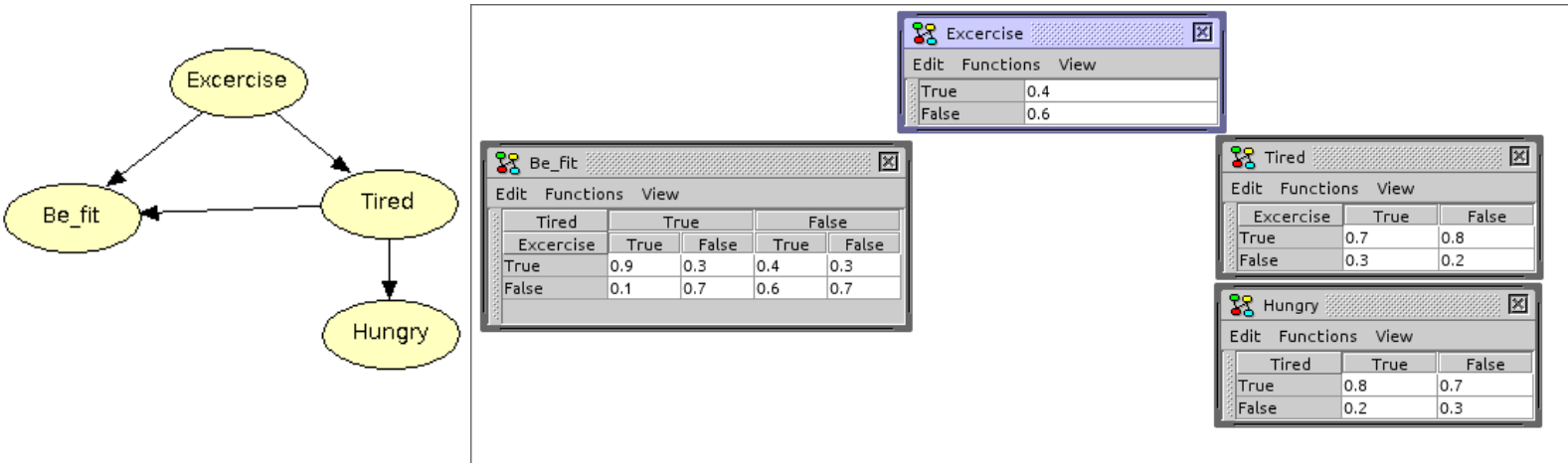
Bayes Network Lab Report

Moisés Montaño Copca
A01271656

This is my example network, with its respective probability tables:



My queries are:

-Hungry
-Be_fit|+Hungry,+Tired
+Excercise|-Hungry,-Be_fit
+Tired|-Hungry,+Be_fit,+Excercise

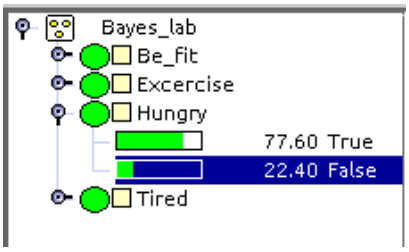
The results (in order):

From my program:

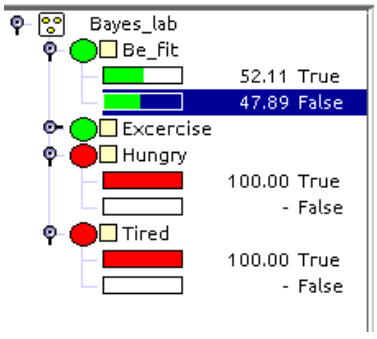
```
0.224
0.4789474
0.2274247
0.7777778
```

From Hugin:

-Hungry

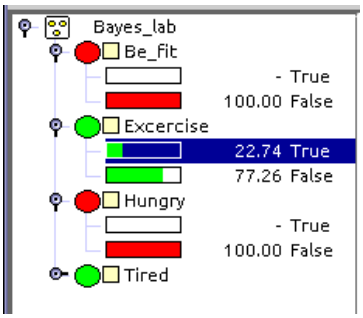


-Be_fit|+Hungry,+Tired

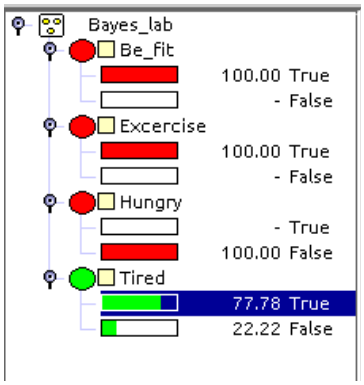


More from hugin:

+Excercise|-Hungry,-Be_fit



+Tired|-Hungry,+Be_fit,+Excercise



Questions

1. What are the differences between what they generate?

Both gave the same results, with the tiny difference in the rounding of values, and Hugin giving it from 0 to 100 instead of 0 to 1.
Another difference is that Hugin computes all the probabilities at once, that's why we see both the *true* and *false* values in the images, and it also computes the probabilities of all the other nodes, so there is not a specific query by itself.

2. Do they use the same algorithms?

For the whole Bayesian Network probabilities and 'queries' they do, however they are implementing the algorithms in different ways. Hugin is computing all the probabilities for each evidence we include, while my program is computing only the specified probability from zero every time, e.g. if the queries are +something and -something, my program will compute each of them by separate instead of just assuming that they have to sum 1. While Hugin will compute both of them and all the other nodes in the Network.

3. What are their common bases?

They share theoretical bases to compute probabilities inside a Bayesian Network, like the total probability theorem, chain rule and the Bayes theorem, however as I mentioned earlier, their implementations are different. That means they also share some probability theory and math theory, but I think Hugin may include more advanced theorems regarding probability, Bayesian models and influence diagrams.

4. Which tool would you use for what cases in real life applications?

In both of them, declaring the network (for the input) may be very time consuming, since it has to be node by node and then do the probability distribution tables, so regarding input I

consider both of them equal. Now, I think that Hugin may use a lot of unnecessary resources on a really big network, since it is computing all the probabilities at once, while my program only computes the queries that were specified. The fact that Hugin computes all the probabilities at once may also be an advantage if that's what we want, to see more diversity. Also Hugin is more complete and easier to understand, since it includes a graphic interface and we can 'adjust' the queries with just two clicks, so it is also easier to use. So for cases where we want only a specific probability, maybe to compute the probability of a natural disaster happening or the probability of someone having a disease I would use the program I developed in this lab, however, in cases where I would like to analyze the causes, effects and relations between certain events I would use Hugin (where there is not a specific query defined).