CS475 Machine Learning, Fall 2012: Homework 7

Daniel Crankshaw

Question 1:

Bayesian Network

(a) **A** and **B** are d-separated in this example. The set of paths going through x_5 is blocked by the tail to tail intersection at x_5, which is in set **C**. The set of paths going through x_{14} is blocked by the head to tail intersection at x_{14} because x_{14} is in **C**.

(b) No the sets are not d-separated.

(c) Yes. The sets are d-separated because every path between **A** and **B** must pass through x_{15} with a head to head intersection. And neither x_{15} nor any of its descendants are in **C**, so that node blocks.

(d) No. The sets are not d-separated. All the head to head intersections of the paths have x_{15} as a descendant (or occur at x_{15}) and so none of those nodes block. And some of the paths do not go through any other nodes in **C** to meet any of the tail to tail or head to tail blocking conditions.

Markov Random Field

(a) **A** and **B** are d-separated in this example because all paths must go through x_5 or $x14$ which are in **C** and thus blocked.

(b) Yes the sets are d-separated because all paths must go through x_{15} which is in set **C** and thus blocks all paths.

(c) No because for example the path $x4- > x6- > x11- > x15- > x12- > x8- > x5$ contains no nodes in **C** and is therefore unblocked.

(d) Yes the sets are d-separated because all paths must go through x_{15} which is in set **C** and thus blocks all paths.

Question 2: This question is essentially asking us for the Markov Blanket for node x_2. That is the set of nodes $\{ x_5 \}$.

Question 3: We can use the results from the textbook to write $P(X_i, X_j|X_b)$ where X_b is the vector X except for X_i and X_j. Using the notation of the book, let the vector $X_a = (X_i, X_j)^T$. Then, we can use equation 2.72 from the text to write the exponential term of the conditional probability as $-\frac{1}{2}X_a^T \Lambda_{aa} X_a$. Now we substitue back in X_i and X_j, and let

$$\Lambda_{aa} = \begin{pmatrix} p & 0 \\ 0 & q \end{pmatrix} \tag{1}$$

because from the problem statement we know that Ω_{ij} (which is equivalent to our Λ) is 0. Therefore,

$$\begin{aligned} P(X_i, X_j|X_b) &= k \exp -\frac{1}{2}(X_i, X_j)\Lambda_{aa}(X_i, X_j)^T \qquad \text{where k is a normalization constant} && (2)\\ &= k \exp -\frac{1}{2}(pX_i^2 + qX_j^2) && (3)\\ &= k \exp -\frac{1}{2}pX_i^2 \exp -\frac{1}{2}qX_j^2 && (4)\\ &= P(X_i|X_b)P(X_j|X_b) && (5) \end{aligned}$$

This proves that if $\Omega_{ij} = 0$ then X_i and X_j are conditionally independent.

Question 4: The data likelihood is given as

$$P(Y_i, X_i) = \prod_{i=1}^{n} P(Y_i, X_i) = \prod_{i=1}^{n}\prod_{j=1}^{m} P(X_{ij}|Y_i)P(Y_i), \tag{6}$$

However, if we don't know Y, we can marginalize over Y to get

$$P(X_i) = \sum_{i=0}^{N}\sum_{y}\prod_{a=1}^{d}\prod_{j=1}^{m} P(X_{dj}|y)P(y), \tag{7}$$

From there we can calculate the log likelihood as a function of our parameters $P(X_{ij}|y)$ and $P(y)$ as

$$L(\theta) = \sum_{i=1}^{N} log\left(\sum_{j=1}^{K} P(Y_j)\prod_{a=1}^{D} P(X_i^a|Y_j)\right) \tag{8}$$

where N is the number of examples, K is the number of possible labels, and D is the number of features. The thought is that for the combination of supervised and unsupervised learning, we can just treat it as fully unsupervised learning for EM, and just fix the probabilities where we have labels. This would mean that for the E step, we would take derivatives of $L(\Theta)$ with respect to each of our parameters and set that equal to 0 and the solve for the parameter. However, I'm not totally sure how to solve those equations. For the M step, we take the updated parameters and relabel the data.

Question 5: No the Max Product algorithm cannot be applied because the factor graph has cycles. From the Viterbi algorithm we have equation 13.68

$$\omega(z_{n+1}) = \log p(x_{n+1}|z_{n+1}) + max_{z_n}\{\log p(z_{n+1}|z_n) + \omega(z_n)\} \tag{9}$$

The relevant part of this equation is the max function, which is where the recursion in this algorithm appears. We can ignore the x's for now and we say that the z nodes in our equation correspond to the x states in the graph. Notice that when evaluating x_8 in the graph, it will have to wait for $\omega(x_7)$ (where ω is the text's shorthand for the message $\mu_{f_n \leftarrow z_n}(z_n)$). But similarly, because of the cycle, when evaluating x_7, it will have to wait for $\omega(x_8)$. This will lead our algorithm into a sort of deadlock, where each node will be waiting on the other to finish it's message.

One algorithm that can be used to compute the maximal assignment is Loopy Belief Propagation, an approximate algorithm. This essentially describes a message passing schedule such that a new message sent across a link (in the same direction) replaces all previous messages sent on that link. We then initiate each link to have some initial message that has already been sent, so that at the start of the algorithm every node can send a message, alleviating our deadlock problem. We then just set up a schedule for when nodes should send messages (some examples of these are described in the text on p.417-18).