The purpose of this problem set is to gain experience with probabilistic inference methods. In the engineering design section you will be building a generic tool for probabilistic inference using approximations.

You can complete the exercises by either directly marking up this pdf, or by printing, completing, and scanning as a pdf. You should complete the Engineering Design Problems by writing the python code as instructed. The resulting pdf and python files should be uploaded to Canvas via the assignment tab by the due date and time.

Exercises

1. Suppose that $f_X(x)$ is a probability density for a random variable X, with $x \in R$; and $P[Y]$ is a probability mass function for a random variable $Y \in \{-3, -2, -1, 0, 1, 2, 3\}$.

   (a) (2 points) What is the expression that will compute the probability that X is positive?

   (b) (2 points) What is the expression that will compute the probability that Y is positive?

   (c) (2 points) Suppose that $P[Y]$ is uniform distributed. What is it's probability mass function?

   (d) (2 points) What are the possible factorings of the joint density of X and Y? Use a lowercase f to denote density functions and P to denote mass functions.

2. Suppose that there are four factors that affect a student's grade in a course, the average grade in prerequisite courses, the difficulty of the course, the intelligence of the student, and the work ethic of the student.

   (a) (2 points) Describe how you would model each of the variables to account for uncertainty.

   (b) (2 points) Sketch a Bayesian Network showing the dependencies among the variables.

   (c) (2 points) Suppose the relationship of factors to grades was a noisy-or. What might the node probabilities look like for each variable? You can use unspecified constants, but specify their semantic interpretation.

3. Suppose you are trying to decide whether to take a job or not. You are primarily concerned about two issues, the pay and your overall interest in the job's duties. Casting the above situation as an inference problem,

   (a) (2 points) Define the random variables involved and their domain, giving a brief description of their semantic interpretation.

   (b) (2 points) Define a reasonable probability relationship among the variables, giving a brief rationale for your choices.

   (c) (2 points) What is the expression required to help you make your decision?

4. (5 points) Suppose two agents have identical internal state variables, X, and evidence variables, E, but different models, M_1 and M_2. Which of the following probabilities is most important for deciding which agent should perform better (circle one)? For simplicity, assume all random variables are discrete.

   (a) $P[X]$

   (b) $P[E]$

   (c) $P[M_1]$ and $P[M_2]$

   (d) $P[X, E, M_1]$ and $P[X, E, M_2]$

   (e) $P[M_1|E]$ and $P[M_2|E]$

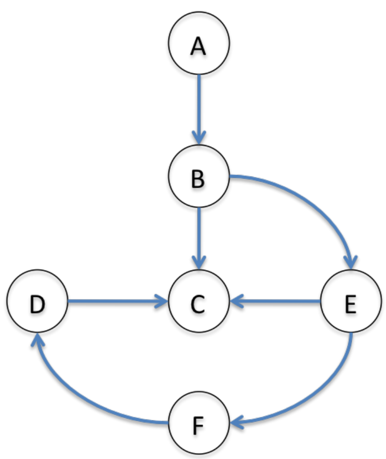
   (f) $P[X|E, M_1]$ and $P[X|E, M_2]$

   Provide a rationale for your choice:

5. (5 points) Given the following factorization of discrete random variables A, B, C, D, E what is the associated Bayesian network structure?

$$P(A, B, C, D, E) = P(E|B, D)P(D|C)P(C|B)P(B|A)P(A)$$

6. (5 points) Given the following graph structure, is it a valid Bayesian network (justify your answer)? If so, what is the associated factorization of the random variables?



7. Suppose you are given the following Bayesian Network for binary random variables

$$P(A, B, C, D) = P(D|B, C)P(B|A)P(C|A)P(A)$$

where

$P(A = 1) = 0.4$,

A	$P(C = 1\|A)$
0	0.2
1	0.8

,

A	$P(B = 1\|A)$
0	0.1
1	0.9

,

B	C	$P(D = 1\|B, C)$
0	0	0.1
0	1	0.3
1	0	0.2
1	1	0.4

Using exact inference determine the following (show your work)

(a) (2 points) Supposing that $B = 0$ and $D = 1$, what is the $P[C = 1]$?

(b) (2 points) Supposing that $A = 1$, $B = 0$, and $C = 0$, what is the $P[D = 0]$?

8. Considering the likelihood-weighting and Gibbs sampling algorithms, answer the following questions:
   (a) (2 points) In both algorithms, what is the purpose of the NORMALIZE function?

   (b) (2 points) In the likelihood-weighting algorithm, when would you expect the weight to be high?

   (c) (2 points) In the Gibbs sampling algorithm, how is the Markov blanket defined?

   (d) (2 points) Suppose I have a Bayesian Network that consists of only one variable. What (if any) difference is there between the algorithms?

9. Suppose that your old clunker of a car breaks down and you have it towed to a mechanic. The mechanic believes the problem could be the battery, the starter, or some unknown cause and asks you what you want to do: replace the battery, replace the starter, or sell him the car. Now that you have taken this course and know how to be a rational decision maker,

   (a) (5 points) What questions would you ask him to inform your decision?

   (b) (5 points) Formulate the expression you would use to make your decision.

Engineering Design Problems

10. (25 points) The purpose of this problem is to gain experience with probabilistic inference by writing a generic program for performing approximate inference in Bayesian Networks of Bernoulli random variables (probabilistic propositions).

You should upload your code (bernoulli.py) and the requested README.txt file as attachments to the Assignment on Canvas.

Program Specification

Your program should be named bernoulli.py. On startup the program should read a file (in the format described below) specifying a Bayesian Networks of Bernoulli random variables. The file name, the number of samples to use, and the inference algorithm should be read from the command line (in that order). The program should work for any size network that will reasonably fit in memory.

After reading the file, the program should prompt the user for one of four commands, Describe, Tell, Ask, and Quit. If any command is malformed the program should print an error and prompt again.

The Describe command should print the available variables and if they are an evidence variable or not, i.e. if a value is assigned to that node. The Quit command should simply exit the program.

The Tell command should be of the form:

```
Tell VAR {0,1,?}
```

where VAR is a variable in the network, setting it to 0 or 1 respectively, or clearing it in the case of ?.

The Ask command should be of the form:

```
Ask VAR
```

where VAR is any variable in the network, printing the P(VAR) using the selected inference algorithm, either "weighted" or "gibbs". If "weighted" is specified, the program should use the likelihood weighting approximate inference algorithm described in section 14.5 of your text (Figure 14.15). If "gibbs" is specified, the program should use the Gibbs sampling approximate inference algorithm described in section 14.5 of your text (Figure 14.16).

For example, to run your program on the BN described in the file test.net using 1000 samples and Gibbs sampling, the program would be invoked as

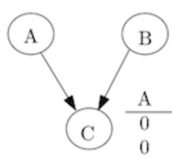
```
python bernoulli.py test.net 1000 gibbs
```

File Format

The file format is text based. The first line are two space separated integers, N and M, specifying the number of nodes and connections in the network respectively. The next N lines are single words indicating the text label for each node. The next M lines are space separated words specifying the connections in the network, with the arc going from the first to the second label. The last N lines specify the probability table of each node, with the first word being the node label followed by P space separated floating point numbers, one for each line of the associated probability table in lexical ordering of the variable labels. Thus the total number of fields on the line depends on the in-degree of that node. For example, the following text file describes the example Bayesian network in the figure below.

```
3 2
A
B
C
A C
B C
A 0.3 0.7
B 0.75 0.25
C 0.1 0.3 0.45 0.15
```

A	P(A)
0	0.3
1	0.7



B	P(B)
0	0.75
1	0.25

A	B	$P(C\|A,B)$
0	0	0.1
0	1	0.3
1	0	0.45
1	1	0.15

Analysis

Create a series of networks (at least five) of varying size and CPT entries. Explore how the query results change as a function of the network structure, CPTs, number of samples, and inference algorithm. Record your observations in plain text file named README.txt.