CS161 Assignment-8

Problem 1

> Suppose that we have a patient who was just tested for a particular disease and the test came out positive. We know that one in every thousand people has this disease. We also know that the test is not reliable: it has a false positive rate of 2% and a false negative rate of 5%. Our goal is then to assess our belief in the patient having the disease given that the test came out positive.
>
> If we let the propositional variable D stand for the patient has the disease, and the propositional variable T stand for the test came out positive, our goal is then to compute Pr(D|T).

Problem Given:

$Pr(D) = 0.001$

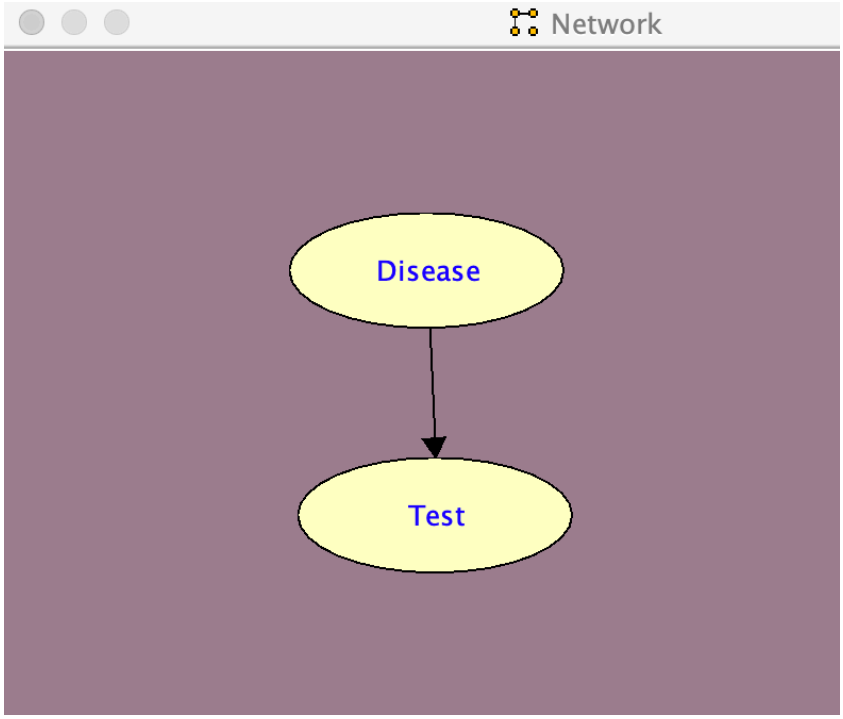
$Pr(\neg D) = 0.999$

$Pr(T|\neg D) = 0.002$

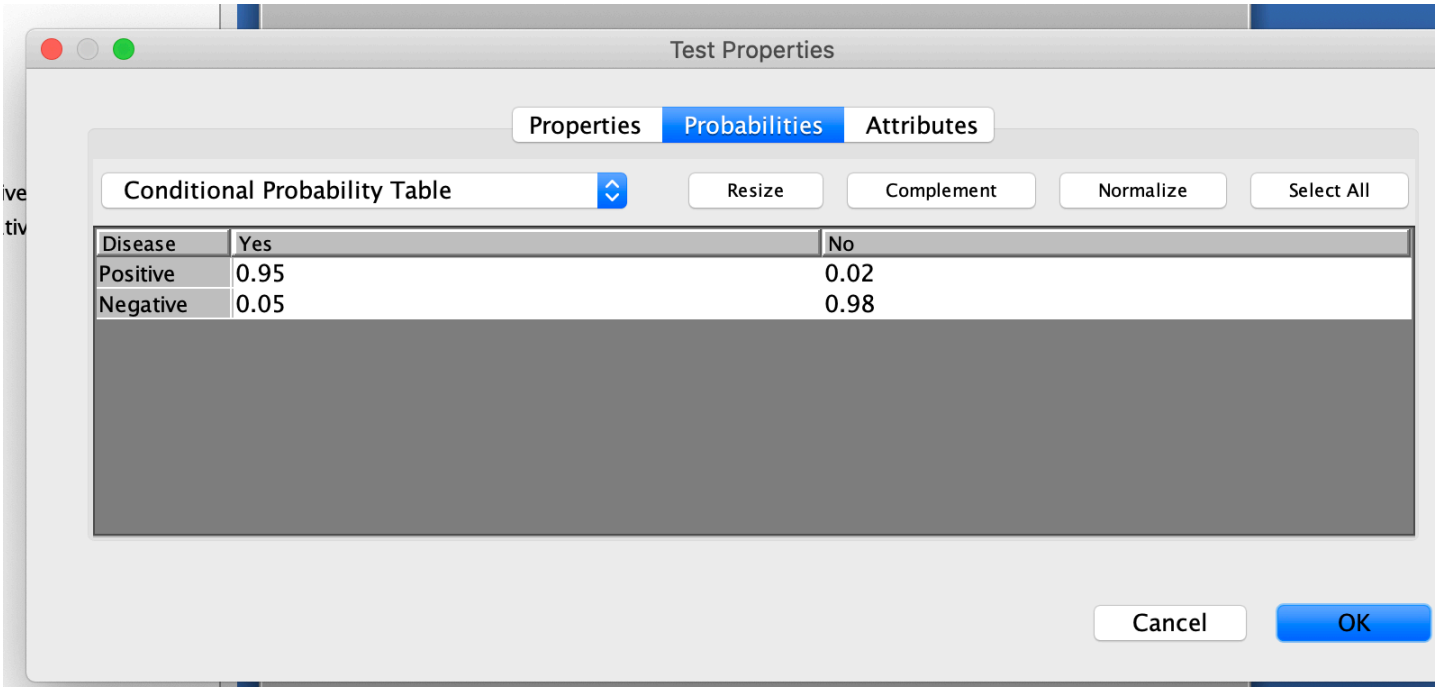
$Pr(\neg T|D) = 0.005$

Goal: $Pr(D|T)$

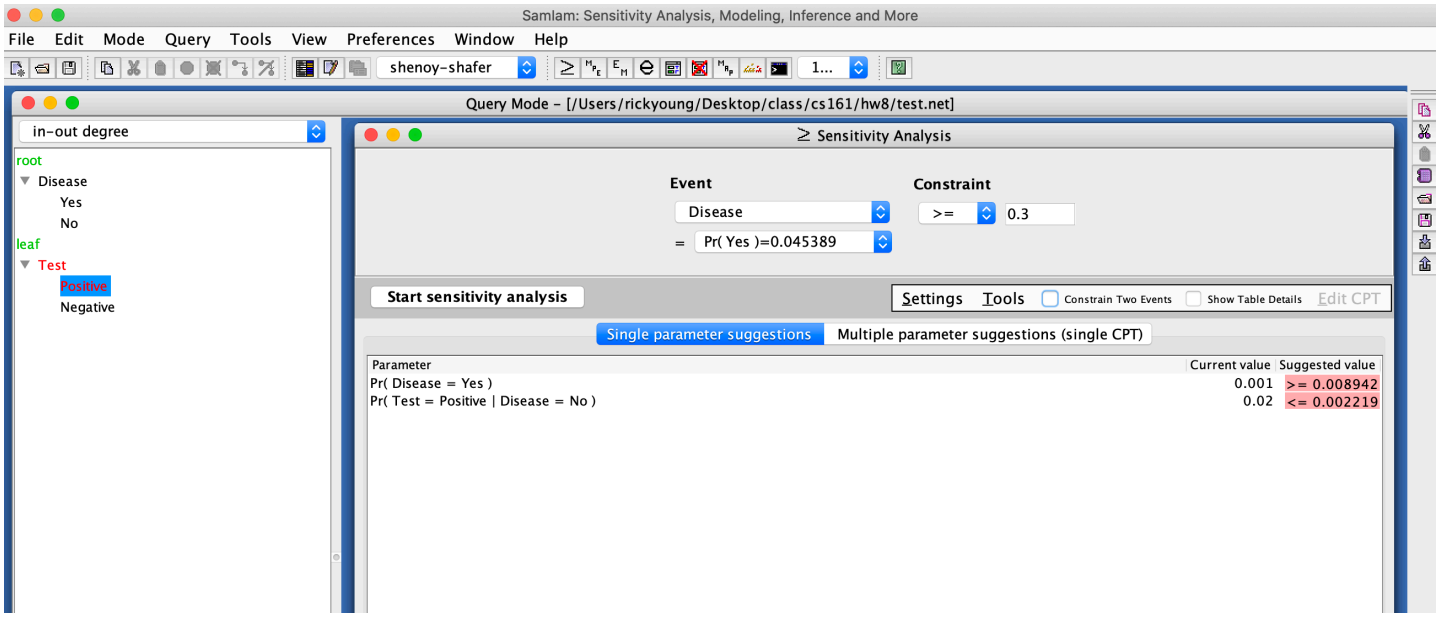
- **Network**:



- **CPTs:**



- **Sensitivity Analysis Result** from Samiam:



As the image shown, the sensitivity analysis indicates that the suggested value for $Pr(D)$ and $Pr(T|\neg D)$. It doesn't output any suggested value for the false negative test because solely changing the false negative probability wouldn't be able to meet the given constraint.

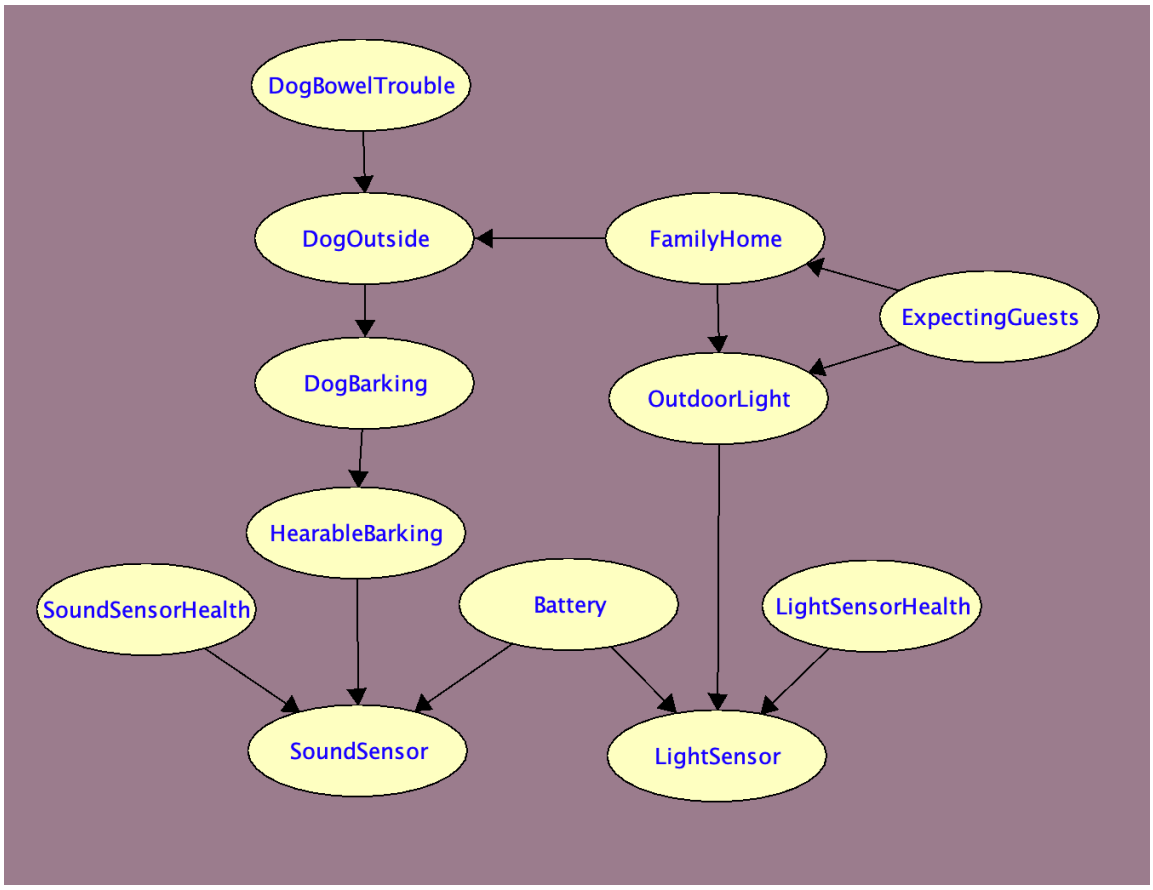
Problem2

> When Sambot goes home at night, he wants to know if his family is home before he tries the doors. (Perhaps the most convenient door to enter is double locked when nobody is home).
>
> Often when Sambot's wife leaves the house she turns on an outdoor light. However, she sometimes turns on this light if she is expecting a guest. Also, Sambot's family has a dog. When nobody is home, the dog is put in the back yard. The same is true if the dog has bowel trouble.
>
> Finally, if the dog is in the backyard, Sambot will probably hear her barking, but sometimes he can be confused by other dogs barking. Sambot is equipped with two sensors:
>
> - a light- sensor for detecting outdoor lights and
> - a sound-sensor for detecting the barking of dogs(s).
>
> Both of these sensors are not completely reliable and can break. Moreover, they both require Sambot's battery to be in good condition.

a) Set of variables and values:

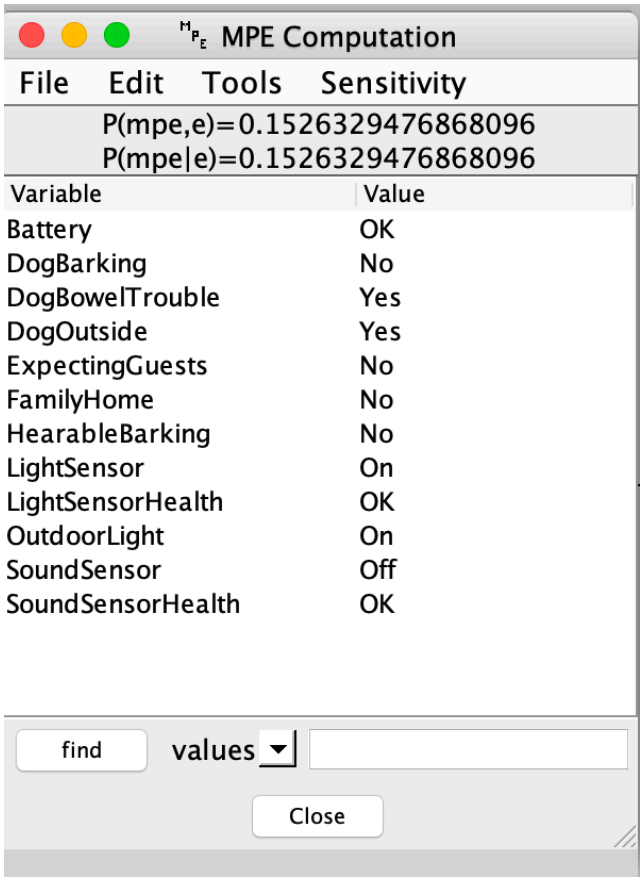
Variable	Value
ExpectingGuests	Yes/No
FamilyHome	Yes/No
SoundSensor	On/Off
LightSensor	On/Off
HearableBarking	Yes/No
Battery	OK/Dead
SoundSensorHealth	OK/Broken
LightSensorHealth	OK/Broken
DogBarking	Yes/No
DogOutside	Yes/No
OutdoorLight	On/Off
DogBowelTrouble	Yes/No

b) Causal structure:



c)

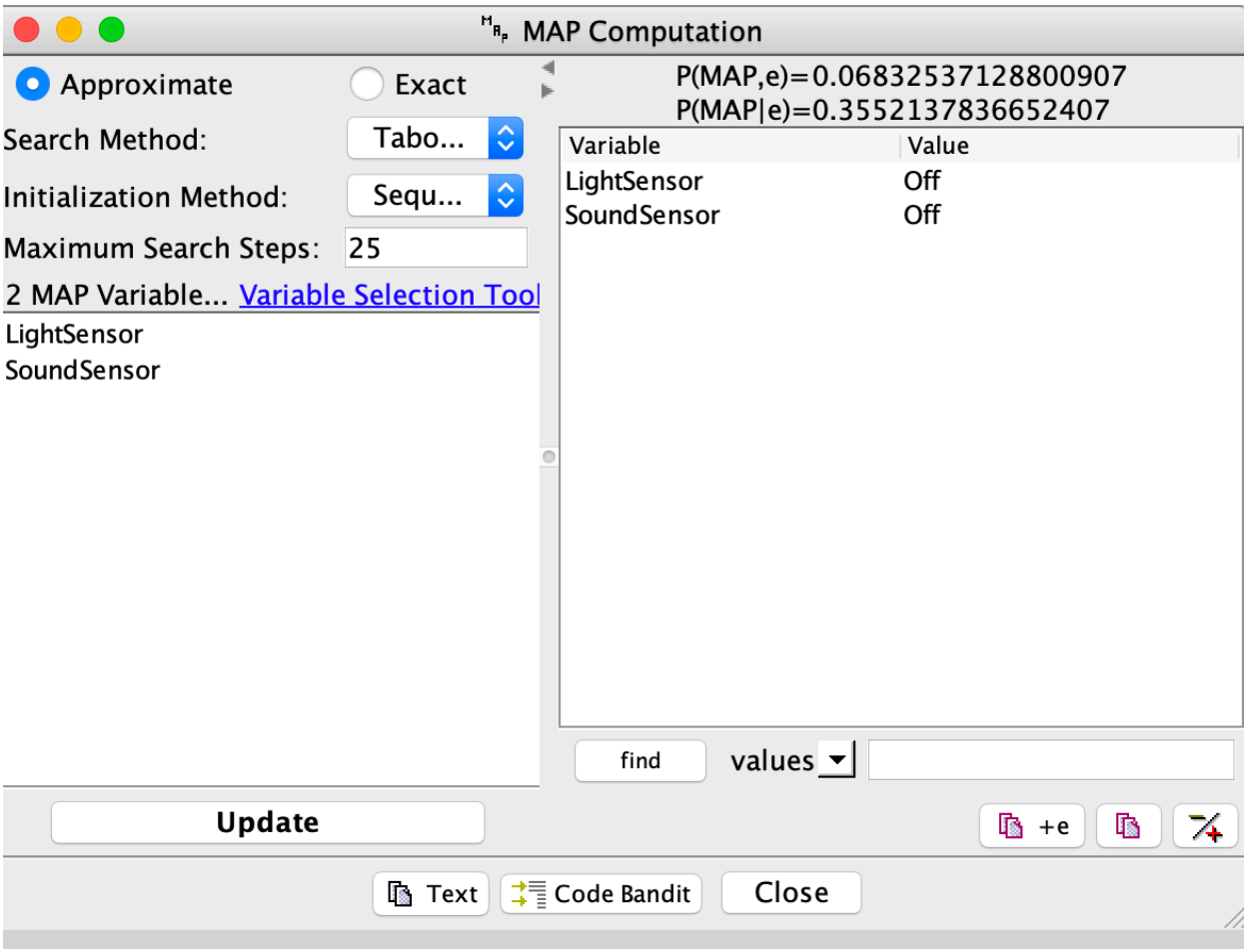
- The most likely instantiation of all variables given that Sambot has sensed the lights to be on, but has sensed no bark as below:



First, I set up the casual structure in my sambot.net and run the EM algorithm in it (load the giving .dat file) to obtain the EM version.

Then I obtained this answer by using Samlam's MPE tool and giving the `LightSensor` variable the value of On and `Hearablebarking` variable the value of No.

- The most likely instantiation of the sensors given that the family is home and no guests are expected.



I obtained this answer by using Samlam's MAP tool and giving the `ExpectingGuests` variable the value of No and `FamilyHome` variable the value of Yes.

- The smallest set Z = { `Battery` , `FamilyHome` } such that the two sensors are independent given Z.

  `FamilyHome` and `Battery` cover all the paths from `SoundSensor` to `LightSensor` in the network. There are three paths:

  - `LightSensor` <- `Battery` -> `LightSensor` : `Battery` is divergent, if it is observed, then the path is blocked.
  - `LightSensor` <- `OutdoorLight` <- `FamilyHome` -> `DogOutside` -> `Dogbarking` -> `HearableBarking` -> `SoundSensor` : `FamilyHome` is a divergent node here, if it is observed, the path is blocked.
  - `LightSensor` <- `OutdoorLight` <- `ExpectingGuests` -> `FamilyHome` -> `DogOutside` -> `Dogbarking` -> `HearableBarking` -> `SoundSensor` : `FamilyHome` is sequential, if it is observed, the path is blocked as well.

  Hence if these two nodes `Battery, FamilyHome` are observed, all paths would be blocked. i.e., d_separated.

- It is a multiply-connected network. Reason: There are more than one path between some nodes in the network. For example, `ExpectingGuests` to `OutdoorLight`, namely, `ExpectingGuests` -> `OutdoorLight` and `ExpectingGuests` -> `FamilyHome` -> `OutdoorLight`.