Homework 2

15-381/681: Artificial Intelligence (Fall 2017)

Out: October 18, 2017
Due: October 29, 2017 at 11:59PM

Homework Policies

- Homework is due on Autolab by the posted deadline. Assignments submitted past the deadline will incur the use of late days.
- You have 6 late days, but cannot use more than 2 late days per homework. No credit will be given for homework submitted more than 2 days after the due date. After your 6 late days have been used you will receive 20% off for each additional day late.
- You can discuss the exercises with your classmates, but you should write up your own solutions. If you find a solution in any source other than the material provided on the course website or the textbook, you must mention the source. All homeworks (programming and theoretical) are always submitted individually.
- Strict honor code with severe punishment for violators. CMUs academic integrity policy can be found here. You may discuss assignments with other students as you work through them, but writeups must be done alone. No downloading / copying of code or other answers is allowed. If you use a string of at least 5 words from some source, you must cite the source.

Submission

For the written portion, please submit your solutions as a pdf to Gradescope.
For the programming portion, please create a tar archive containing just `sampling.py`, and submit it to Autolab. The programming portion will be autograded.

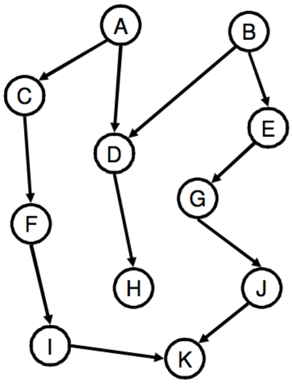
1 Part 1: Written [50 Points]

1.1 Hosting Tournament [7 Points]

You are hosting a tournament with n players in a circle. You want to make a binary random generator that gives X belonging to $X \in \{\text{left}, \text{right}\}$. Each round, every person in the circle will simultaneously flip the random generator and turn to their X side. A match happens when two players are facing each other. For each match, the person who flipped 'right' will leave the circle. Then, the next round begins. The tournament continues until there is only 1 player left.

1. What's the probability you want to assign to X if you want the tournament to end fast? Explain. (Hint: you want to maximize the expected number of matches).

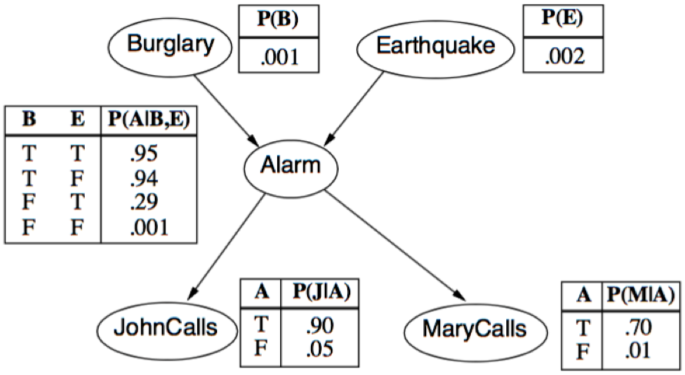
1.2 Bayes Net Independence [9 Points]



Using the above Bayes net, determine whether the following statement is True or False. Explain your answers.

1. $P(A, F|C, H, K) = P(A|C, H, K)P(F|C, H, K)$
2. $P(E, F|D, I, J) = P(E|D, I, J)P(F|D, I, J)$
3. $P(E, F|B, I, K) = P(E|B, I, K)P(F|B, I, K)$

1.3 The Burglar Scenario [14 Points]



1. (8 points) Enumerate all the 2^5 joint probabilities of the five variables in the above Burglary/Earthquake Bayes Net. Compute conditional probabilities $P(b, e|a)$, $P(b, e|\neg a)$, $P(b|a)$, $P(e|a)$ and marginals $P(A)$, $P(M)$, $P(J)$, where capital letter means Variable, and small letter means the variable is True (e.g. $A = a$) or False (e.g. $A = \neg a$).
2. (6 points) Consider the Burglary/Earthquake Bayes Net given, when no evidence is given, the Burglary and Earthquake variables are independent. However, if we observe Alarm = true, things change. In particular, Burglary and Earthquake are no longer conditionally independent. Explain why. Justify your answer by computing the appropriate probabilities involved. For example, you may want to evaluate $P(B, E|a)$, $P(B|a)$, $P(E|a)$.

1.4 Approximate Inference of the Bayes Nets [20 Points]

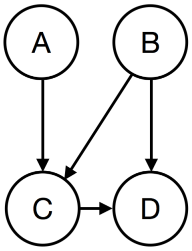
The objective is to experiment with the sampling algorithms you implemented in Part 2 so as to understand their utilities, strengths and limitations. You can work on this section concurrently as you work on the programming assignment – the solutions here can help you check your work.

Mid-term Net

Consider the Bayes Net you have considered in the midterm. The conditional tables of the mid-term net has been coded in the required input format for you in Part 2 so that you can apply the appropriate sampling algorithms to perform approximate inference.

P(A)	
+A	0.0
-A	1.0

P(B)	
+B	0.9
-B	0.1



P(C | A,B)			
+A	+B	+C	0.2
+A	+B	-C	0.8
+A	-B	+C	0.6
+A	-B	-C	0.4
-A	+B	+C	0.5
-A	+B	-C	0.5
-A	-B	+C	0.0
-A	-B	-C	1.0

P(D | B,C)			
+B	+C	+D	0.75
+B	+C	-D	0.25
+B	-C	+D	0.1
+B	-C	-D	0.9
-B	+C	+D	0.5
-B	+C	-D	0.5
-B	-C	+D	0.2
-B	-C	-D	0.8

1. (5 points) Compute the probabilities of $P(d|c)$, $P(b|c)$, and $P(d|\neg a, b)$ by enumeration. Then, employ rejection sampling, likelihood weighting, and Gibb's sampling to approximate them. Use 1000 samples, and document your results below. How do the approximations and the actual values compare?

2. (5 points) Next, we focus on $P(d|c)$. We know that the accuracy of the approximation depends on the number of samples used. For each of the three sampling methods, plot the probability as a function of the number of samples used. Do you notice large differences in the convergence rates among the methods?

3. (5 points) Construct a query on this Bayes Net such that the convergence and effectiveness of rejection sampling is noticeably worse than that of likelihood weighting and Gibb's sampling. List the query you are using, and provide the probability plot as a function of samples used. Why is it the case that rejection sampling is noticeably worse for this query?

Burglary Net

Consider the Burglary Net from Question 1.3 above.

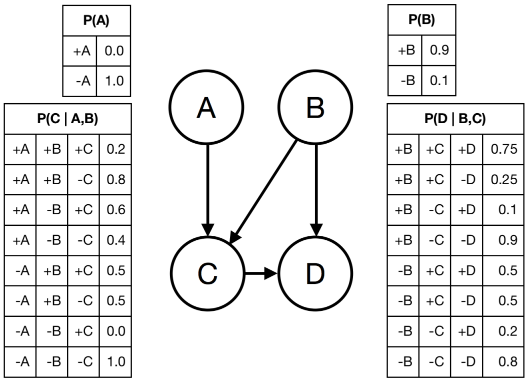
1. (5 points) Code the Burglary Bayes Net conditional tables into the required input format for Part 2's program. Employ rejection sampling, likelihood weighting, and Gibb's sampling to approximate $P(e|A), P(b|a)$, and $P(A)$. Use 1000 samples, and document your results below. How do the approximations and the actual values compare?

2 Part 2: Sampling: Programming [50 Points]

2.1 The Problem

Your task here is to implement three types of sampling techniques to perform approximate inference on any given Bayes Net: Rejection Sampling, Likelihood Sampling, and Gibbs Sampling. You will employ these algorithms to answer the written Question 1.4 above.

An example Bayes Net is given in the midterm, which we called Midterm Net:



P(A)	
+A	0.0
-A	1.0

P(B)	
+B	0.9
-B	0.1

P(C | A,B)			
+A	+B	+C	0.2
+A	+B	-C	0.8
+A	-B	+C	0.6
+A	-B	-C	0.4
-A	+B	+C	0.5
-A	+B	-C	0.5
-A	-B	+C	0.0
-A	-B	-C	1.0

P(D | B,C)			
+B	+C	+D	0.75
+B	+C	-D	0.25
+B	-C	+D	0.1
+B	-C	-D	0.9
-B	+C	+D	0.5
-B	+C	-D	0.5
-B	-C	+D	0.2
-B	-C	-D	0.8

You will develop codes to compute conditional probabilities, such as $P(d)$, $P(d|c)$, $P(b|c)$, or $P(d|-a,b)$, where d = (D=T) = (+D) in our convention. Note that we only have one query variable, and an arbitrary number of evidence variables.

2.2 Format

Please use Python 2.7 to complete this assignment, in a file named `sampling.py`. We have given you some starter code (in the Autolab handout) that provides a working solution for direct sampling (ignores evidence variables). In this file, please fill in the three methods, one for each sampling technique. The specifics of what each line of the input and output should be is described below.

2.2.1 Input

The format of the input file is as follows:

1. The first line of every file contains the names of all nodes in the graph, in topological (& lexicographical) order.
2. The lines that follow until the first blank line contain conditional probabilities to describe the local tables of the Bayes Net. These will always be given to describe the distribution in topological (lexicographical) order. Each line contains zero or more conditions (denoted by a +/- followed by a variable name, then a space), the dependent variable (denoted with just the variable name and a space), then the probability as a float.
3. Finally, after a single empty line, the file will contain a single query, which asks the probability of a particular variable, given the evidence that follows.

For example, the input file for the above (midterm) example would look like the following:

```
A B C D
A 0.0
B 0.9
+A +B C 0.2
+A -B C 0.6
```

```
-A +B C 0.5
-A -B C 0.0
+B +C D 0.75
+B -C D 0.1
-B +C D 0.5
-B -C D 0.2

D +C
```

Note that the query corresponds to $P(D|C)$. The sign of the variable that you have to compute is always positive.

2.2.2 Output

Please print your output to stdout.
For each test case, print a single floating point number. This number should represent the value of the corresponding test case.

2.3 Method

Please refer to the textbook or lectures for implementation details with regards to each sampling technique. If needed, you can compute the conditional probability of a variable given its Markov blanket using the following equation (taken from 14.12 in the book):

$$P(x_i'|mb(X_i)) = \alpha P(x_i'|parents(X_i)) \times \prod_{Y_j \in Children(X_i)} P(y_j|parents(Y_j))$$

In each sampling method, make sure to follow the topological variable order. A specific random.seed(10000) has been provided in the starter code, please keep this seed when you are submitting to autolab. Additionally, you should only generate random assignments using the sample function provided; this is to help ensure the random samples generated will be identical to that of the autograder. For Gibb's sampling, please initialize all non-evidence variables to True.

2.4 Testing

You are encouraged to manually test your code with the results you calculate from problem 1.4. To run your code, you should run `python sampling.py <filename> <trials> <sampling_type>`. `trials` refers to the number of samples you want to generate, and `sampling_type` can be 0,1,2, or 3 depending on which sampling method you want to use.

When you submit to Autolab, we will compare your results with the results generated from the reference solution on several different Bayes Nets and queries. Because we have provided the specific random seed, we expect the output to be nearly identical to the reference output (we will check up to 5 decimal places). If you find that your code seems to be working correctly (i.e. correct answers from section 1.4) but the autograder is not passing, be sure that you are looping through the variables in topological order. If you think it is some precision error in Python, please post on Piazza.

The Autolab tests will generate your final score. There are currently no restrictions on the number of Autolab submissions.

2.5 Submission

Please submit to Autolab a tar file that contains the file named `sampling.py` and `README.txt`. The `README.txt` file should contain anyone that you've collaborated with in this assignment. To create a tar file, you may run the following command on the same directory as your `sampling.py` and `README.txt` files:

```
tar cvzf handin.tar sampling.py README.txt
```

Please submit your written section to Gradescope.