

---

## CSE 250a. Assignment 6

**Out:** Thu Nov 08

**Due:** Tue Nov 13

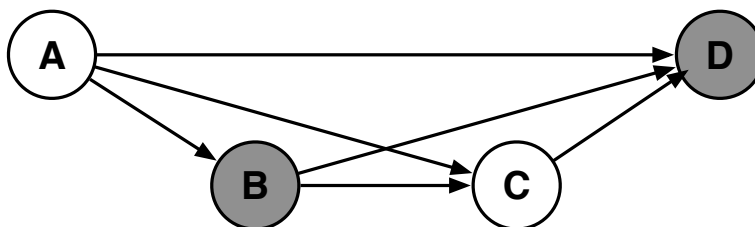
---

### 6.1 Survey

As soon as possible please fill out the short survey on movies at this link: <https://tinyurl.com/yawe7yqy>. We are collecting this data for an upcoming assignment in which you will build a simple movie recommendation system and use it to make personalized recommendations based on your own tastes. *Note that you will not be able to complete this assignment if you do not fill out the survey by the due date.*

---

### 6.2 EM algorithm



(a) **Posterior probability**

Consider the belief network shown above, with observed nodes  $B$  and  $D$  and hidden nodes  $A$  and  $C$ . Compute the posterior probability  $P(a, c|b, d)$  in terms of the CPTs of the belief network—that is, in terms of  $P(a)$ ,  $P(b|a)$ ,  $P(c|a, b)$  and  $P(d|a, b, c)$ .

(b) **Posterior probability**

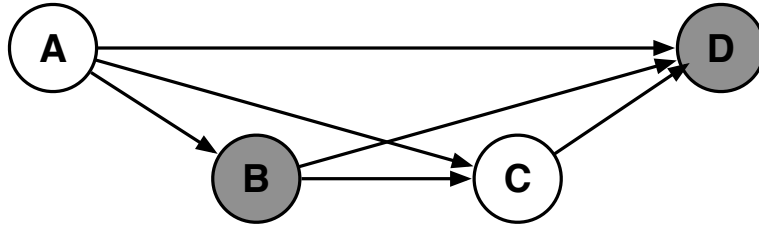
Compute the posterior probabilities  $P(a|b, d)$  and  $P(c|b, d)$  in terms of your answer from part (a); that is, for this problem, you may assume that  $P(a, c|b, d)$  is given.

(c) **Log-likelihood**

Consider a partially complete data set of *i.i.d.* examples  $\{b_t, d_t\}_{t=1}^T$  drawn from the joint distribution of the above belief network. The log-likelihood of the data set is given by:

$$\mathcal{L} = \sum_t \log P(B=b_t, D=d_t).$$

Compute this log-likelihood in terms of the CPTs of the belief network. You may re-use work from earlier parts of the problem.



(d) **EM algorithm**

Give the EM updates to estimate CPTs that maximize the log-likelihood in part (c); in particular, complete the numerator and denominator in the below expressions for the update rules. Simplify your answers as much as possible, expressing them in terms of the posterior probabilities  $P(a, c|b_t, d_t)$ ,  $P(a|b_t, d_t)$ , and  $P(c|b_t, d_t)$ , as well as the functions  $I(b, b_t)$ , and  $I(d, d_t)$ .

$$P(A=a) \leftarrow \frac{\quad}{\quad}$$

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

$$P(C=c|A=a, B=b) \leftarrow \frac{\quad}{\quad}$$

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

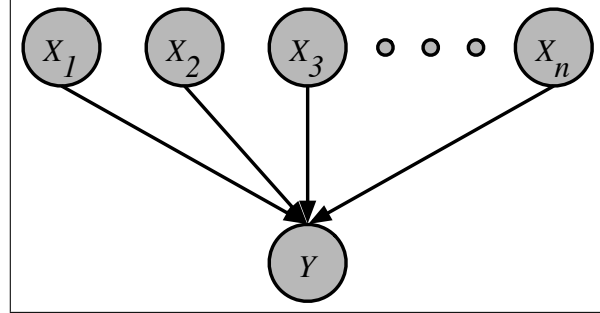
### 6.3 EM algorithm for noisy-OR

Consider the belief network on the right, with binary random variables  $X \in \{0, 1\}^n$  and  $Y \in \{0, 1\}$  and a noisy-OR conditional probability table (CPT). The noisy-OR CPT is given by:

$$P(Y = 1|X) = 1 - \prod_{i=1}^n (1 - p_i)^{X_i},$$

which is expressed in terms of the noisy-OR parameters  $p_i \in [0, 1]$ .

In this problem, you will derive and implement an EM algorithm for estimating the noisy-OR parameters  $p_i$ . It may seem that the EM algorithm is not suited to this problem, in which all the nodes are observed, and the CPT has a parameterized form. In fact, the EM algorithm can be applied, but first we must express the model in a different but equivalent form.

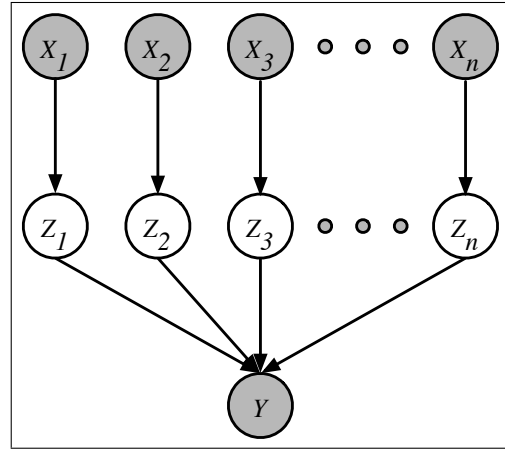


Consider the belief network shown to the right. In this network, a binary random variable  $Z_i \in \{0, 1\}$  intercedes between each pair of nodes  $X_i$  and  $Y$ . Suppose that:

$$\begin{aligned} P(Z_i = 1|X_i = 0) &= 0, \\ P(Z_i = 1|X_i = 1) &= p_i. \end{aligned}$$

Also, let the node  $Y$  be *determined* by the logical-OR of  $Z_i$ . In other words:

$$P(Y = 1|Z) = \begin{cases} 1 & \text{if } Z_i = 1 \text{ for any } i, \\ 0 & \text{if } Z_i = 0 \text{ for all } i. \end{cases}$$



- (a) Show that this “extended” belief network defines the same conditional distribution  $P(Y|X)$  as the original one. In particular, starting from

$$P(Y = 1|X) = \sum_{Z \in \{0,1\}^n} P(Y = 1, Z|X),$$

show that the right hand side of this equation reduces to the noisy-OR CPT with parameters  $p_i$ . To perform this marginalization, you will need to exploit various conditional independence relations.

- (b) Consider estimating the noisy-OR parameters  $p_i$  to maximize the (conditional) likelihood of the observed data. The (normalized) log-likelihood in this case is given by:

$$\mathcal{L} = \frac{1}{T} \sum_{t=1}^T \log P(Y = y^{(t)} | X = \vec{x}^{(t)}),$$

where  $(\vec{x}^{(t)}, y^{(t)})$  is the  $t$ th joint observation of  $X$  and  $Y$ , and where for convenience we have divided the overall log-likelihood by the number of examples  $T$ . From your result in part (a), it follows that we can estimate the parameters  $p_i$  in either the original network or the extended one (since in both networks they would be maximizing the same equation for the log-likelihood).

Notice that in the extended network, we can view  $X$  and  $Y$  as observed nodes and  $Z$  as hidden nodes. Thus in this network, we can use the EM algorithm to estimate each parameter  $p_i$ , which simply defines one row of the “look-up” CPT for the node  $Z_i$ .

Compute the posterior probability that appears in the E-step of this EM algorithm. In particular, for joint observations  $x \in \{0, 1\}^n$  and  $y \in \{0, 1\}$ , use Bayes rule to show that:

$$P(Z_i = 1, X_i = 1 | X = x, Y = y) = \frac{yx_i p_i}{1 - \prod_j (1 - p_j)^{x_j}}$$

- (c) For the data set  $\{\vec{x}^{(t)}, y^{(t)}\}_{t=1}^T$ , show that the EM update for the parameters  $p_i$  is given by:

$$p_i \leftarrow \frac{1}{T_i} \sum_t P(Z_i = 1, X_i = 1 | X = x^{(t)}, Y = y^{(t)}),$$

where  $T_i$  is the number of examples in which  $X_i = 1$ . (You should derive this update as a special case of the general form presented in lecture.)

- (d) Download the data files on the course web site, and use the EM algorithm to estimate the parameters  $p_i$ . The data set<sup>1</sup> has  $T = 267$  examples over  $n = 23$  inputs. To check your solution, initialize all  $p_i = \frac{1}{n}$  and perform 256 iterations of the EM algorithm. At each iteration, compute the log-likelihood shown in part (b). (If you have implemented the EM algorithm correctly, this log-likelihood will always increase from one iteration to the next.) Also compute the number of mistakes  $M \leq T$  made by the model at each iteration; a mistake occurs either when  $y_t = 0$  and  $P(y_t = 1 | \vec{x}_t) \geq 0.5$  (indicating a false positive) or when  $y_t = 1$  and  $P(y_t = 1 | \vec{x}_t) \leq 0.5$  (indicating a false negative). The number of mistakes should generally decrease as the model is trained, though it is not guaranteed to do so at each iteration. Complete the following table:

iteration	number of mistakes $M$	log-likelihood $\mathcal{L}$
0	195	-1.04456
1	60	
2		-0.41076
4		
8		
16		
32		
64	37	
128		
256		-0.31016

You may use the already completed entries of this table to check your work.

- (e) Turn in your source code. As always, you may program in the language of your choice.

---

<sup>1</sup>For those interested, more information about this data set is available at <http://archive.ics.uci.edu/ml/datasets/SPECT+Heart>. However, be sure to use the data files provided on the course web site, as they have been specially assembled for this assignment.

---

## 6.4 Auxiliary function

In class we derived an auxiliary function for optimizing the log-likelihood in belief networks with hidden variables. In this problem you will derive an auxiliary function for optimizing a simpler function that is nearly quadratic near its minimum, but nearly linear far away from its minimum.

- (a) Consider the function  $f(x) = \log \cosh(x)$ . Show that the minimum occurs at  $x = 0$ .
  - (b) Show that  $f''(x) \leq 1$  for all  $x$ .
  - (c) Consider the function  $Q(x, y) = f(y) + f'(y)(x - y) + \frac{1}{2}(x - y)^2$ . Plot  $f(x)$ ,  $Q(x, -2)$ , and  $Q(x, 1)$  as a function of  $x$ .
  - (d) Prove that  $Q(x, y)$  is an auxiliary function for  $f(x)$ . In particular, show that it satisfies:
    - (i)  $Q(x, x) = f(x)$
    - (ii)  $Q(x, y) \geq f(x)$*Hint:* use part (b), and note that  $f(x) = f(y) + \int_y^x du f'(u) = f(y) + \int_y^x du \left[ f'(y) + \int_y^u dv f''(v) \right]$ .
  - (e) Derive the form of the update rule  $x_{n+1} = \operatorname{argmin}_x Q(x, x_n)$ .
  - (f) Write a simple program to show that your update rule in (e) converges numerically for the initial guesses  $x_0 = -2$  and  $x_0 = 1$ . **Turn in your source code as well as plots of  $x_n$  versus  $n$ .**
  - (g) Repeat parts (e) and (f) using the update rule for Newton's method: namely,  $x_{n+1} = x_n - f'(x_n)/f''(x_n)$ . What happens and why? Determine an upper bound on  $|x_0|$  so that Newton's method converges. (*Hint:* require  $|x_1| < |x_0|$ .)
  - (h) Plot the function  $g(x) = \frac{1}{10} \sum_{k=1}^{10} \log \cosh \left( x + \frac{1}{k} \right)$ . Is it still simple to find the exact minimum?
  - (i) Consider the function  $R(x, y) = g(y) + g'(y)(x - y) + \frac{1}{2}(x - y)^2$ . Prove that  $R(x, y)$  is an auxiliary function for  $g(x)$ .
  - (j) Derive the form of the update rule  $x_{n+1} = \operatorname{argmin}_x R(x, x_n)$ .
  - (k) Use the update rule from part (j) to locate the minimum of  $g(x)$  to four significant digits. In addition to your answer for the minimum, **turn in your source code as well as plots of  $x_n$  versus  $n$ .**
-