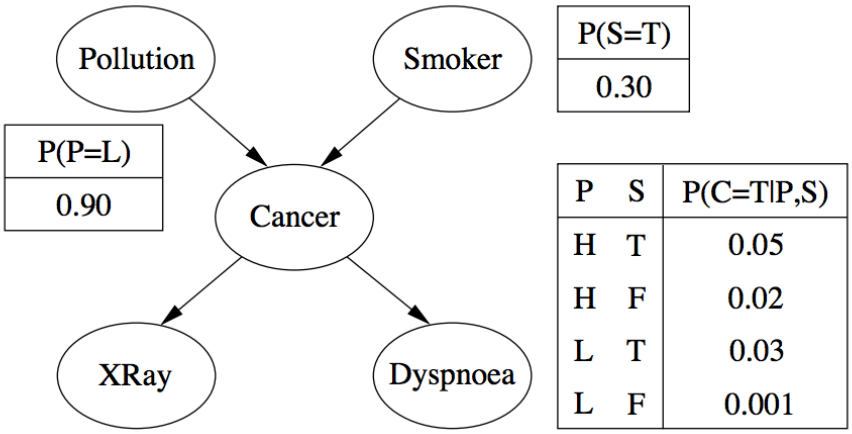
CSCI 3202 Introduction to Artificial Intelligence
Instructor: Hoenigman
Assignment 6
Due Friday, October 23, by 4pm.

Github
In your *ai3202* repository, create a folder called *Assignment6*, and place all code for this assignment in that folder. We will be grading your assignment by pulling your code from there.

Bayes Net Disease Predictor

For this assignment, you will implement a Bayes Net to predict the probability of cancer given several disease variables and explore how changes in prior probabilities for Pollution and Smoking change the probability of cancer. The network structure and probability distributions are taken from the Bayes Net Tutorial document posted on Moodle, and are as follows:



P(S=T)
0.30

P(P=L)
0.90

P	S	P(C=T|P,S)
H	T	0.05
H	F	0.02
L	T	0.03
L	F	0.001

C	P(X=pos|C)
T	0.90
F	0.20

C	P(D=T|C)
T	0.65
F	0.30

The variable types are:

Node name	Type	Values
Pollution	Binary	{*low, high*}
Smoker	Boolean	{*T, F*}
Cancer	Boolean	{*T, F*}
Dyspnoea	Boolean	{*T, F*}
X-ray	Binary	{*pos, neg*}

What your program needs to do:

Handle user queries from the command line
Your program needs to handle queries from the user on the joint, marginal, and conditional probabilities for any variables in the network.

Queries will be generated from the command line using options and arguments. The easiest way to do this in Python is to use the getopt module that reproduces the Unix getopt functionality. More information is available here:
https://docs.python.org/2/library/getopt.html

The option flags in your program need to be:
-g conditional probability (Yes, that's a g.)
-j joint probability
-m marginal probability
-p to set a prior for either Pollution or Smoking

Following each of the options will be the variable abbreviations to include in the calculation. Use the following abbreviations:

P = Pollution
S = Smoker
C = Cancer
D = Dyspnoea
X = X-ray

Use the lowercase letter for Variable = True, or Variable = low in the case of Pollution.

Use ~ before the lower case letter to indicate Variable = False, or Variable = high in the case of Pollution.

Use the capital letter to return the probability distribution for the variable.

For the -p option of setting a prior, use P or S, followed by the new numerical value.

You can assume that your program only needs to handle one calculation each time it is run. The only exception is when you are setting different priors for smoking and pollution and then also performing the calculations.

For example,
-mD is the marginal probability distribution of Dyspnoea

-jPSC is the joint probabilities for Pollution, Smoker, and Cancer (Yes, this will return a lot of information.)

-jpsc is the joint probability for pollution = low, smoker = true, cancer = true

-j~p~s~c is the joint probability for pollution = high, smoker = false, cancer = false

-gc|s is the conditional probability for Cancer given that someone is a smoker.
-pS .40 sets the probability that smoking is True to .40.

-pP = .80 sets the probability that pollution is Low to .80.

Implement the Bayes Net
Your program needs to handle the user queries by querying the Bayes net. For this to work, you need to implement the Bayes net. Each node in your network needs to store the conditional probabilities for that node given the parents. One suggestion is to use a dictionary for each node in the Bayes net. You might also be able to build upon an early assignment using a graph library.

All probabilities other than the conditional probability tables should be calculated upon user request, not stored in the nodes themselves.

Before you begin coding on this assignment, it will probably be useful to work out all calculations on paper and plan your implementation, including the data structure you will use for the Bayes Net and the functions you will need for your calculations.

Grading
Your code needs to handle all of the calculations given in Table 2.2 of the Bayesian Networks tutorial posted on Moodle. The correct numbers are given for P(s) = .3 and P(s) = .5. Use those values to verify that your calculations are correct before changing those priors and using the new priors in other calculations.