Daniel Adalaid

Mayumi Luque

Report

Our Bayesian Network is based on the favorite activities of every average student, attending class and doing homework. The network looks something like Figure 1.

Being Sick affects our chances of doing Homework, as well as our chances at attending Class.

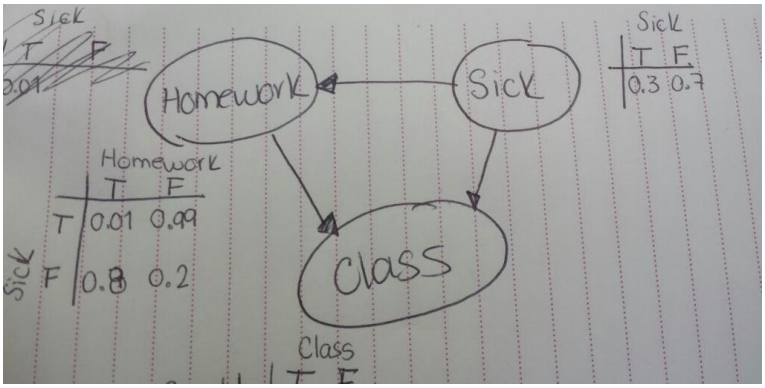

Figure 1. Please don't judge the failed attempts

As you can see in Figure 2, if we are sick and we haven't done our homework, there are zero chances we will put a foot in class. This principle is held for the opposite. If we aren't sick and we've done our homework, you can fill that checkbox while taking roll call.

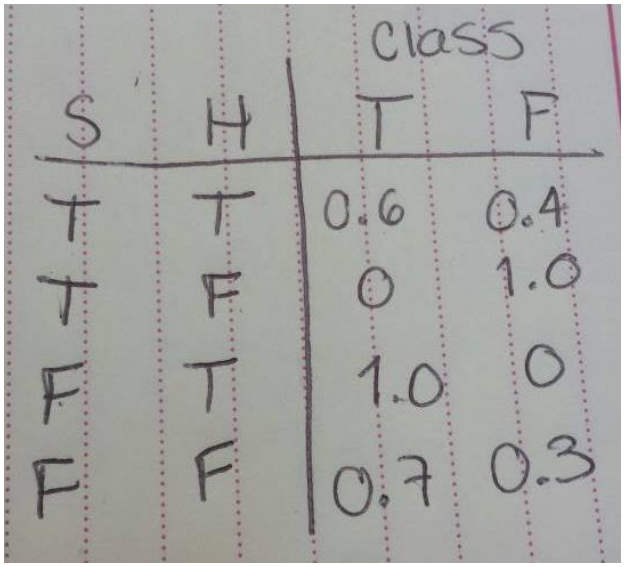

Figure 2.

We built the Bayesian Network in Hugin, and it turned out like Figure 3.

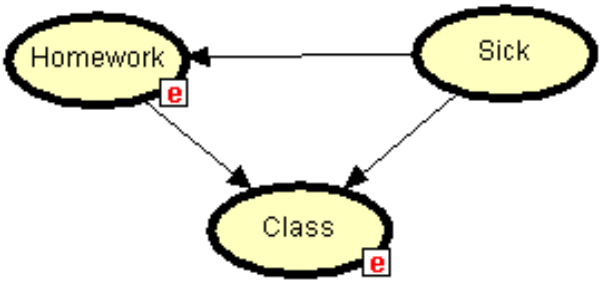

Figure 3.

```
1  [Nodes]
2  Sick, Homework, Class
3
4  [Probabilities]
5  +Sick = 0.3
6  +Homework|+Sick=0.01
7  +Homework|-Sick=0.8
8  +Class|+Sick, +Homework= 0.6
9  +Class|+Sick, -Homework= 0
10 +Class|-Sick, +Homework= 1
11 +Class|-Sick, -Homework= 0.7
12
13 [Queries]
14 +Sick| -Class, -Homework
15 -Sick| -Class
16
```

Figure 4.

The queries we did in both programs are in Figure 4.

The results that our program and Hugin threw were a little different in the decimal sense. For the first query we got a 0.87611 in our program and Hugin gave us 0.85. On the other one we had a 0.1234 and Hugin gave us 0.13. We attribute these little differences to the accuracy of the results and rounding up and down.

Given that the results differ, and that the interfaces are different, we can conclude that the algorithms used could be different. At least some of them. For example, the algorithms used to build the Bayes Networks. Since Hugin has a graphic interface and we construct the network based on the probabilities, the algorithms are bound to be different. However, the algorithms for the calculations made for the chain rule and probabilities are bound to be, at the very least, similar, since these are established formulas.

If we are talking about the application that I would use in real life, I would definitely go for Hugin. Why? Because even though it has a less than desirable UI, (we had to search how to use it) it has a friendlier interface than our own program. The nodes and relationships are easy to establish, and it automatically gives you the queries you ask it. On the other hand, we have our program that needs a very specific way of entering the information, and the relationships are hard to understand and follow. Hugin also has the feature to do as many queries as you want, and it will answer them in real time. Plus, it has a graphic interface, which is always a better for commercial use. However, if I were to need something that gives me a more specific result, I'd go for our program. Not because I think of us as better programmers than the makers of Hugin, but because I'd be able to edit the program as I see fit.

For real world use, though, I would still go for Hugin.