Homework 2
Statistical methods in AI/ML

Instructor: Vibhav Gogate

[Problems marked with ** are time consuming. Start early.]

Problem 1:[55 points]**

Write a C++ (or Python or Java) program for performing variable elimination. The program should take as input two files: (a) a Markov network in UAI format and (b) Evidence (namely an assignment of values to some subset of variables) in UAI format and output the partition function given evidence. Your program should eliminate the variables along the min-degree order (ties broken randomly). The UAI format is described here:

`http://www.hlt.utdallas.edu/~vgogate/uai16-evaluation/uaiformat.html`

Recall that the variable elimination algorithm has three steps:

1. Instantiate Evidence (Reduce the CPTs or factors).
2. Order the variables. (You are using the min-degree order for the purpose of this assignment)
3. Eliminate variables one by one along the order. Recall that to eliminate a variable X, we compute a product of all functions that mention X. Let us call the new function f. Then, we sum-out the variable from f to yield a new function f'. Then we replace all functions that mention X with f'.

Thus, following the divide and conquer approach to programming, you can first develop the following helper functions and then put them together into a variable elimination algorithm.

(Note that either you can use the following approach or develop your own. The following might be easier).

1. Function `Read`: Create a class called GraphicalModel and create a object of this class from the given UAI file
2. Function `Order`: Compute a min-degree ordering over the non-evidence variables (note that you only have to eliminate only the non-evidence variables if you instantiate the evidence properly).

3. Function `Instantiate`: Take a factor ϕ and evidence as input and instantiate evidence in the factor.

4. Function `Product`: Take two factors ϕ_1 and ϕ_2 as input and output $\phi_3 = \phi_1 \times \phi_2$.

5. Function `Sum-out`: Take a factor ϕ and a set of variables $\mathbf{Y}$ as input and output $\phi_1 = \sum_{\mathbf{Y}} \phi$.

To efficiently implement the operations describe above, see Box 10.A, pages 358-361 in Koller and Friedman.

How to test your code?. A number of networks along with their correct probability of evidence or partition function values are available on the course website.

What to turn in? Source code and a README file on how to compile/execute your code. Submit using E-learning. Please don't email me your code.

Problem 2: [5 points] Do Exercise 6.5. from AD

Problem 3: [5 points] Do Exercise 6.9 from AD

Problem 4: [5 points] Do Exercise 6.10 from AD

Problem 5: [15 points] Consider the Bayesian network given in Figure 1.

- Convert this Bayesian network into an equivalent Markov network. Convert the resulting Markov network into an equivalent Bayesian network.

- Let H be evidence variables. Trace the operations of Bucket elimination for computing $\Pr(H = h)$ along the order (A, E, B, C, D, F, G).

- Is the ordering (A, E, B, C, D, F, G) optimal? What is the treewidth of this network (assume that H is an evidence variable and so the resulting network does not contain H)?

- Construct a tree decomposition for this network (again assume that H is an evidence variable). Show how the junction tree propagation algorithm will operate on this tree decomposition. Show the expression for each message.

Problem 6: (15 points) Exercise 5.5 from Koller and Friedman

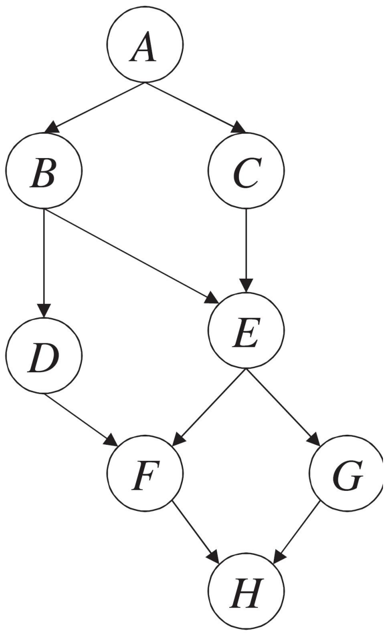

Figure 1: A Bayesian network