Homework 1: Due Oct 4th 2016

Prof. Sushmita Roy

Instructions

There are four problems in all. Question 1 and 2 are coding assignments and need to be submitted via the `mi1.biostat.wisc.edu` machine using the accounts we have created for you. Questions 3 and 4 are due in class on Oct 4th. For questions 1 and 2, please put your code into `/u/medinfo/bmi826/2016-hw//hw1/username` where username is your user name you use to login to this machine. You can use either C++, Java, Perl, or Python to implement this program. If you want to use R, please make sure to write an R command line script for me to run your program.

Your program should be executable on `mi1.biostat.wisc.edu`. I will run this using one of the following versions depending upon the language. Assume the name of the executable of your program is `myprogram` and it takes in two inputs, `input1` and `input2` and produces an output file `output1`, I will execute this as follows:

```
C++: ./myprogram input1.txt input2.txt output.txt
Java: java ./myprogram input1.txt input2.txt output.txt
Perl: perl ./myprogram input1.txt input2.txt output.txt
```

Please note, I am most familiar with C++.

Problem 1 (15 points). Breadth-first search on graphs

The goal here is to store a graph in memory and try out the breadth-first search graph traversal algorithm. Write a program `findReachableNodes`, which takes as input a graph, a binary flag (0 or 1) to denote whether it is directed or undirected, and an input node. If the binary value is 0, the graph should be treated as undirected and if it is 1, the graph should be treated as directed. As output, it should print out all reachable nodes from the input node. The format of `graph.txt` is an edge per line, each line is tab-delimited with two columns, one for each of the vertices of the edge. For a directed graph, the first column is the source and the second column is the target. An example for `graph.txt` is given below:

```
A B
A C
A D
C E
C F
E F
B G
```

Example usage is below:

```
./findReachableNodes graph.txt 0 A
B
C
D
G
E
F
```

Your program should be able to handle exception cases such as the input node is not in the graph or if there are no reachable nodes in the graph. This can be done by printing out informative messages. For example, if there are no reachable nodes your program should return the message: "no neighbors found". If the node is not found in the graph, your program should print out: "node not found".

Problem 2 (15 points). Computing the probability distribution of degrees of an undirected graph

Using the graph data structures you created in the previous problem, in this assignment we will compute the degree distribution of a very large graph with tens of thousands of nodes. The degree distribution of a graph is a discrete distribution which specifies the number of times we observe a node of a particular degree.

Write a program `computeDegreeDist` that takes as input an undirected input graph (formatted as in question 1) and prints out the degree distribution in a tab-delimited file (see example `output.txt` below). Please ensure the output is formatted correctly. The usage of the program is as follows:

```
./computeDegreeDist inputgraph.txt output.txt
```

For example, suppose contents of `inputgraph.txt` is

```
A B
A C
D E
D F
```

And we run the program as follows:

```
./computeDegreeDist inputgraph.txt output.txt
```

The contents of `output.txt` should be

```
1 4
2 2
```

Problem 3 (10 points). Concepts in probability

Assume we are interested in the distribution of food items, namely, drinks and snacks, ordered at a local grocery store. We are also interested in studying if there is any dependency between the food items purchased and the time of day. We will represent these food items and time of day using the random variables D, S and T. D denotes a drink and takes on values $\{$*coffee*, *water*$\}$. S denotes a snack and takes on values

$\{cereal, chips\}$. T denotes the time of day and takes values $\{$*morning*, *noon*$\}$. In each row of the table below, you are given nine observations for all three random variables. That is, each row in the table below represents a combination for D, S and T. Using these observations, address the questions (a)-(f) below. Please show your work for each question by computing the required probability values.

D	*S*	*T*
coffee	chips	morning
coffee	cereal	morning
coffee	cereal	morning
coffee	cereal	morning
water	cereal	morning
water	chips	noon
coffee	chips	noon
water	cereal	noon
coffee	cereal	noon

(a) Estimate the probability distribution, $P(S)$

(b) Estimate the joint probability distribution, $P(S, D)$

(c) Estimate the marginal probability distribution of $P(S)$ from the joint, $P(S, D)$.

(d) Is *S* independent of *D*?

(e) Is *S* independent of *D* given $T =$ *morning*?

(f) Is *S* independent of *D* given $T =$ *noon*?

Problem 4. (20 points) Bayesian networks

You are given the following set of observations for each of the five binary random variables, A, B, C, D and E and the following partially known Bayesian network structure. This Bayesian network is partially known because we do not know the parents of A yet.

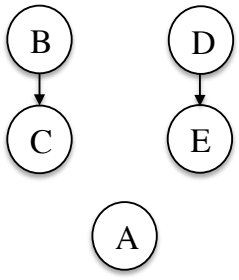

Figure 1: A partial Bayesian network for five boolean random variables

A	B	C	D	E
0	1	1	1	1
1	1	0	1	0
1	0	0	1	0
1	0	0	1	0
0	0	0	1	1
1	0	0	0	1
1	1	1	0	1
0	0	1	0	1
0	0	1	1	1
0	1	1	0	0

Table 1: Samples of observations for the random variables

(a) Using the Bayesian network structure of **Figure 1**, estimate the relevant conditional probability tables for B and C.

(b) Describe a score-based procedure to determine the parents of A. Assume that A can have no more than 2 parents and the parents can be any of the other variables. Assume that the rest of the network is the same as in **Figure 1**.

(c) Briefly describe a greedy structure learning algorithm for Bayesian networks. Describe one advantage and weakness of a greedy structure search learning algorithm.