**CSE 5522 Homework 1: Probability theory, Bayes' Nets**

Your responses to these questions should be submitted electronically; please see the submission instructions below.

Answers to questions 1, 2 and 3 should either be written in an ASCII text file or a PDF file. PLEASE CONVERT WORD DOCUMENTS TO PDF FILES. You may also write it out by hand and scan/photograph and convert to PDF.

PLEASE NOTE: YOU NEED TO SUBMIT RUNNING INSTRUCTIONS FOR THE TA FOR PROBLEM 4!

1. (20 points) You are a grader for a CSE class that has an option of using Java or C++ for the programming language.   There are only CSE and ECE students in the class; 80% of the class is CSE students. CSE students tend to turn in Java programs more frequently (75% of the time), whereas ECE students are pretty balanced (50% Java, 50% C++).
   1. (5 points) Draw a Bayesian network that describes the situation, including the complete set of Conditional Probability Tables.
   2. (10 points) You receive a C++ program to grade. Is it more likely to be from a CSE or ECE student? Prove your answer.
   3. (5 points) What would the prior class distribution have to be to have a 50/50 chance of CSE or ECE student having turned in a C++ program?
2. (20 points) Let J, K, L, M, and N be five discrete random variables. Assume that I have given you a distribution for P(J|M,N), P(K|J), P(L|J,K), P(M), P(N). Moreover, I will tell you that there are conditional independence assumptions P(L|J,K,M,N)=P(L|J,K), P(K|J,M,N)=P(K|J), and P(N|M) = P(N).
   1. (7 pts) Show, with explicit steps in the derivation, how you could compute P(M|J,K,L,N) in terms of only the given distributions.
   2. (7 pts) Draw a Bayesian network for the above distributions. You need not include CPTs since I didn't give any here.
   3. (3 pts) Which nodes are in K's Markov blanket?
   4. (3 pts) Add a node X to the network such that X is not in J's Markov blanket.
3. (20 points) (Exercise 13.10, modified) Deciding to put probability theory to good use, we encounter a slot machine with three independent wheels, each producing one of the five symbols BAR, BELL, ORANGE, LEMON, or CHERRY with equal probability. The slot machine has the following payout scheme for a bet of 1 coin (where “?” denotes that we don’t care what comes up for that wheel):

BAR/BAR/BAR pays 25 coins   
BELL/BELL/BELL pays 10 coins  
ORANGE/ORANGE/ORANGE pays 5 coins  
LEMON/LEMON/LEMON pays 4 coins   
CHERRY/CHERRY/CHERRY pays 3 coins   
CHERRY/CHERRY/? pays 2 coins   
CHERRY/?/? pays 1 coin

* 1. Compute the expected “payback” percentage of the machine. In other words, for each coin played, what is the expected coin return?
  2. How many coins can the casino offer as the “jackpot” (BAR/BAR/BAR) without (statistically) losing money?
  3. Compute the probability that playing the slot machine once will result in a win (defined as winning anything, including breaking even).
  4. Estimate the mean and median number of plays you can expect to make until you go broke, if you start with 10 coins. You can run a simulation to estimate this, rather than trying to compute an exact number.

4.  (40 points) Poisonous mushrooms!  Using the following data sets, subsetted from the [UCI Machine Learning Repository](https://archive.ics.uci.edu/ml/datasets/Mushroom), create a Naive Bayes classifier that tries to predict whether a mushroom is edible (e) or poisonous (p) given 22 attributes.  Each line in the file is one mushroom instance, with the first field edible or poisonous, and the remaining fields giving information about the mushroom.  See the link above for descriptions about the attributes.  
  
Your program should read in the training file and a testing file (given on the command line), and print the probability of edible or poisonous for each test example, as well as computing the accuracy of the predictions for the test set.

[agaricus-lepiota.data.train.txtPreview the documentView in a new window](https://osu.instructure.com/courses/18256/files/2368730/download?wrap=1)  
[agaricus-lepiota.data.test1.txtPreview the documentView in a new window](https://osu.instructure.com/courses/18256/files/2368734/download?wrap=1)  
  
You may use any programming language, but make sure that it is runnable on stdlinux.  Give the TA instructions on how to run your program.  YOU MAY NOT USE OFF THE SHELF SOLUTIONS - CODE THIS YOURSELF.