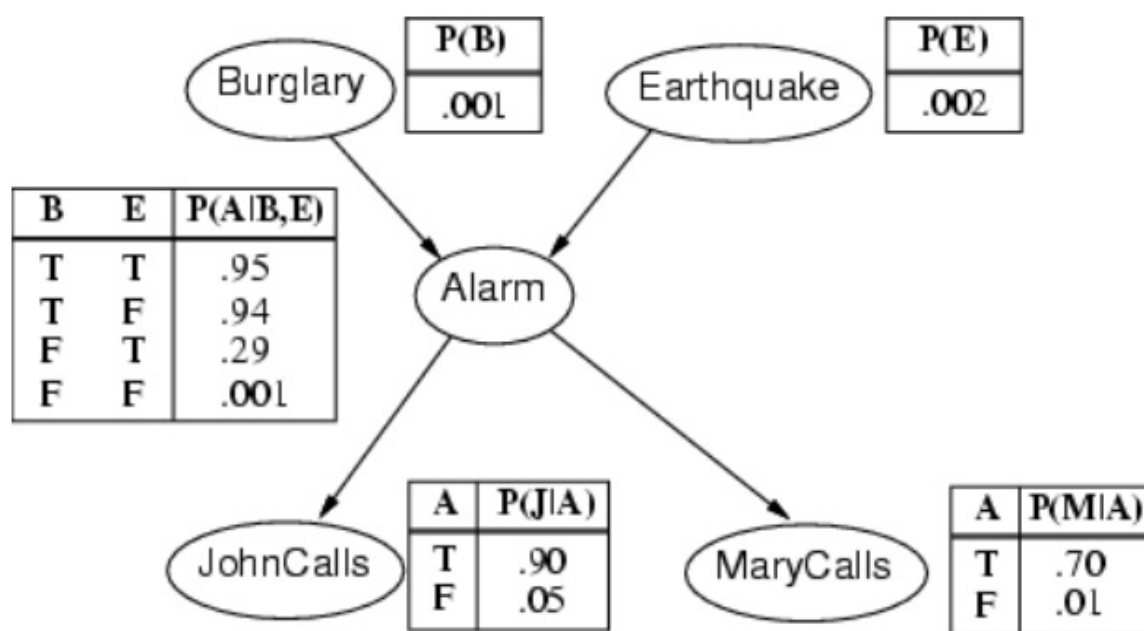


Inference in Bayesian Network

PA3 (Due Nov 20)

The goal of this programming assignment is to understand the relative properties of the inference algorithm. In order to do so, you are required to implement four inference algorithms enumeration, prior sampling, rejection sampling, and likelihood weighting. Evaluate them with the alarm bayes net presented in the class, shown in Figure 14.2 in the Russell and Norvig book and implemented in the `alarmNetwork.py`.



For approx inference methods, you need to vary number of samples between 1 to 1000 and then run the algorithms. This will explain how the probabilities vary with the number of samples. Report your results for the following queries:

- Alarm is false, infer Burglary and JohnCalls being true.
- JohnCalls is true, Earthquake is false, infer Burglary and MaryCalls being true.
- MaryCalls is true and JohnCalls is false, infer Burglary and Earthquake being true.

For each query, you need to submit a table as shown below. Note that the last row in the table is the result of the enumeration algorithm. You should run each sampling algorithm 10 times for each number of samples and report the average. For example run prior sampling 10 times and average the inferred probability over the test queries.

No of Samples	Prior Sampling	Rejection Sampling	Likelihood Weighting
1			
...			
...			
...			
1000			
Enumeration		X	X

Additional instructions

- Functions are defined in the `inference.py` for all four inference algorithms. You have to add the code underneath. You may add more functions as needed. You are expected to submit only this file and the report.
- You will find the alarm bayesian network implemented in the `alarmNetwork.py`. It is a very naive implementation and assumes that the evidence provided when getting conditional probability are in the topological order. So, take care of that in topological order.
- For simplicity, `alarmNetwork.py` uses the node indexes as A, B, E, J, M corresponding to alarm, burglary, earthquake, John and Mary calling.
- For sampling, use the `psuedorandom.uniform()` defined in the `alarmNetwork.py`.
- Program takes string as input query in the following format: `[< E, t >> J, f >][M, A]`. The first list contains the evidence variables and their values. The next list contains the query variable. The above input has Earthquake:True, JohnCalls:True and query variable are MaryCallas and Alarm.
- You are expected to return the probability of the True value for each of the query variable.
Sample Output: `['<M, 0.6176188392283065>', '<A, 0.8806070133743574>']`
- Sample code to run inference is provided in `driver.py` and `report.py`.

I acknowledge that the code and the comments in there are not up to the mark. Any feedback or questions are welcome.