Tec de Monterrey Campus Querétaro

Arturo Rivera – A01202457

Cuauhtémoc Suárez – A01206503

Francisco Villanueva – A01202413

Lab 4

Artificial Intelligence

24/10/16

Lab 4: Implementing Bayesian Networks

Write a reflection or make a table where you compare Hugin Lite to your implementation. Within this, you must answer the following questions: What are the differences between what they generate? Do they use the same algorithms? What are their common bases? Which tool would you use for what cases in real life applications? (400 to 500 words)

Include the diagrams of the developed networks.

For comparing the way of working Bayes Networks between our implementation and Hugin Lite, the following network was implemented:

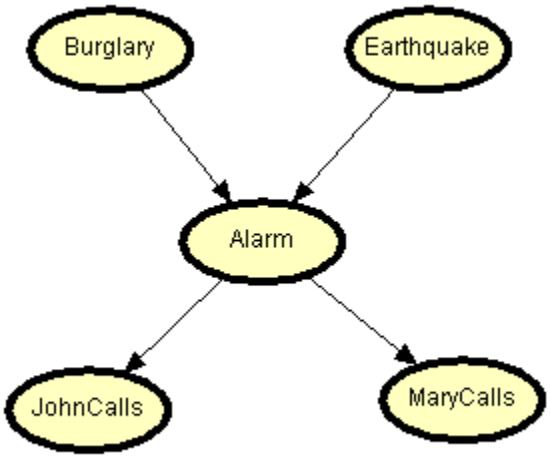

Figure 1.1 Selected Network

Moreover, the used probabilities are:

```
[Probabilities]
+Burglary = 0.001
+Earthquake = 0.002
```

```
+Alarm|+Earthquake,+Burglary = 0.95
+Alarm|-Earthquake, +Burglary = 0.94
+Alarm|+Earthquake, -Burglary = 0.29
+Alarm|-Earthquake, -Burglary = 0.001
+JohnCalls|+Alarm = 0.9
+JohnCalls|-Alarm = 0.05
+MaryCalls|+Alarm = 0.7
+MaryCalls|-Alarm = 0.01
```

Introducing these values to Hugin Lite will result in the following image:

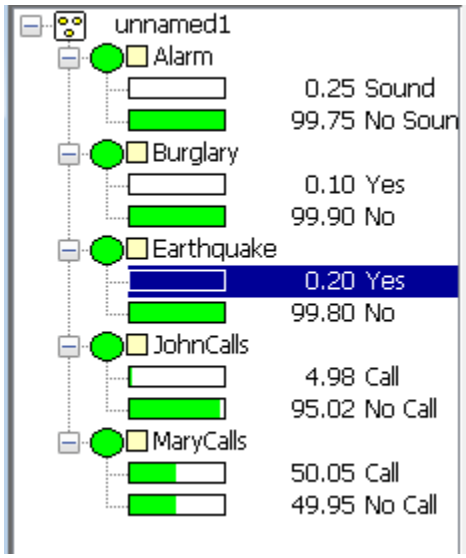

Figure 1.2 Total Probabilities

Hugin Lite gives directly the Total probabilities of each of the nodes as opposed to our implementation which needs to calculate them using the enumeration algorithm.

The used queries for comparing the results are:

```
[Queries]
+Burglary|+Earthquake, +JohnCalls
+Earthquake
-MaryCalls|+Earthquake, +Alarm
+JohnCalls|-Earthquake, -MaryCalls, +Burglary
```

```
-Alarm|-Earthquake, -MaryCalls, +Burglary

-Alarm, +JohnCalls|-Earthquake, -MaryCalls, +Burglary
```

By analyzing the way Hugin Lite calculates probabilities one cannot say the used method (enumeration, variable elimination, etc.) but, Hugin Lite gives directly the joint probabilities, e.g., P(+A, +B), total probabilities, e.g. P(+A), and conditional probabilities with only one query node and 1 to many evidence nodes, e.g. P(+A | +B, +C, -D). This implies that when having more than one query node the results are not as straightforward as one would expect; in these cases, the conditional probability should be converted to the division of two joint probabilities for obtaining the result requiring some middle steps.

For a better analysis of this, let's calculate the first query, which is (+Burglary|+Earthquake,+JohnCalls). Hugins is pretty straightforward as you only need to specify the evidence nodes, as follows:

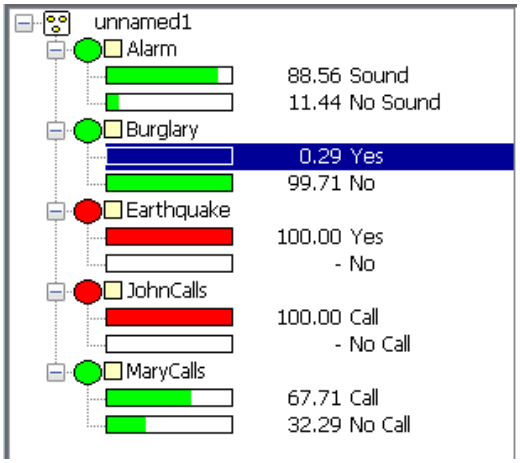

Figure 1.3 Hugin Lite Probability (+Burglary|+Earthquake,+JohnCalls)

In Figure 1.3, the result is shown in blue (0.29%), as opposed to the one calculated in the implemented code of 0.002866 so there is some rounding being done. For this kind of examples Hugin is quite simple, now lets try with (-Alarm, +JohnCalls|-Earthquake, -MaryCalls, +Burglary); here the query nodes are more than 1 so the probability should be calculated as follows (also the implemented code do this as it is the base for the enumeration algorithm):

$$P(-A,+J|-E,-M,+B) = \frac{P(-A,+J,-E,-M,+B)}{P(-E,-M,+B)}$$

Equation 1.1

Thus, both joint probabilities are gathered from Hugin Lite like this:

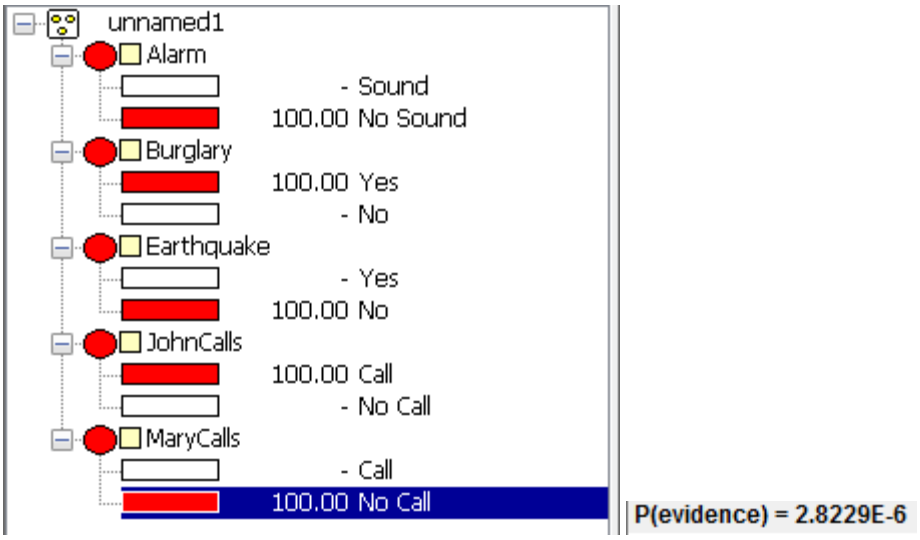

Figure 1.4 Hugin Lite Probability P(-A, +B, -E, +J, -M)

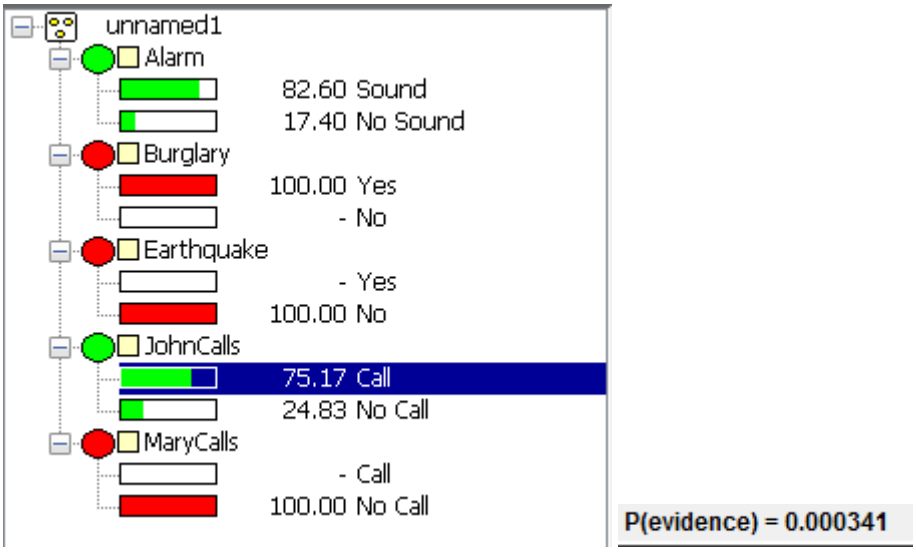

Figure 1.5 Hugin Lite Probability P(+B, -E, -M)

Takingin account figures 1.4 and 1.5, the complet probability is:

$$P(-A,+J|+E,-M,+B) = \frac{P(-A,+J,+E,-M,+B)}{P(+E,-M,+B)} = \frac{2.8229E-6}{0.000341} = 0.008278$$

As opposed to the result of the code being 0.0086995.

Taking everything into account, as the base probabilities of the Hugin Lite software are the joint probabilities, it would wise saying that they use a variable elimination algorithm for calculating the probabilities. Furthermore, the rounding made to every joint probability could be a problem when computing conditional ones with multiple query nodes as you are losing accuracy with each iteration so in these cases using Hugin Lite is not as good as the code we implemented.

Finally, Hugin Lite is perfect for the previously described situations (conditional probabilities with only one query node, joint probabilities and total probabilities) as calculating them is as direct as it could get, you only specify your evidence nodes and “voilá” you get the result; for every other situation, using the implemented code is better.