10708 Graphical Models: Homework 3

Due October 29th, beginning of class

October 15, 2006

Instructions: There are six questions on this assignment. Each question has the name of one of the TAs beside it, to whom you should direct any inquiries regarding the question. Please submit your homework in two parts, one for each TA. Also, please put the TA's name on top of the homework.

Note: Starting this homework, you *will* be penalized points for not splitting the homework.

The last problem involves coding, which should be done in MATLAB. Do *not* attach your code to the writeup. Instead, copy your implementation to

`/afs/andrew.cmu.edu/course/10/708/Submit/your_andrew_id/HW3`

Refer to the web page for policies regarding collaboration, due dates, and extensions.

1 Variable Elimination **Dhruv [10 pts]**

Your friendly neighbourhood TA, like most other TAs, is intent on world domination. His first step, obviously, was to build a graphical model, as shown in Figure 1. The variables being: Graduate (G), Free Food (FF), The Force (TF), Knowledge (K), Money (M), Power (R) and World Domination (WD). All the variables are binary valued $\{T, F\}$. The CPT parameters are:

$$P(G = T|TF = T, FF = T) = 0.9, \quad P(G = T|TF = T, FF = F) = 0.7$$
$$P(G = T|TF = F, FF = T) = 0.5, \quad P(G = T|TF = F, FF = F) = 0.1$$

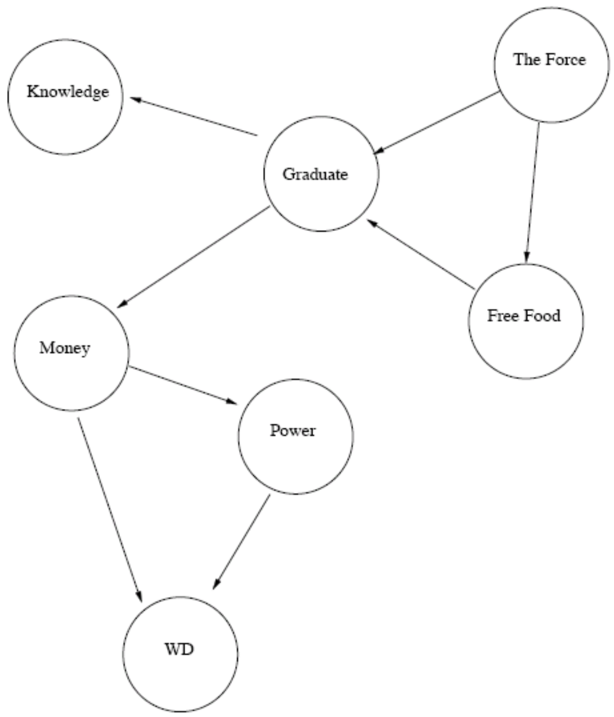

Figure 1: World Domination network

$$P(FF = T|TF = T) = 0.8, \quad P(FF = T|TF = F) = 0.6$$
$$P(TF = T) = 0.1$$
$$P(K = T|G = T) = 0.7, \quad P(K = T|G = F) = 0.6$$
$$P(R = T|M = T) = 0.7, \quad P(R = T|M = F) = 0.1$$
$$P(M = T|G = T) = 0.6, \quad P(M = T|G = F) = 0.5$$

$$P(WD = T|M = T, R = T) = 0.7, \quad P(WD = T|M = T, R = F) = 0.5$$
$$P(WD = T|M = F, R = T) = 0.6, \quad P(WD = T|M = F, R = F) = 0.05$$

Help your friendly TA make some urgent inferences from his world domination network; but make sure you're not baited by his nemesis: Exponential Computational Complexity.

1. How likely is the TA to take over the world, if he manages to graduate? $P(WD = T|G = T) = ?$
2. $P(WD = T|FF = T) = ?$
3. Should we even be worried about him graduating? $P(G = T) = ?$
4. $P(M = T|K = T) = ?$

Additionally, report the ordering used and the factors produced after eliminating each variable for the first query $[P(WD = T|G = T)]$.

2 Conditional Probabilities in Variable Elimination Amr [13pts]

Consider a factor produced as a product of some of the CPDs in a Bayesian network $\mathcal{B}$:

$$\tau(\mathbf{W}) = \prod_{i=1}^{k} P(Y_i|\mathbf{Pa}_{Y_i})$$

where $\mathbf{W} = \cup_{i=1}^{k}(\{Y_i\} \cup \mathbf{Pa}_{Y_i})$.

1. Show that τ is a conditional probability *in some network*. *Hint*: Partition $\mathbf{W}$ into two disjoint sets, $\mathbf{Y}$ and $\mathbf{Z}$, i.e., $\mathbf{W} = \mathbf{Y} \cup \mathbf{Z}$, and $\mathbf{Y} \cap \mathbf{Z} = \emptyset$. Then show that $\tau(\mathbf{W}) = P(\mathbf{Y}|\mathbf{Z})$. See also **KF, Section 8.3.1.3**
2. Show that the intermediate factors produced by the variable elimination algorithm are also conditional probabilities *in some network*.

3 Triangulation Dhruv [7 pts]

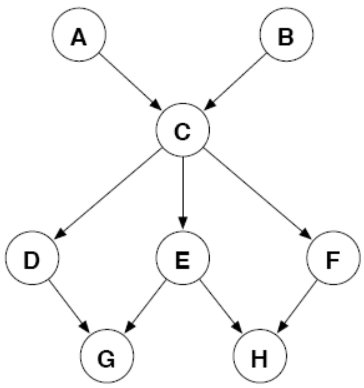

Figure 2: Bayes net for question 3

1. Moralize the Bayes net in figure 2.
2. Supply a perfect elimination ordering (*i.e.*, one that yields no fill edges).
3. Supply an elimination ordering that yields a triangulated graph with at least 5 nodes in one or more cliques

4. Draw clique trees for the elimination orderings in parts 2 and 3.

4 Clique Tree Representation **Dhruv [20 pts]**

In this question you will formalize the relationship between clique trees for a Bayesian network and the probability distribution the Bayes network encodes. In summary, if P factorizes according to a Bayesian Network, then any clique tree $\mathcal{T}$ for this BN is an I-map for P, moreover, P also factorizes according to $\mathcal{T}$ in a way that we will make explicit below.

1. [Clique tree I-map] In a clique tree, consider a separator $\mathbf{S}_{ij}$ between two cliques $\mathbf{C}_i$ and $\mathbf{C}_j$. Let $\mathbf{X}$ be any set of variables in the $\mathbf{C}_i$ side of the tree, and $\mathbf{Y}$ be any set of variables in the $\mathbf{C}_j$ side of the tree. **Prove** that $P \models (\mathbf{X} \perp \mathbf{Y} \mid \mathbf{S}_{ij})$.
   (Hint: Consider using an independence property we derived in HW1)

2. [Clique tree factorization] Using the independencies above, **prove** that in a clique tree for a BN, *when the clique tree is calibrated*, we can represent the joint distribution by:

   $$P(\mathbf{X}) = \frac{\prod_i P(\mathbf{C}_i)}{\prod_{ij} P(\mathbf{S}_{ij})}.$$

   You should not "prove" by corollary from the correctness of BP in clique trees.
   (Hint: combine the chain rule of probabilities with the definition of conditional probabilities.)

5 Variable Elimination in Clique Trees **Dhruv [20 pts]**

Consider a chain graphical model with the structure $X_1 - X_2 - \cdots - X_n$, where each X_i takes on one of d possible assignments. You can form the following clique tree for this GM: $\mathbf{C}_1 - \mathbf{C}_2 - \cdots - \mathbf{C}_{n-1}$, where $Scope[\mathbf{C}_i] = \{X_i, X_i + 1\}$. You can assume that this clique tree has already been calibrated. Using this clique tree, we can directly obtain $P(X_i, X_i + 1)$. As promised in class, your goal in this question is to compute $P(X_i, X_j)$, for any $j > i$.

1. Briefly, describe how variable elimination can be used to compute $P(X_i, X_j)$, for some $j > i$, in linear time, given the calibrated clique tree.

2. What is the running time of the algorithm in part 5.1 if you wanted to compute $P(X_i, X_j)$ for all n choose 2 choices of i and j?

3. Consider a particular chain $X_1 - X_2 - X_3 - X_4$. Show that by caching $P(X_1, X_3)$, you can compute $P(X_1, X_4)$ more efficiently than directly applying variable elimination as described in part 5.1.

4. Using the intuition in part 5.3, design a dynamic programming algorithm (caching partial results) which computes $P(X_i, X_j)$ for all n choose 2 choices of i and j in time asymptotically much lower than the complexity you described in part 5.2. What is the asymptotic running time of your algorithm?

6 Variable Elimination **Amr [30 pts]**

6.1 [15 pts]

Implement the variable elimination algorithm from class. You **shall not** implement pruning for inactive variables but we **require** that you implement the min-fill heuristic in order to select an elimination order. However, we will **NOT** require that your min-fill implementation be *query-specific* , *i.e.*, you should first apply the min-fill heuristic once on the whole graph to get an elimination order which will be then used in answering all the queries [1].

You can reuse any code you wrote for hw2, and you are free to use the the posted solution for hw2.

Important: Submit your implementation to your AFS code directory, and in addition submit a script called ***"run.m"*** that when invoked reproduces all the output in parts 6.1 and 6.2[2]. Moreover, answer the following questions in your writeup, reporting all probabilities to *four significant digits.*

1 Using the network in Figure 1, what is the elimination order produced by your min-fill implementation? And how many fill-edges added?

2 Using the Alarm network in `alarm.m` compute the value of the following queries.

   (a) How many fill-edges added to this network by your min-fill implementation?

   (b) P(StrokeVolume = High | Hypovolemia = True, ErrCauter = True, PVSat = Normal, Disconnect = True, MinVolSet = Low)

   (c) P(HRBP = Normal | LVEDVolume = Normal, Anaphylaxis = False, Press = Zero, VentTube = Zero, BP = High)

   (d) P(LVFailure = False | Hypovolemia = True, MinVolSet = Low, VentLung = Normal, BP = Normal)

   (e) P(PVSAT = Normal ,CVP = Normal | LVEDVolume = High, Anaphylaxis = False, Press = Zero)

[1] In practice, as we discussed in class, taking the evidence variables into consideration will result in a better elimination order, however, for simplicity we will ignore this point.

[2] Yes, we are aware that for 6.2 the results depend on the machine used to run your code, but that is fine.

6.2 [5 pts]

Another naive way of answering a conditional probability query of the form $P(X = x|Y = y)$ is as follows:

$$P(X = x|Y = y) = \frac{P(X = x, Y = y)}{P(Y = y)} = \frac{\sum_z P(X = x, Y = y, Z = z)}{\sum_{x',z} P(X = x', Y = y, Z = z)} \tag{1}$$

where X and Y are two subsets of the variables in the network, and $Z = \text{Var(Network)} - X \cup Y$, i.e. all other variables in the network other than X and Y. Each of the terms in the above summations can be evaluated by simple multiplications of the factors in the network as you were asked to implement in HW2. We would like to compare the time consumed by the variable elimination algorithm you implemented and the above naive approach. However, as you will be convinced below, you should NOT attempt to run the naive approach on the *alarm* network. Instead, we will estimate its running time as follows:

1. Compute the time needed to evaluate each of the terms in the summations above — that is the time needed to execute a call to *assignProb*. You can do that by simply running *assignProb* once using any random assignment to the variables in the network. **Write** down this time in your submission.

2. Compute the number of calls to *assignProb* needed to evaluate Eq.(1). **Write** down a formula for this number of calls using the dimensionality of the variables in X, Y and Z.

3. Estimate the time needed by the naive approach based on the above two quantities[3].

Using the above procedure, compare the running time of both variable elimination and the naive approach over each of the queries in part 6.1.2

6.3 [10 pts]

Create a naïve Bayes network on binary variables $\mathcal{V} = \{C, X_1, \ldots, X_k\}$ where C is the class and $X_1, X_2, \ldots, X_k$ are the features. Choose some parameterization of the network such that each parameter $0 < \theta_{x_i|\mathbf{u}_i} < 1$. In the context of variable elimination

1 What ordering on $\mathcal{V}$ has minimum induced treewidth (call it $\prec_o$) ?

2 What ordering on $\mathcal{V}$ has maximum induced treewidth (call it $\prec_w$) ?

Using your implementation of variable elimination

3 For $k = 1 \ldots 10$ compute $\sum_{\mathcal{V}} P(C, X_1, \ldots, X_k)$ using $\prec_o$ and $\prec_w$. Plot the running time of each method vs. k.

[3]We will ignore the time needed to sum the terms in the summations in (1)

(*Note*: Yes, we know that the quantity you are computing is 1.0. The point of the question is to compare running times.)