1. Consider the following problem which was discussed in class:

> 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).

You may also recall being surprised that Pr(D|T) ≈ 0.045. The goal of this question is then to identify conditions under which this probability will be no less than .30. You will need to find the answer to this by constructing a Bayesian Network and using the sensitivity analysis engine of SamIam. You need to turn in test.net and contain the following Information in hw8.pdf :

- Your complete Bayesian network (Structure and CPTs) in test.net file.
- A constraint on each of the following, which is sufficient to ensure that Pr(D|T) ≥ 0.3: The prior probability of having the disease, the false positive for the test, and the false negative for the test. Screenshot the results from SamIam and attach the pictures in the report.

Hint: you need to choose an algorithm like sheony-shafer to run the sensitivity analysis.

P(D) = 0.001
P(¬D) = 0.999
False positive rate = P(T | ¬D) = 0.02
P(¬T | ¬D) = 0.98
False negative rate = P(¬T | D) = 0.05
P(T | D) = 0.95

P(D | T) = ?

P(T) = P(T | D) * P(D) + P(T | ¬D) * P(¬D) = 0.95 * 0.001 + 0.02 * 0.999 = 0.02093
P(D | T) = P(T | D) * P(D) / P(T)
= (0.95 * 0.001)/0.02093
≈ 0.045
≈ 4.5%

Conditions to ensure that P(D | T) >= 0.3:
Single parameter suggestions:

Parameter	Current value	Suggested value
Pr(Test = Positive | Disease = False)	0.02	<= 0.002386
Pr(Disease = True)	0.001	>= 0.008322

Multiple parameter suggestions:

Variable	Log-odds
Disease	2.126232
Test	2.047947

True	0.001	0.008322
False	0.999	0.991678

Variable	Log-odds
Disease	2.126232
Test	2.047947

Disease	True		False	
Positive	0.95	0.710228	0.02	0.002626
Negative	0.05	0.289772	0.98	0.997374

P(D) >= 0.008322

False positive rate: P(T | ¬D) <= 0.002386

False negative rate, in multiple parameter suggestion: P(¬T | D) <= 0.289772

2. Consider the following scenario:

> 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.

Your task is to build a belief network that Sambot will use to reason about the above situation using the modeling and inference tool SamIam. Specifically, given sensory input, Sambot needs to compute his beliefs in various events: whether his family is home, whether any of his sensors are broken, whether the dog is in the backyard, and whether it has bowel trouble. You need to proceed as follows:

(a) Decide on the set of variables and their values. These variables and values must match those in the given data file sambot.dat.

(b) Construct the causal structure.

(c) Learn the network CPTs using the EM algorithm and the data file (sambot.dat) available from the class homepage. Your initial network should have uniform parameters.

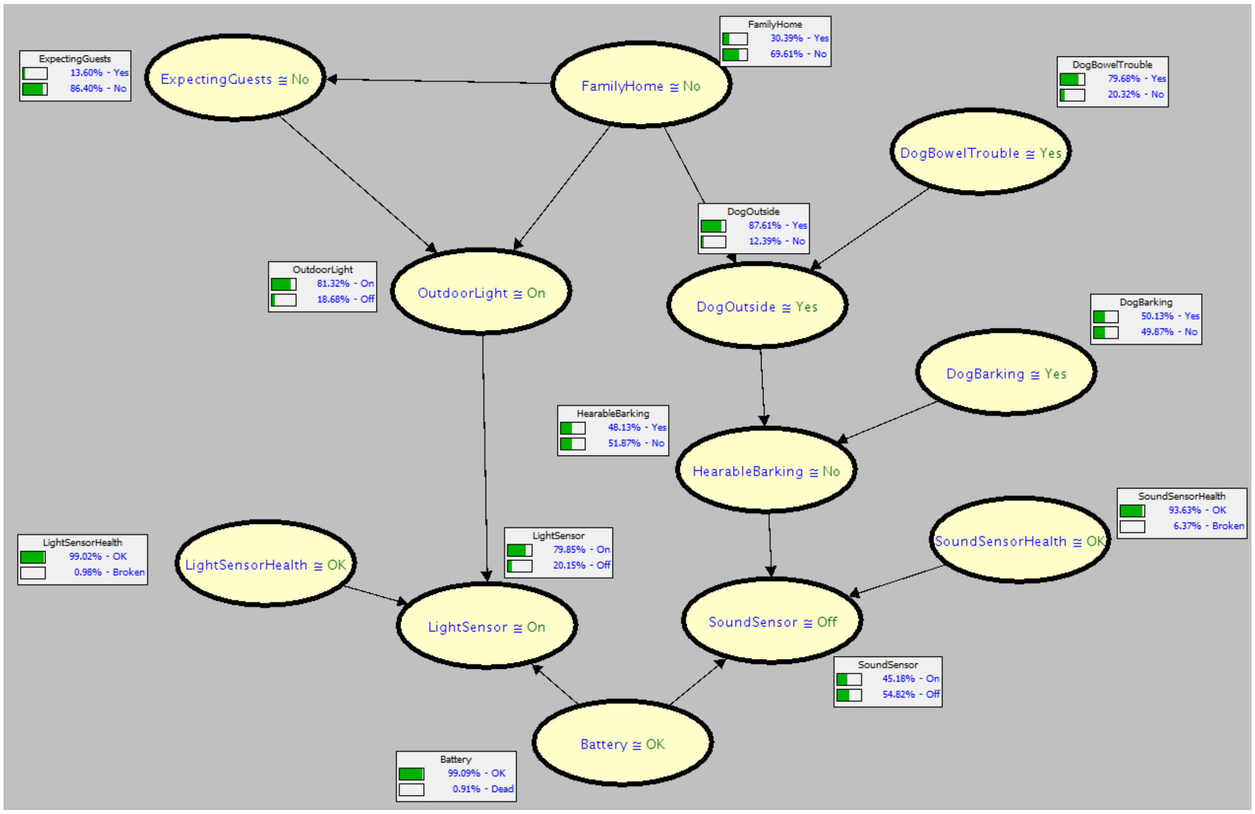
a)

Variable	Set of values
ExpectingGuests	Yes: expecting guests; No: not expecting guests
FamilyHome	Yes: family at home; No: no one at home
SoundSensor	On: detects hearable barking; Off: otherwise
LightSensor	On: detects light; Off: otherwise
HearableBarking	Yes: Any dog barking No: otherwise
Battery	OK: battery is on Dead: battery is dead
SoundSensorHealth	OK: sound sensor is not broken Broken: sensor is broken
LightSensorHealth	OK: light sensor is not broken Broken: sensor is broken
DogBarking	Yes: other dogs barking No: no other dogs barking

DogOutside	Yes: own dog is outside No: own dog is indoor
OutdoorLight	On: outdoor light is on Off: outdoor light is off
DogBowelTrouble	Yes: own dog has Bowel No: otherwise

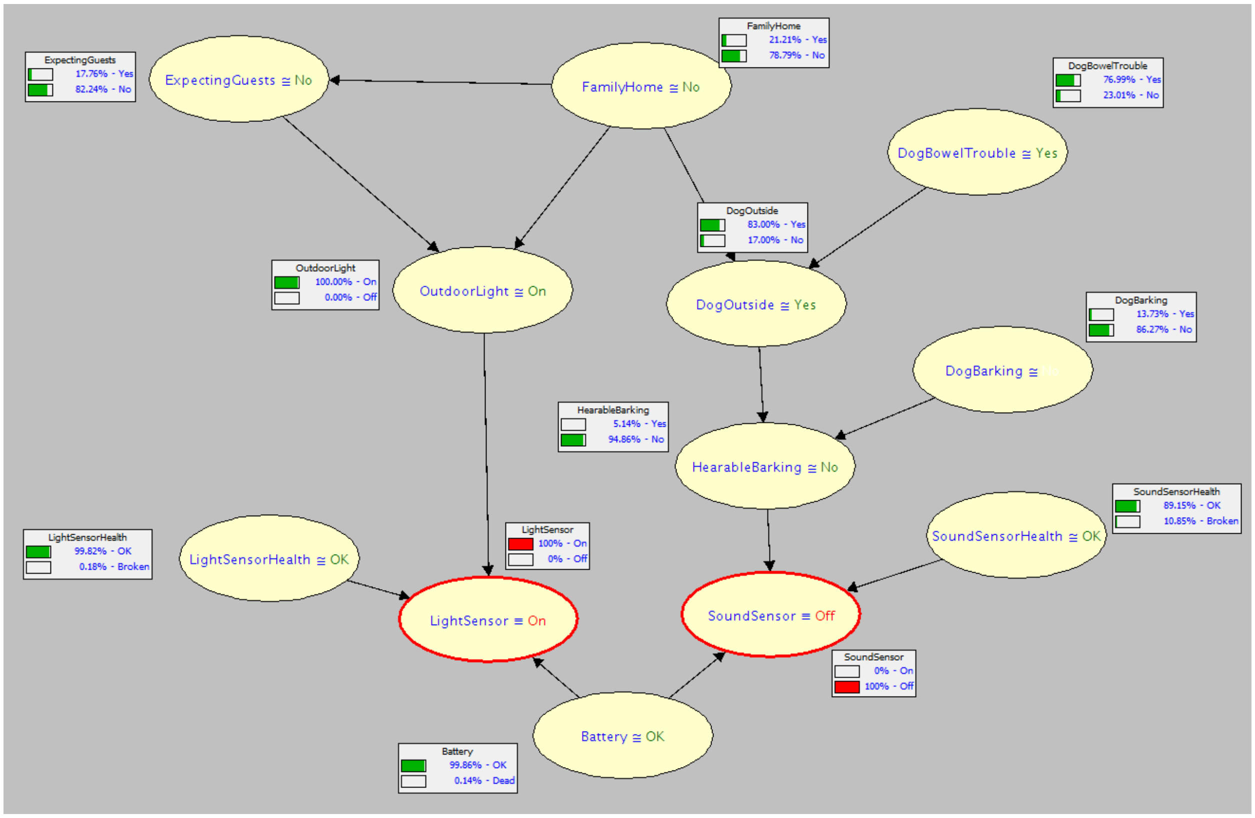
b) Causal structure

FamilyHome → OutdoorLight
ExpectingGuest → OutdoorLight
FamilyHome → DogOutside
DogBowelTrouble → DogOutside
DogOutside → HearableBarking
DogBarking → HearableBarking
HearableBarking → SoundSensor
OutdoorLight → LightSensor
SoundSensorHealth → SoundSensor
LightSensorHealth → LightSensor
Battery → SoundSensor
Battery → LightSensor
FamilyHome → ExpectingGuest (from Piazza)

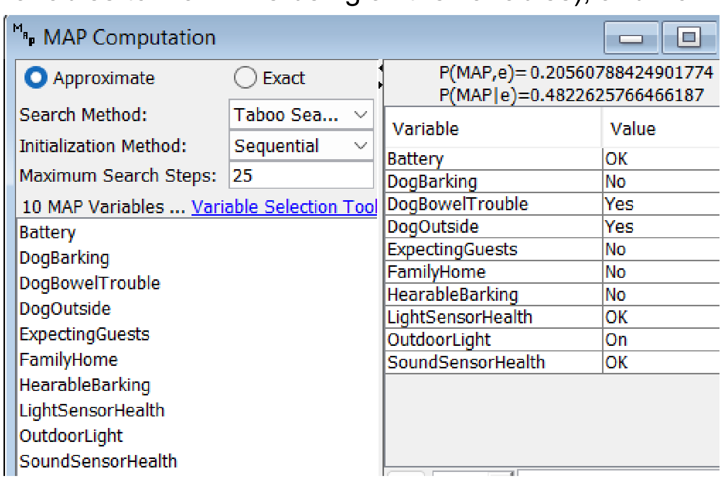
c) 
ExpectingGuests
13.60% - Yes
86.40% - No
ExpectingGuests ≅ No
FamilyHome ≅ No
FamilyHome
30.39% - Yes
69.61% - No
DogBowelTrouble
79.68% - Yes
20.32% - No
DogBowelTrouble ≅ Yes
DogOutside
87.61% - Yes
12.39% - No
OutdoorLight
81.32% - On
18.68% - Off
OutdoorLight ≅ On
DogOutside ≅ Yes
DogBarking
50.13% - Yes
49.87% - No
DogBarking ≅ Yes
HearableBarking
48.13% - Yes
51.87% - No
HearableBarking ≅ No
SoundSensorHealth
93.63% - OK
6.37% - Broken
SoundSensorHealth ≅ OK
LightSensorHealth
99.02% - OK
0.98% - Broken
LightSensorHealth ≅ OK
LightSensor
79.85% - On
20.15% - Off
LightSensor ≅ On
SoundSensor ≅ Off
SoundSensor
45.18% - On
54.82% - Off
Battery ≅ OK
Battery
99.09% - OK
0.91% - Dead

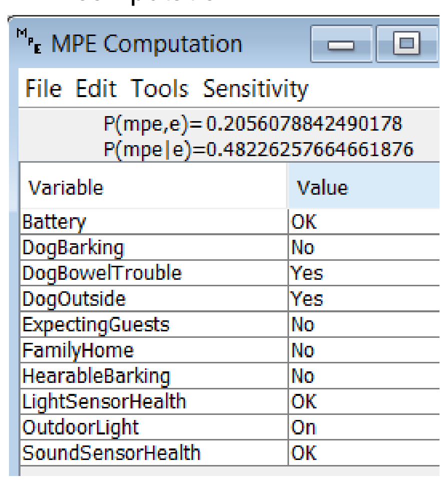
- The most likely instantiation of all variables given that Sambot has sensed the lights to be on, but has sensed no bark. Explain how you obtained this answer (for partial credit in case you get the wrong answer). Screenshot the results from SamIam and attach the pictures in the report.

To get the most likely instantiation of variables given Sambot has sensed the lights on but no bark, I set LightSensor to On and SoundSensor to Off



We can see the most likely instantiation above, from MAP computation (with the subset of variables to maximize being all the variables), and from MPE computation:

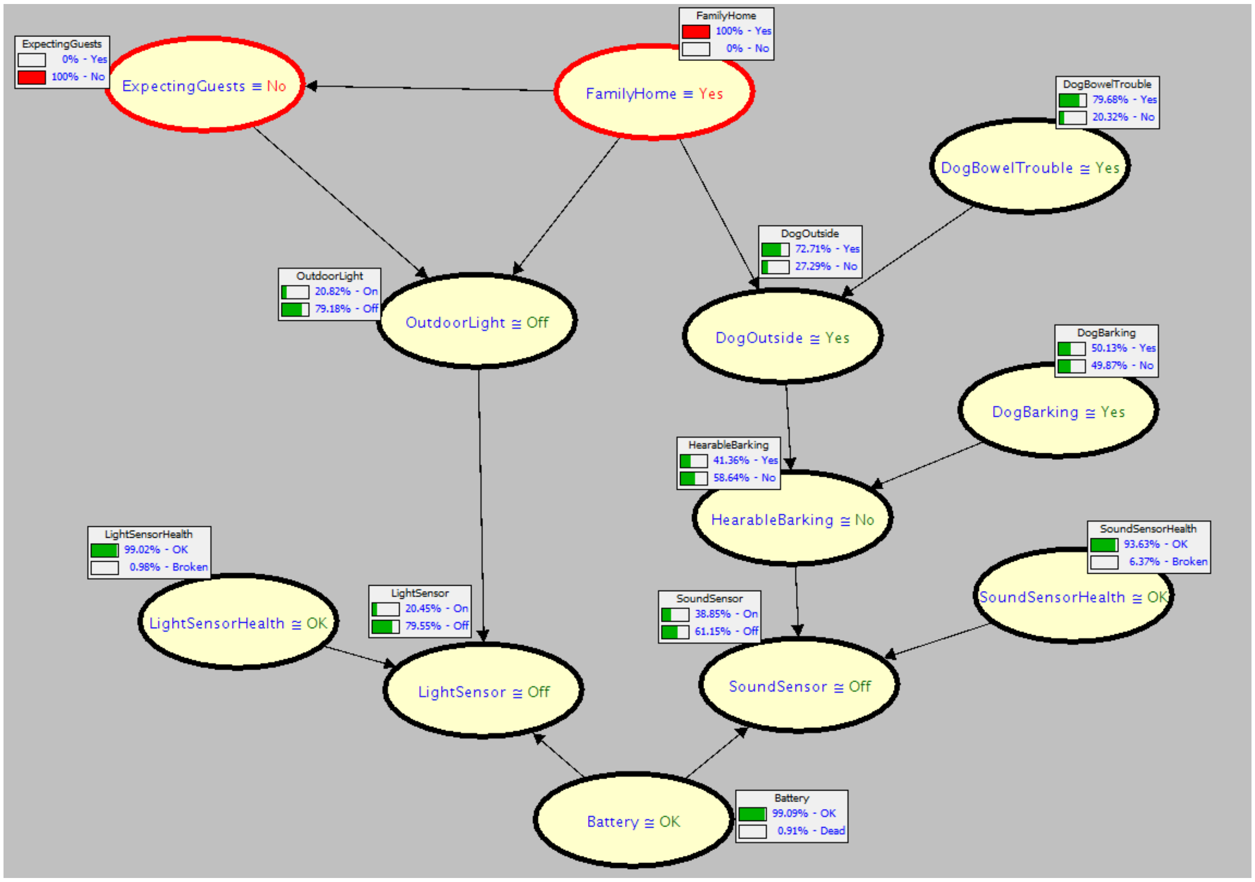




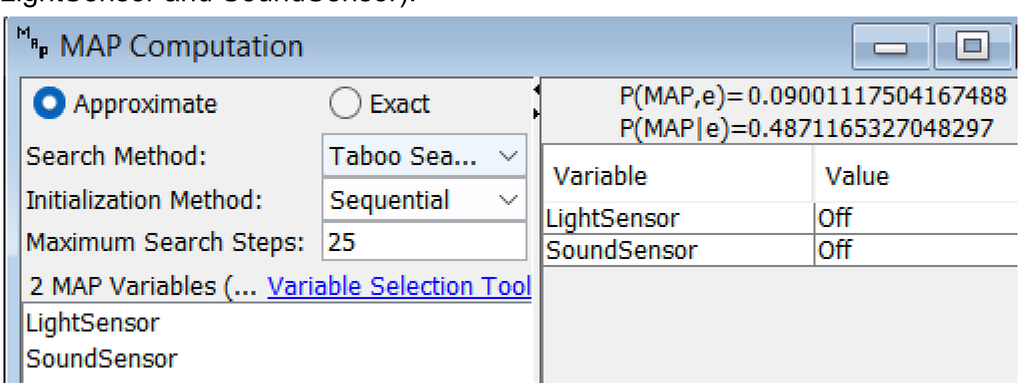
As we can see, we get the same result from both MAP and MPE.

- The most likely instantiation of the sensors given that the family is home and no guests are expected. Explain how you obtained this answer (for partial credit in case you get the wrong answer). Screenshot the results from SamIam and attach the pictures in the report.

To get the most likely instantiation of the sensors given that the family is home and no guests are expected, I set FamilyHome to Yes and ExpectedGuests to No.



We can see the most likely instantiation for the sensors from the monitors **(they are both Off)**, and from MAP computation (we use MAP since we are maximizing a subset of variables, LightSensor and SoundSensor):



As we can see, both sensors are Off.

- The smallest set of variables Z in your network such that the two sensors are independent given Z. Justify your answer based on d-separation.

The smallest set of variables Z such that LightSensor and SoundSensor are independent is Z = { Battery, DogOutside }.
All paths between LightSensor and SoundSensor are blocked (at least one value on each path is closed by Z)
LightSensor - Battery - SoundSensor path:
Battery is divergent and in Z, so it is closed, so this path is blocked
LightSensor - OutdoorLight - (ExpectedGuests -) FamilyHome - DogOutside - HearableBarking - SoundSensor path (ExpectedGuests can be included in the path, as both paths go through DogOutside):
DogOutside is sequential, in Z, so it is closed, so the path is blocked

Both paths are blocked, so d-sep(LightSensor, Battery DogOutside, SoundSensor) is true, so the two sensors are independent given Z = { Battery, DogOutside }.

- The type of network you constructed: tree, polytree (singly-connected network), or multiply-connected network.

My network is multiply-connected (it is a DAG). The underlying undirected graph is not a tree.