Name: Shuqi Liu

November 12th, 2016

Bayesian Network and Basic Queries

**The Problem**

We are given a Bayesian network with five nodes: Earthquake, Burglary, Alarm, JohnCalls and MaryCalls. Alarm is conditionally dependent on Burglary and Earthquake. JohnCalls and MaryCalls are conditionally dependent on Alarm. We are asked to compute the probability of any combinations of events involving the five nodes.

**Algorithm and Implementation**

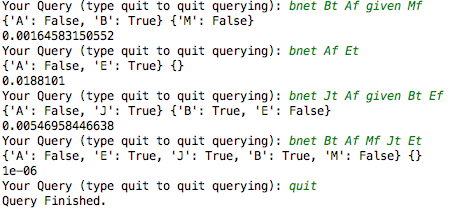
The probability inference is implemented with the rejection sampling algorithm. The algorithm is implemented in Python. In the *priorSample()* method, the probabilities of each individual event and their dependencies are hard coded. The *priorSample()* method returns a dictionary of the Boolean assignments of the five nodes to help check consistency in the later when the input query and evidence variables might not be in order. The key is the initial of the node and the value is the assignment of the event. For example, the key and value pair {‘B’: True} would represent that Burglary did happen. The assignment is randomly generated according to its probability. In the random generation, the program assumes that the probability given is within two decimal places.

The rejection sampling algorithm takes query, evidence, and trials as parameters. The trials are set to 1,000,000. Query and evidence are dictionaries generated from standard input following the format “bnet Bt Af given Mf”, which represents the query of the probability of Burglary happens and Alarm does not ring given that Mary calls. The program will keep reading input until user types “quit”. The program also assumes that the query from user input always follows the right format and thus does not check the correctness of the input.

Query, evidence and the prior sample are both recorded using a dictionary with the event initials as the key and the assignment as the value. By using dictionary, the program could easily look up the value of the evidence and randomly generated sample using the same event key.

**Example Output**

An example for four listed queries is presented below.



By generating random samples following the conditional probabilities, we are able to calculate the probability of combination of different events in a fairly short program. We do not need to work through the exactly mathematical formula to find certain probabilities, which could be complicated with a complex network and multiple nodes. However, it always comes with trade-offs. By using the relatively simple simulation technique, we lost the precision found in the direct calculation.