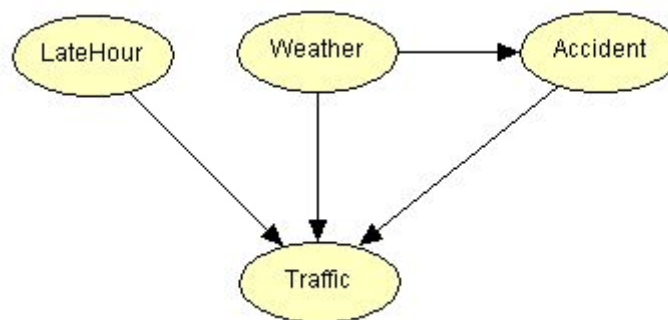


## Bayes Networks

### Comparison Hugin Lite vs Our Implementation

Example Network:

We wanted to model the probability of us to be stuck on a traffic jam in the mornings when we head to school, for this we will consider the LateHour node as the probability of us to wake up on time, Weather node as the probability of a bad weather, Accident node as the probability of an accident happened and Traffic node as the probability of us being stuck at the traffic thus arrive late to class.



In this hypothetical world, the probabilities would be the following:

+L	0.4
-L	0.6

+W	0.1
-W	0.9

Weather	+A	-A
+W	0.8	0.2
-W	0.3	0.7

			Traffic	
LateHour	Weather	Accident	+T	-T
+L	+W	+A	0.99	0.01
+L	+W	-A	0.95	0.05
+L	-W	+A	0.97	0.03
+L	-W	-A	0.6	0.4
-L	+W	+A	0.9	0.1
-L	+W	-A	0.4	0.6
-L	-W	+A	0.8	0.2
-L	-W	-A	0.1	0.9

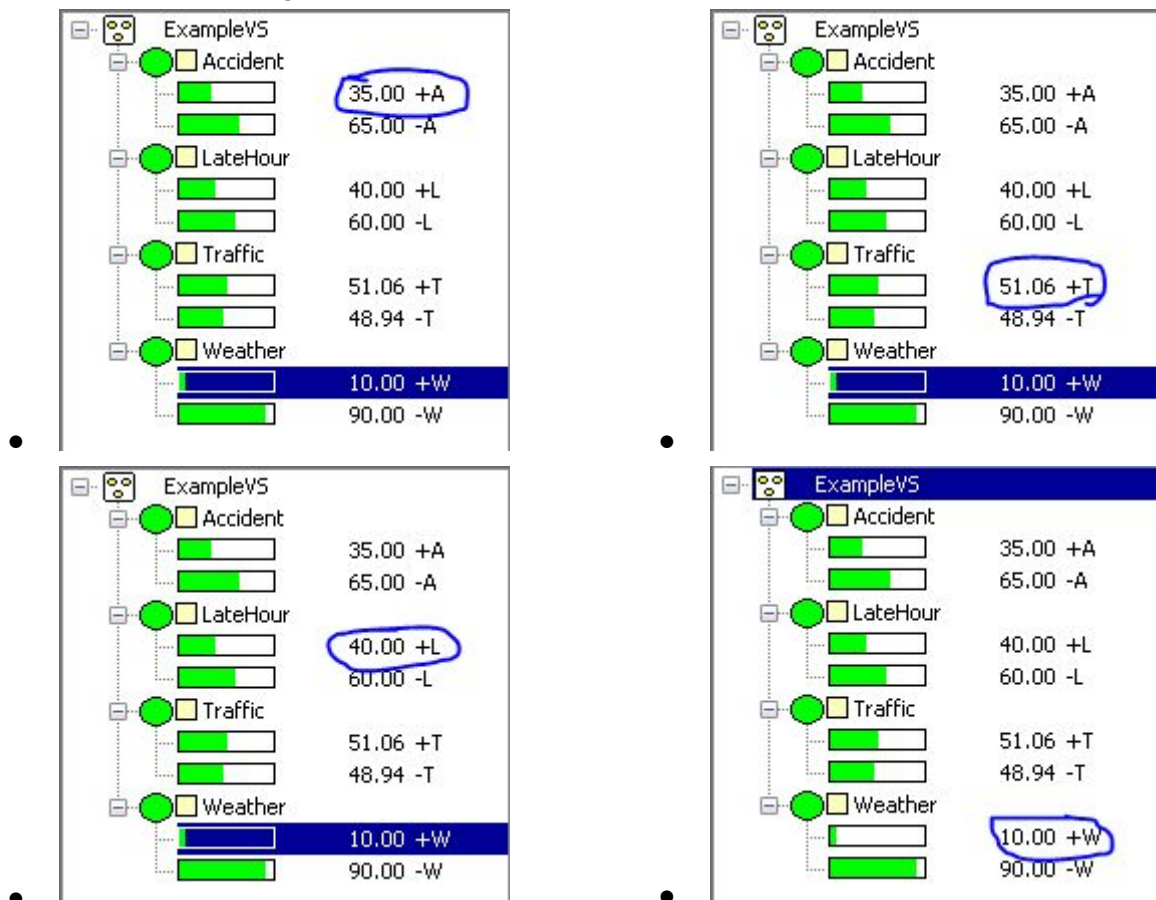
## Queries

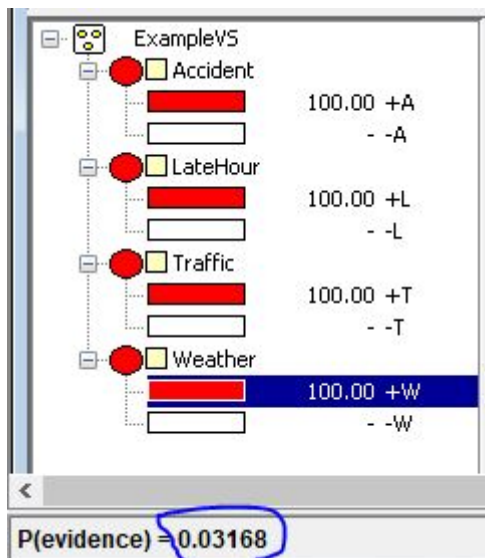
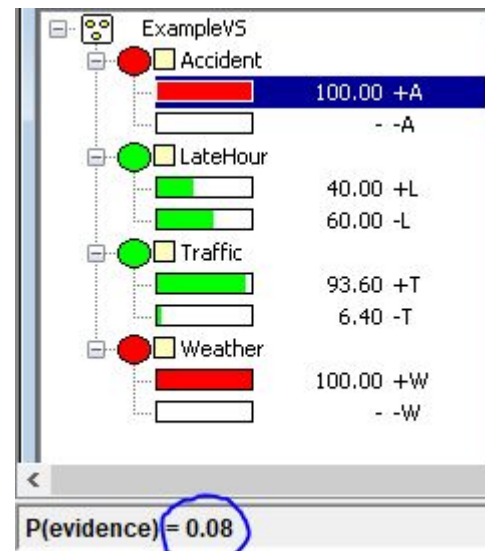
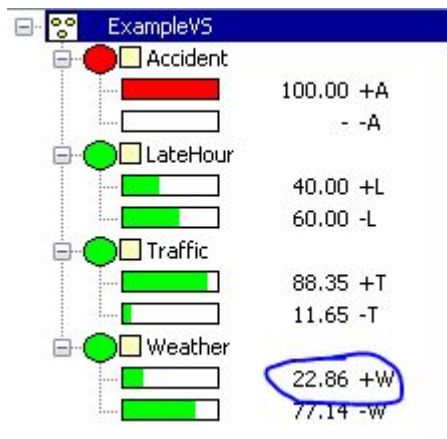
- +Accident
- +LateHour
- +Traffic
- +Weather
- +Weather|+Accident
- +Accident,+LateHour,+Traffic,+Weather
- +Accident,+Weather

On the same order in our implementation:

```
0.35
0.4
0.51064
0.1
0.2285714
0.03168
0.08
```

On the same order in Hugin





## What are the differences between what they generate?

Both, our program and Hugin gave us the same results, the difference was the way to show the operations, Hugin shows them in the range from 0 to 100 and our implementation from 0 to 1; also when we run the Bayes network on Hugin it computes the sum-propagation, that's why we only see the probability values of the states like True, False, + or -, giving us the ability to trigger the nodes that we want to make a "query".

## Do they use the same algorithms?

Yes and no, we both use the same theory of Bayesian Network behind the calculus like chain rule, enumeration algorithm, Bayes theorem and total probability but Hugin computes all the probabilities for each node assuming that they have to sum 1 so it retrieves all the nodes in the Network at once, on the other hand our program computes the probability one by one based on the query to analyze calculating the join probability of each related node by applying the enumeration algorithm.

## What are their common bases?

As we said earlier, we use the same theory behind but the implementation of them are different, it's obvious that Hugin is a program with more features so it may include more advance theorems or formulas of probability to calculate other things like decisions policies among others.

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

Both have its advantages and disadvantages, the fact that Hugin computes all probabilities at once may sound time and resource consuming but that's where its robustness come from, you can analyze the full behavior of the Bayes network and with its graphic interface the tool is not only relatively easy to use and understand but it gives you the ability to change in real time the "queries" just by triggering the states of the nodes, the problem we see is that, imagine a really big Bayes Network, it will represent a heavy load for the computer.

The main difference with our program is that it computes just the probabilities that it needs to test the queries given as input, thanks to the enumeration algorithm we calculate the joint probabilities, but since the algorithm used to compute the distribution is a brute-force algorithm, the complexity may vary from query to query and of course with its lack of graphical interface, you cannot see the full behavior of the Bayes Network and also the way it process input may be difficult to recreate for other Bayesian Networks, but for educational purposes and effect for this lab its perfect as is.

In real cases where you need to analyze and understand the behavior of the belief propagation, the causes and effect, correlation and all that stuff between events, we think it is better to use Hugin because you can build a probabilistic graphical model easy to understand, it is unlikely that you will retrieve all of that data with our program (in an easy way lick double click), however, our program is more suitable when the user's needs are only computing a specific probability like what's the probability of a natural disaster to happen or what's the probability of someone to have flu.

Obviously to use one or the other would depend on the scenario but if you wanted, you could even modify our code and add other functions to it, maybe with matplotlib or TKinter add a user interface and see each step for the enumeration algorithm.