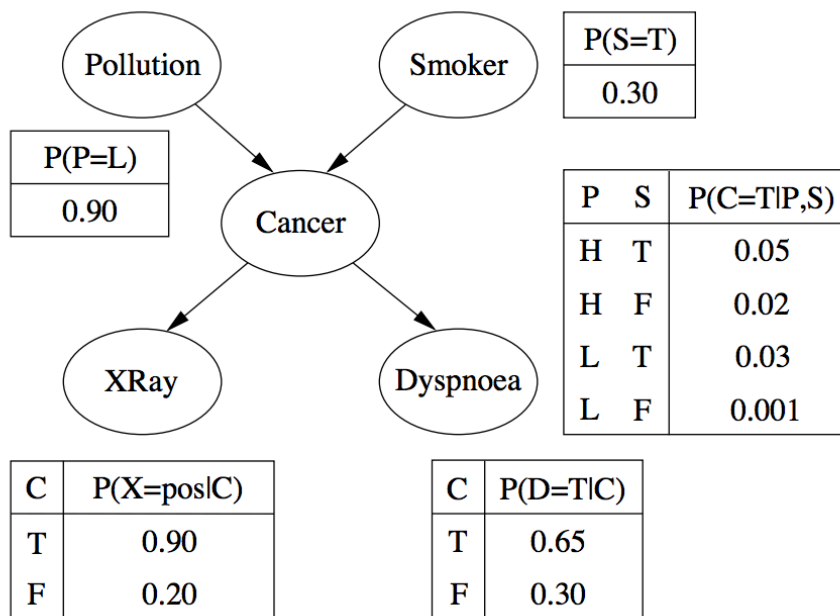CSCI 3202 Introduction to Artificial Intelligence
Instructor: Hoenigman
Assignment 4
Due Wednesday, November 12 before 3pm

# Bayes Net Disease Predictor

For this assignment, you will implement a Bayes Net to predict the probability of cancer given several disease variables. The network structure and probability distributions are taken from the Bayes Net Tutorial document posted on Moodle, and are as follows:

| | P(S=T) |
|---|---|
| Smoker | 0.30 |

| | P(P=L) |
|---|---|
| Pollution | 0.90 |

Cancer

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

XRay

Dyspnoea

| 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

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 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.

**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. The node also needs to store a list of the parent nodes. (Note: By 'list' I don't necessarily mean the Python list type, but rather just some way of storing all parents.)

Marginal probabilities should not be stored in the node, but rather, calculated upon user request.
Any other conditional probability information should also not be stored, but rather, calculated upon user request.

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 what functions you will need for your calculations.

**Grading**
We will be writing a grading script for this assignment that will use the command line argument structure listed above. Your code will need to run on the grading script to receive full credit.