expert.
(c) as long as there are no duplicate examples in the data set.
[✓ (d)] monotonically, never decreasing the log-likelihood at each iteration.
(e) none of the above (unless the belief network happens to be a polytree).

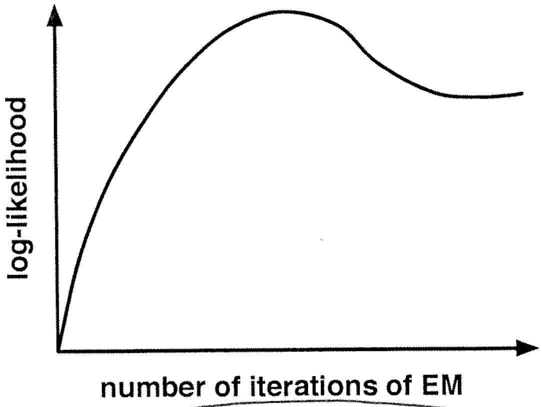
______ **In general each iteration of the EM algorithm scales:**

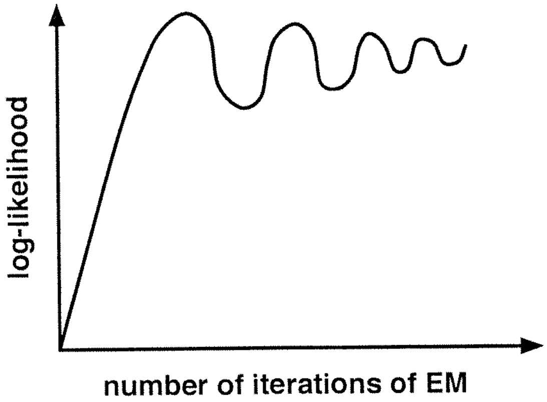
[✓ (a)] linearly in the number of examples, T.
(b) linearly in the number of edges of the belief network.
(c) inversely in the number of observed nodes.
(d) exponentially in the number of hidden nodes.
(e) all of the above.
(f) none of the above.

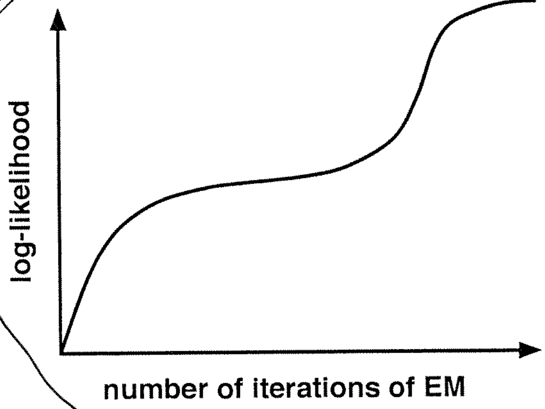
(e) **Convergence** (5 pts)

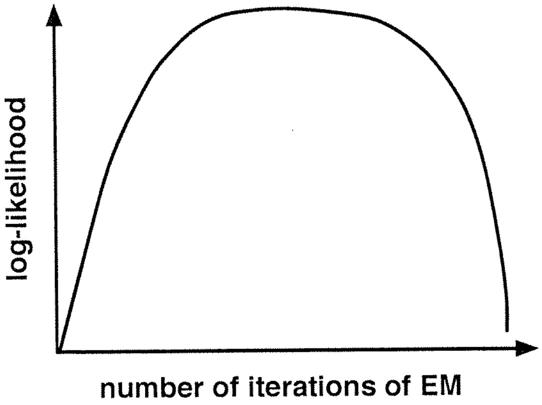
The EM algorithm generally converges to a local maximum of the log-likelihood. Circle the plots below that might be obtained from a correct implementation of the EM algorithm.







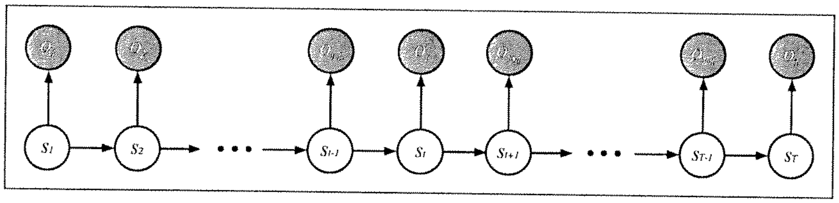




2. Hidden Markov modeling

Consider a discrete hidden Markov model (HMM) with the belief network shown below. Let $S_t \in \{1, 2, ..., n\}$ denote the hidden state of the HMM at time t, and let $O_t \in \{1, 2, ..., m\}$ denote the observation at time t. As in class, 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 parameters of the HMM. You may also use $b_i(k)$ to denote the matrix element b_{ik}.

(a) **Backward algorithm** (4 pts)

Consider the probabilities $\beta_{it} = P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i)$. Derive an efficient recursion to compute these probabilities at time $t < T$ from those at time $t+1$ and the transition and emission matrices of the HMM. Justify briefly each step in your derivation.

$$\beta_{it} = P(o_{t+1}, o_{t+2}, \ldots, o_T | S_t = i)$$

$$= \sum_j P(S_{t+1}=j, O_{t+1}, \ldots O_T \mid S_t = i) \quad \text{marginalization}$$

PR & CI

$$= \sum_j P(S_{t+1}=j | S_t = i) P(O_{t+1} | \cancel{S_t=i}, S_{t+1}=j) P(O_{t+2} \ldots O_T | \cancel{S_t=i}, S_{t+1}=j, \cancel{O_{t+1}})$$

$$= \sum_j P(\overset{=j}{S_{t+1}} | S_t = i) \; P(O_{t+1} | S_{t+1}=j) P(O_{t+2} \ldots O_T \mid S_{t+1}=j)$$

$$= \sum_j a_{ij} \, b_j(O_{t+1}) \, \beta_{j,t+1}$$

(d) **Viterbi algorithm** (6 pts)

For a particular sequence of observations $(o_1, o_2, \ldots, o_T)$, suppose that recursive algorithms are implemented to compute the $n \times T$ matrices (from lecture) with elements:

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

$$\Phi_{j,t+1} = \operatorname*{argmax}_i \Big[\ell^*_{it} + \log a_{ij}\Big]$$

(i) From the matrix elements shown above, which you may assume to be given, show how to compute the most likely hidden state sequence

$$\{s^*_1, s^*_2, \ldots, s^*_T\} = \operatorname*{argmax}_{s_1,\ldots,s_T} P(s_1,s_2,\ldots,s_T|o_1,o_2,\ldots,o_T).$$

In particular, do this by completing the following pseudocode for the Viterbi algorithm (making use of the matrices defined above):

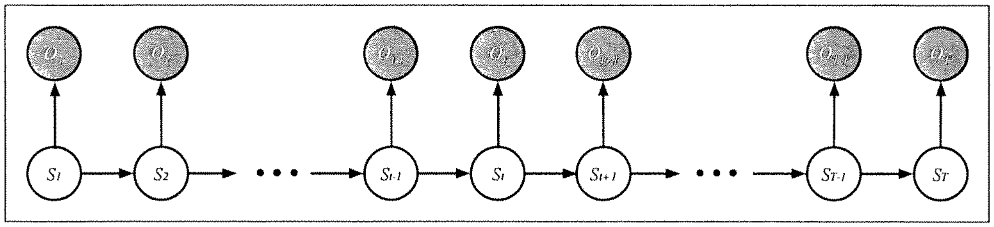
```
$s^*_T = \operatorname*{argmax}_i [\ell^*_{iT}]$

for $t = T-1$ to 1

    $s^*_t = \Phi(s^*_{t+1}, t+1)$
```

(ii) Shown below are the matrix elements Φ_{it} (for $i\!=\!1$ to n and $t\!=\!1$ to T) for a particular observation sequence $\{o_1,\ldots,o_T\}$ of an n-state HMM with $n = 5$ and $T = 6$. Given that $s_T = 4$, use the matrix elements to deduce the missing values of the most likely state sequence:

4	4	3	1	3	4
s_1	s_2	s_3	s_4	s_5	s_6

$$\Phi = \begin{pmatrix} * & 1 & 2 & \boxed{3} & 4 & 5 \\ * & 2 & 3 & 4 & 5 & 1 \\ * & 3 & \boxed{4} & 5 & \boxed{1} & 2 \\ * & \boxed{4} & 5 & 1 & 2 & \boxed{3} \\ * & 5 & 1 & 2 & 3 & 4 \end{pmatrix}$$

Note: the asterisk symbols $(*)$ indicate undefined elements at time $t\!=\!1$.



(d*) **Most likely hidden state** (2 pts)

As before, suppose that for a particular sequence of observations, the forward-backward algorithm in HMMs is used to compute 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, show how to compute the most likely value of the (single) hidden state at time t:

$$\tau_t^* = \underset{i}{\text{argmax}} \Big[P(S_t{=}i | o_1, o_2, \ldots, o_T) \Big].$$

Show your work for full credit, briefly justifying each step in your derivation.

$$= \underset{i}{\text{argmax}} \left[\frac{P(S_t=i, O_1, O_2 \ldots O_T)}{P(O_1 \ldots O_T)} \right]$$ prod rule

$$= \underset{i}{\text{argmax}} \left[P(S_t=i, O_1 \ldots O_T) \right]$$ den. ind of i

$$= \underset{i}{\text{argmax}} \left[P(S_t=i, O_1 \ldots O_t)\, P(O_{t+1} \ldots O_T | S_t=i, \cancel{O_1 \ldots O_t}) \right]$$ prod rule, CI

$$= \underset{i}{\text{argmax}} \left[\alpha_{it} \beta_{it} \right]$$

(e*) **Compare and contrast** *(2 pts)*

The Viterbi algorithm computes the most likely *sequence* of hidden states for a particular sequence of observations:

$$\{s_1^*, s_2^*, \ldots, s_T^*\} \;=\; \underset{\{s_1, s_2, \ldots, s_T\}}{\operatorname{argmax}} \left[P(s_1, s_2, \ldots, s_T | o_1, o_2, \ldots, o_T)\right]$$

Consider how these collectively optimal states s_t^* differ (if at all) from the individually optimal states τ_t^* defined in part (d) of this problem:

$$\tau_t^* \;=\; \underset{i}{\operatorname{argmax}}\left[P(S_t\!=\!i | o_1, o_2, \ldots, o_T)\right].$$

Answer the following `[yes/no]` questions:

no Is it always true that $\tau_t^* = s_t^*$ for all t?

no Is it always true that $P(\tau_1^*, \tau_2^*, \ldots, \tau_T^*) > 0$?

Yes Is it possible that $\tau_t^* = s_t^*$ for all t?

never!! Is it possible that $P(\tau_1^*, \ldots, \tau_T^* | o_1, \ldots, o_T) > \underbrace{P(s_1^*, \ldots, s_T^* | o_1, \ldots, o_T)}_{s_t^* \text{ defined to maximize this}}$?

1. Suppose X, Y, and Z are random variables. Which of the following expressions can be used to compute $P(X|Y, Z)$?

   (a) $P(X, Y, Z)/P(Y, Z)$

   (b) $P(Y|X, Z)P(X|Z)/P(Y|Z)$

   (c) $P(Y, Z|X)P(X)/P(Y, Z)$

   (d) some of the above

   (e) all of the above

2. In a polytree belief network,

   (a) each node has no more than one parent.

   [✓ (b)] there is at most one undirected path between any two nodes.

   (c) the graph is a disjoint union of trees.

   (d) each loop in the graph contains exactly one root node.

3. A naive Bayes model for classification of multivariate data assumes that

   (a) the features of the data are marginally independent.

   (b) the features of the data are conditionally independent given the label.

   (c) the features of the data are uniformly distributed within the unit hypercube.

   (d) there are fewer class labels than features.

4. Consider a discrete hidden Markov model with n hidden states and m observations. Given an observation sequence of length T, the Viterbi algorithm computes the most likely hidden state sequence in time

   (a) $O(n^2m^2T)$

   (b) $O(m^2T)$

   [✓ (c)] $O(n^2T)$

   (d) $O(n^T)$

5. Let W_k represent the kth word in a sentence. Which of the following is true in a trigram model of natural language?

   (a) $P(W_k) = P(W_k|W_{k-1}, W_{k-2})$

   (b) $P(W_{k-1}, W_k, W_{k+1}) = P(W_{k-1})P(W_k)P(W_{k+1})$

   (c) $P(W_k|W_{k-1}, W_{k-2}) = P(W_k|W_1, W_2, \ldots, W_{k-1})$

   (d) $P(W_k|W_{k-1}, W_{k-2}) = P(W_k|W_1, W_2, \ldots, W_{k-1}, W_{k+1}, W_{k+2}, \ldots, W_{\text{final}})$