CS 578 (Fall 2015) – Bareinboim
Homework 2

Due: Oct 28, 2015

Problem 1 (50 points)

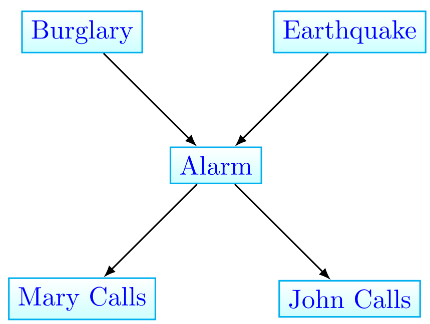
In this problem, you will implement the belief network algorithm. Your solution should include all code and instructions on how to run your algorithm[1]. Please note that you are expected to automatically transform the given graphs into forms suitable for computation, and to correctly handle 0 values.

In addition to Koller's book, you may be interested in checking these (historic) references:

1. Reverend Bayes on Inference Engines: A Distributed Hierarchical Approach, Judea Pearl, 1982. https://www.aaai.org/Papers/AAAI/1982/AAAI82-032.pdf

2. A Computational Model for Causal and Diagnostic Reasoning in Inference Systems, Jin Kim and Judea Pearl, 1983 http://ijcai.org/Past%20Proceedings/IJCAI-83-VOL-1/PDF/041.pdf

Part 1. How likely is burglary?



The probability tables for this network are:

$$P(B) = 0.001, P(E) = 0.002$$

A	$P(J\|A)$	$P(M\|A)$
T	0.9	0.7
F	0.05	0.01

B	E	$P(A\|B,E)$
T	T	0.95
T	F	0.94
F	T	0.29
F	F	0.001

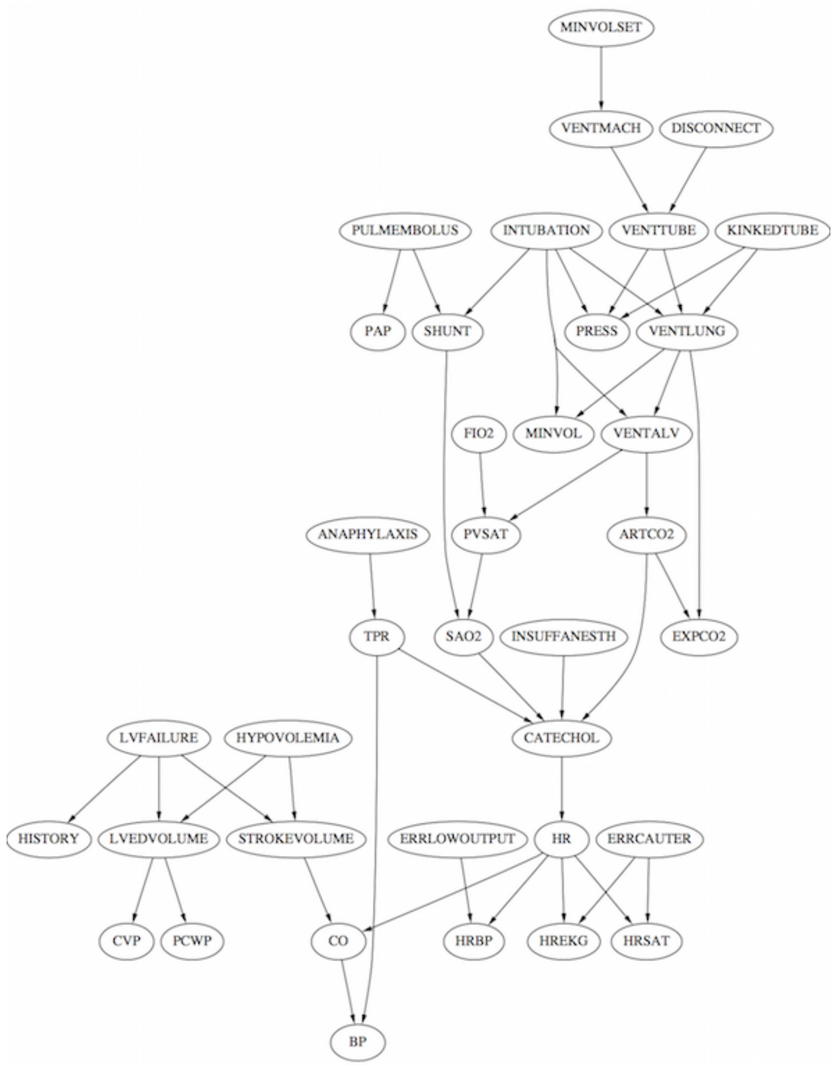
Given that John called, generate a table of the probabilities of all configurations of the variables (your table should have 2^4 elements). Generate this table using both brute force (enumerating all the possibilities) and the sum product algorithm.

[1]You are free to use any language you want so long as a compiler/interpreter as well as any packages you use are freely available on the internet. If you are not sure what to use, Python is HIGHLY recommended (it will make your life very easy compared to something like Java). Not to mention that the TA can help you with python.

Part 2. The ALARM Network

The 1989 ALARM network is a well-known example of bayesian networks used in practice. The network, which can be found at http://compbio.cs.huji.ac.il/Repository/Datasets/alarm/alarm.htm was used to monitor patients in intensive care. The link contains the network in multiple file types. Using your favorite programming language, write a program that reads in the network from one of the file types[2], and use your implementation of belief propagation to answer the questions below. Run your algorithm for 500 iterations or until messages change less than 10^{-6} each iteration.

Compute the query for the probability of $KinkedTube$ given $SaO2 = NORMAL$, $BP = NORMAL$, $ArtCO2 = NORMAL$, $Press = NORMAL$ and $ExpCO2 = LOW$.



Part 3. Convergence of Sum-Product

Under what conditions will belief propagation converge on the true marginals? Given a node N in a graph G, as well as a set Z of given nodes, give an algorithm that will return "yes" if belief propagation should converge to the true value, and "no" if it will not. Explain. (No need to implement it, and no need for proofs. We just want an intuition on why your algorithm is correct.)

[2]You are free to use any helping libraries for parsing the file into a form suitable for computation. If you have any issues with this part of the problem, the TA will be able to help out in office hours.

Problem 2 (50 points)

Consider the causal model:

$$x_{k+1} = XOR(OR(x_k, y_k), e_k)$$
$$y_{k+1} = XOR(OR(x_k, y_k), e'_k)$$

where:

- X_k, Y_k, e_k, e'_k are binary variables with $k = 0, 1, 2, 3, 4, 5$
- e_k, e'_k are independent Bernoulli variables taking the value 1 with probability p.
- X_0, Y_0 are independent Bernoulli variables with $P(X_0 = 1) = P(Y_0 = 1) = \frac{1}{4}$

Using this model, generate n random samples, where each sample is 6 pairs:

$$\{X_k = x, Y_k = y_k\}, k = 0, 1, 2, 3, 4, 5$$

1. Implement and apply the IC algorithm to learn the structure of the generating model, and give the resulting structure for:

$$n = 100, 1000$$
$$p = 0, 0.05, 0.2$$

2. Repeat using the IC* algorithm
3. For the case $n = 1000, p = 0.2$ explore and explain the reasons that the structures discovered by your algorithm differ from those of the generating model.
4. Repeat 3 for $n = 100, p = 0$