CSE 5522 Homework 1: Probability theory, Bayes' Nets

Due 11 September 2015, 11:59 PM

1. ***10 point penalty for every day late (or part thereof) up to three days. E.g., submitting 12 September 2015, 12:00 AM is a 10 point penalty.***

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. 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; 90% of the class is CSE students. CSE students tend to turn in Java programs most frequently (80% of the time), whereas ECE students are pretty balanced (50% Java, 50% C++).
   a. (5 points) Draw a Bayesian network that describes the situation, including the complete set of Conditional Probability Tables.
   b. (10 points) You receive a C++ program to grade. Is it more likely to be from a CSE or ECE student? Prove your answer.
   c. (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 Q, R, S, T, and U be five discrete random variables. Assume that I have given you a distribution for P(Q|T,U), P(R|Q,S,T), P(S), P(T), P(U). Moreover, I will tell you that there are conditional independence assumptions P(R|Q,S,T,U)=P(R|Q,S,T), P(Q|S,T,U)=P(Q|T,U), P(U|T,S) = P(U), and P(T|S)=P(T).

   a. (7 pts) Show, with explicit steps in the derivation, how you could compute P(T|Q,R,S,U) in terms of only the given distributions.
   b. (7 pts) Draw a Bayesian network for the above distributions. You need not include CPTs since I didn't give any here.
   c. (3 pts) Which nodes are in U's Markov blanket?
   d. (3 pts) Add a node X to the network such that X is not in T'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 four symbols BAR, BELL, 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
   LEMON/LEMON/LEMON pays 4 coins
   CHERRY/CHERRY/CHERRY pays 3 coins
   CHERRY/CHERRY/? pays 2 coins
   CHERRY/?/? pays 1 coin

   a. Compute the expected "payback" percentage of the machine. In other words, for each coin played, what is the expected coin return?
   b. How many coins can the casino offer as the "jackpot" (BAR/BAR/BAR) without (statistically) losing money, ass?
   c. Compute the probability that playing the slot machine once will result in a win (defined as winning anything, including breaking even).
   d. 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 + 15 bonus points) In class, I discussed regression models for predicting the price of a house given it size in square feet. The datafile I used for that class can be found on the Carmen website.

   a. (5 points) Write a program to read in the data file and compute the mean and standard deviation, min and max for both price and square feet. Please do not use built-in functions for a language.*Note: we will evaluate your code on another dataset, so your program should take the name of an input file as an argument. Also, don't throw away this code, we will use it in a later assignment.*

   b. (20 points) Following equation 18.3 on p719 (and the slides), modify your program to analytically compute the linear regression coefficients w0 and w1 for the input data file.

   c. (15 points) Following equations 18.4-18.5 (and the discussion following them), write a program to start from a non-optimal w0 and w1 and update using batch gradient descent. Use a very small alpha (for this data set, say alpha=0.0000000001), and start with initial choice of [w0,w1] to be your answer from the last question plus [100,100]. Iterate until convergence (where the difference in the new [w0,w1] and old [w0,w1] is less than epsilon, where you define epsilon). How many iterations did it take to converge? Do you get the same answer as in the previous question? What happens if you make alpha orders of magnitude bigger? Smaller? (Explicitly address these questions in your writeup.)

   d. (Bonus: 5 points) Implement stochaistic gradient descent, where you randomly select training points and update after each one. (Your alpha parameter will need to be different. Why and how?) Does it converge? If so, does it converge more quickly in terms of number of updates? number of points examined?

   e. (Bonus: 10 points) Implement gradient descent for learning quadratic regression (i.e. y=w2*x^2 + w1*x + w0). You'll need to work out for yourself, or do some research, into what the update equations should be.

You may use any language you like as long as it runs on the CSE unix cluster (stdlinux). CHECK THAT YOUR LANGUAGE IS SUPPORTED BEFORE YOU START! Please give the grader instructions on how to run your code. Remember that your code must run on the CSE unix cluster. Did I say that it has to run on the unix cluster? MATLAB has builtin functions for some of this, particularly reading in files. Please do not submit answers using these built-in functions, although you may check your results against this. Please also note: do not copy code from other places -- I want you to work out these equations yourself.

Submission instructions:

Write up all of your answers to the questions in a text file or PDF file so that it can be submitted electronically. Put that file as well as your programs, and ONLY those files, in a directory called hw1, and then create a zip archive of that directory. Submit via the dropbox function on Carmen.

For the code, you should turn in either

- An executable version of your program along with the source code.
- If your program is written in an interpreted language, such as perl, lisp or python, you can turn in just the source. In that case, make sure to use a standard file extension so that we can determine how to run your program.

We will test your program on the CSE dept unix computers, so make sure that the program you turn in is built to run on Linux. Don't depend on cross-platform language implementations, you should make sure your code runs on the CSE dept Linux system. If your program doesn't execute successfully, we will not attempt to recompile it or otherwise correct the error. Your program will be graded solely on whether its actions are appropriate to the inputs we supply in our testing, not on your programming style or design sophistication.

If you don't know what your CSE account is, talk to me ASAP. You do have one.