CS 726: Advanced Machine Learning, Fall 2020, Mid-Semester exam

February 24, 2020.
6:30 – 8:30 pm

Roll: _____

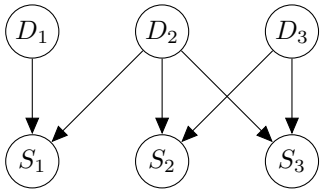
Name: _____

Mode: Credit/Audit/Sit-through _____

Write all your answers in the space provided. Do not spend time/space giving irrelevant details or details not asked for. Use the marks as a guideline for the amount of time you should spend on a question. You are allowed to write elsewhere only under special circumstances like total cancellation of a previously written answer. Use the last sheet of this booklet in such cases. You are only allowed to refer your notes, no one else's notes or textbook.

---

1. Consider a QMR Bayesian network comprising of disease nodes $D_1, \ldots, D_n$ and symptom nodes $S_1, \ldots S_k$. Each symptom node has an arbitrary subset of disease nodes as parents. There are no other edges. An example appears below.



   (a) State which of the following CIs will hold, with a brief justification. No marks without correct justification. You can use D-separation test or local-CI as justification.

      i. $D_i \perp\!\!\!\perp D_j$ ..1 yes. By D-separation either they will be disconnected or will have a V-node between them in all paths.

      ii. $S_i \perp\!\!\!\perp S_j | D_{pa(S_i)}$ ..1 True since S_j has to be non-descendant of S_i

      iii. $S_i \perp\!\!\!\perp S_j | D_1 \ldots D_n$ ..1 Yes since none of the D_is are descendants of any S node, and all paths from S_i to S_j are through one of the disease nodes.

      iv. $D_i \perp\!\!\!\perp D_j | S_k$ where S_k is not a child of D_i or D_j. ..1 True. Any path from D_i to D_j will have a V-node between them. If that V-node is S_k then there must be some other V-node that will block the path.

      v. $S_i \perp\!\!\!\perp S_j$ where S_i and S_j do not have a common parent. ..1 Yes. since then either they are disconnected or all paths between them will go through another symptom node that is a V-node.

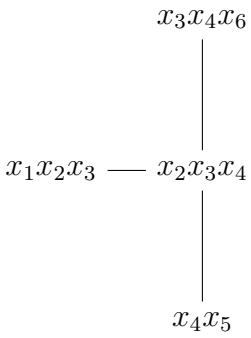
   (b) Show an example QMR bayesian network where moralizing by adding an undirected edge between all disease pairs sharing a symptom leads to a non-chordal graph. You need to ignore the edge directions when checking the cycle. [Hint: one solution exists at $n = K = 4$.] ..2 Let $n = 4, k = 4$ connect D_1 to S_1 and S_4 and each other D_i to S_{i-1} and S_i. We have a cycle of length 4 over the disease nodes.

2. In a Bayesian network the potentials represent conditional probabilities. This eliminates the need for certain messages when we run sum-product message passing inference to compute single-variable marginals. For example, the marginal of a variable without any parents are already available as potentials and do not need to be computed via inference.

(a) In a single chain Bayesian network $x_1 \to x_2 \to \ldots x_n$ with only $x_{i-1} \to x_i$ edges, state the list of messages that do not need to be computed if our goal is to compute $P(x_i)$ for $i = 1 \ldots n$. ..2 The backward messages from $x_i x_{i+1}$ clique to $x_i x_{i-1}$ clique are redundant and can be shown to be one.

(b) Now consider a general Bayesian network. For each variable x_i in a Bayesian network G, what are the variables in G whose potentials are not required to compute $P(x_i)$? ..3 Only ancestors of x_i are required. All others can be pruned.

(c) In any junction tree, let C_i, C_j be two cliques and S_{ij} be the separator between them. Let V_j denote the variables in the C_j side of the tree away from C_i excluding S_{ij}. Likewise, define V_j. For example, in the junction tree below, if $C_i = x_1x_2x_3$ and $C_j = x_2x_3x_4$, then $V_j = x_4x_5x_6$, $V_i = x_1$, and $S_{ij} = x_2x_3$.



Show that the variables in V_i are independent of V_j given S_{ij}, that is $V_i \perp\!\!\!\perp V_j | S_{ij}$. The proof needs to be for a general JT, not just the above example. ..3 The potentials in the graphical model either belong $V_i \cup S_{ij}$ or $V_j \cup S_{ij}$. Thus $P(V_i, V_j, S_{ij}) = g(V_i \cup S_{ij})h(g(V_j \cup S_{ij})$ This implies our result.

(d) For general Bayesian networks on which we have drawn a junction tree provide a simple test you can run to determine which set of messages are not required to be sent. [Of course, you need to determine these using only the graph structure and without actually computing those messages.] ..2 For each clique C_i is no variable in V_j contains an ancestor of C_i, we do not need the message from C_j to C_i. This is because

3. Assume we are trying to learn parameters of a simple chain of three variables represented as an undirected graphical model y_1—z—y_2. Assume all variables are binary, and the potential $\psi_1(y_1, z) = \psi_2(z, y_2)$ is expressed as a table with four parameters: $\left[\begin{smallmatrix} \theta_{00} & \theta_{01} \\ \theta_{10} & \theta_{11} \end{smallmatrix}\right]$ which we jointly call θ. For example, $P(y_1 = 1, z = 0, y_2 = 1) = \frac{\psi_1(1,0)\psi_2(01)}{Z} = \frac{\theta_{1,0}\theta_{01}}{Z}$ In training data only y_1, y_2 variables are observed in each example, and denoted as $(y_1^1, y_2^1), \ldots, (y_1^N, y_2^N)$. Let us denote the values of the parameter at time t as $\theta^t = \left[\begin{smallmatrix} 2 & 3 \\ 4 & 5 \end{smallmatrix}\right]$. Fill in the following answers:

(a) Calculate the value of normalizer Z for the above graphical model at θ^t using sum-product inference. ..2
$Z = \sum_{y_1 z y_2} \theta_{y_1,z}\theta_{z,y_2} = \sum_{y_1 z} \theta_{y_1,z} \sum_{y_2} \theta_{z,y_2} = \sum_{y_1 z} \theta_{y_1,z}[5, 9]$
$= \sum_{y_1 z} \left[\begin{smallmatrix} 2*5 & 3*9 \\ 4*5 & 5*9 \end{smallmatrix}\right] = 2*5 + 3*9 + 4*5 + 5*9$

(b) The values calculated in the E-step for the ith instance for which $(y_1^i, y_2^i) = (0, 1)$ [Give your answer as absolute numbers, but show the steps.] ..2
$P(z = 0 | \mathbf{y}^i, \theta^t) = \theta_{00}\theta_{01}/(\theta_{00}\theta_{01} + \theta_{01}\theta_{11}) = 2*3/(2*3 + 3*5)$

(c) If in the training data N_{uv} denotes the number of instances for which the $y_1 = u$ and $y_2 = v$ for binary values u, v. Let q_{uv}^t denote $P(z = 1 | y_1 = u, y_2 = v, \theta^t)$. Write the

M-step objective purely in terms of N_{uv} and q^t_{uv}. ..3
$P(z=1|y_1=u,y_2=v) = \theta_{u1}\theta_{1v}/(\theta_{u1}\theta_{1v}+\theta_{u0}\theta_{0v}) = q^t_{uv}$
The M-step is then $\sum_{uv} n_{uv} q^t_{uv} \log(\theta_{u1}\theta_{1v}) + (1-q^t_{uv})\log(\theta_{u0}\theta_{0v}) - N\log(\sum_{uv}\theta_{u1}\theta_{1v} + \theta_{u0}\theta_{0v})$

(d) Assume $N_{11} = N$ and all other $N_{uv} = 0$. For what value of θ is the data likelihood globally maximized? If there are multiple optimal solutions state those too. [Justify briefly] ..2 $\theta_{11} = 1$ all other theta-s 0.

4. Consider training a CRF $P(\mathbf{y}|\mathbf{x},\theta) = \frac{\exp(\sum_c F_\theta(\mathbf{y}_c,c,\mathbf{x})}{Z_\theta(\mathbf{x})}$ where the potentials $F_\theta(\mathbf{y}_c,c,\mathbf{x})$ are computed using a neural network (NN). The neural network takes $\mathbf{y}_c, c, \mathbf{x}$ as input, all its parameters are jointly called θ, and outputs a real score which we call $F_\theta(\mathbf{y}_c,c,\mathbf{x})$. Fiven training sample $D = \{(\mathbf{x}^i,\mathbf{y}^i) : i = 1,\ldots,N\}$, we will train the parameters θ of NN by maximizing following likelihood.

$$LL(\theta, D) = \sum_{i=1}^{N} \log \Pr(\mathbf{y}^i|\mathbf{x}^i,\theta) = \sum_i \sum_c F_\theta(\mathbf{y}^i_c, c, \mathbf{x}^i) - \log Z_\theta(\mathbf{x}^i)$$

where an undirected graph H defines the cliques c

(a) If we use gradient ascent training, write down the gradient of the above at θ^t. Your answer should be in terms of appropriate marginals and gradients of F_θ. ..3

$$\nabla LL(\boldsymbol{\theta}) = \sum_i \sum_c \nabla F_\theta(\mathbf{y}^i_c, c, \mathbf{x}^i) - \sum_i \sum_c \sum_{\mathbf{y}_c} \mu^t(\mathbf{y}_c, c, \mathbf{x}^i) \nabla F_\theta(\mathbf{y}^i_c, c, \mathbf{x}^i)$$

(b) If an instance $\mathbf{x}^i$ has C cliques, each of size 2 and each y_j takes m possible values, how many times do we need to backprop on the NN to calculate the above gradient for that instance $\mathbf{x}^i$. ..1 m^2C

(c) What trick can you suggest to reduce the number backprop steps down to a constant (like 1 or 2) per clique? ..2 Feed only $\mathbf{x}^i, c$ as input, and a softmax layer on top to capture dependence on $\mathbf{y}_c$.

5. In class we showed a construction for proving that marginal inference is NP-hard on a Bayesian network by reduction from 3-SAT over n literals and k clauses.

- $C_j = l_{j1} \vee l_{j2} \vee l_{j3}$
- $l_{jp} = x_i$ or $\bar{x_i}$ for some $i \in [1,n], p = 1,2,3$
- $S_j = S_{j-1} \wedge C_j$ for $j = 2 \ldots k$ and $S_1 = C_1$

Now suppose our goal is to count the number of satisfying assignments in 3-SAT problem. In other words out of all 2^n possible assignments of the $x_1,\ldots,x_n$ variables, we wish to find the number of those for which $S_k = 1$. For example, when we have only one clause $C_1 = x_1 \vee x_2 \vee x_3$ and $n = 3$ we know that $2^3 - 1 = 7$ assignments of x_i variables satisfy C_1. Show how you can find such an answers using the result of a graphical model inference. ..3 Run sum-product inference on the Bayesian network discussed in class with $P(x_i) = [0.5, 0.5]$. $2^n P(S_k = 1)$ gives the number of satisfying assignments.

Total: 35