Alejandro Rosas Ceballos A00231795
Christopher Ojeda Rivera A01203980

Lab: Bayes 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.

Version used: Python 3.5.2 :: Anaconda 4.1.1 64bit

Hugin 8.4

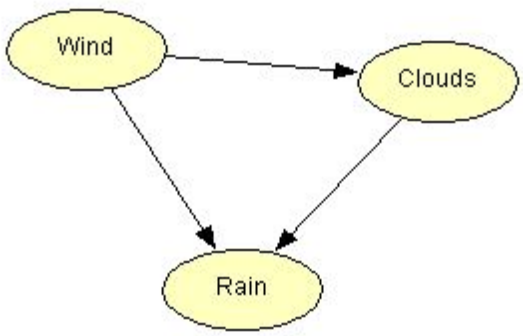
Test case:

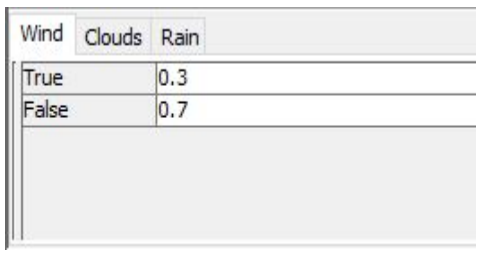
[Nodes]
Clouds, Wind, Rain

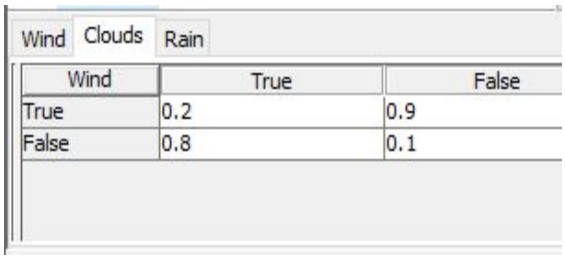
[Probabilities]
+Wind = 0.3
+Clouds|-Wind = 0.9
+Clouds|+Wind = 0.2
+Rain|+Wind,+Clouds=0.7
+Rain|+Wind,-Clouds=0.2
+Rain|-Wind,+Clouds=0.9
+Rain|-Wind,-Clouds=0.0

[Queries]
+Rain|-Wind
+Wind|+Rain
+Wind,+Clouds|+Rain

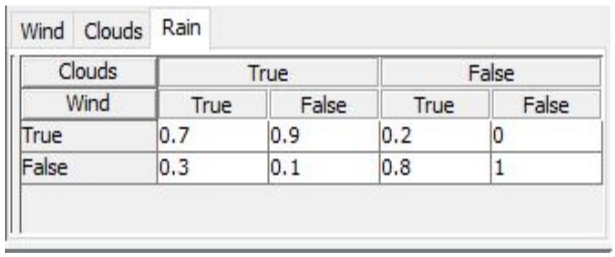
Hugin Lite diagram:



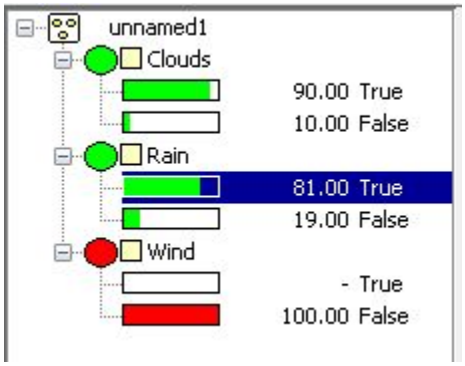


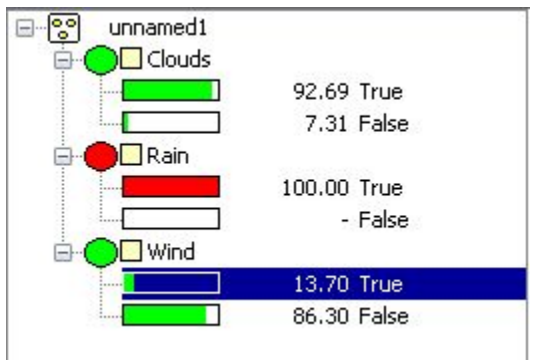


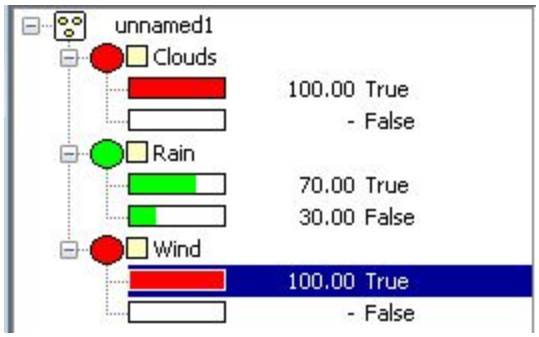
Wind	True	False
True	0.2	0.9
False	0.8	0.1

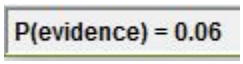


Clouds	True		False	
Wind	True	False	True	False
True	0.7	0.9	0.2	0
False	0.3	0.1	0.8	1









Script output:

```
[Nodes]
Clouds, Wind, Rain

[Probabilities]
+Wind = 0.3
+Clouds|-Wind = 0.9
+Clouds|+Wind = 0.2
+Rain|+Wind,+Clouds=0.7
+Rain|+Wind,-Clouds=0.2
+Rain|-Wind,+Clouds=0.9
+Rain|-Wind,-Clouds=0.0

[Queries]
+Rain|-Wind
+Wind|+Rain
+Wind,+Clouds|+Rain
^D
0.81
0.1369863
0.0639269
```

Comparison:

Version used: Python 3.5.2 :: Anaconda 4.1.1 64bit

Hugin 8.4

Characteristic	GUI	Easy to understand	Easy to use	Suitable for real life applications	Shows network values	Control over the functionality
Hugin Lite	Yes	Yes	With training	Yes	Yes	No
Python script	No	No	With correct syntax	No	No	Yes

The purpose of this lab was to understand and create a bayes network while Hugin Lite is a program made from various components that make it able to process information in more possible ways than the python script made for the lab.
The differences are various being the most notorious the GUI which helps the user to understand what is happening and also lets him see the overall information about the network, including more tools that help to investigate the status after changing the presence of evidence.
Both solutions use the bayes rules, total probability and enumeration. But they differ in the implementation. In theory the algorithms can be the same because of the common base, but since they have different behaviours and purposes the actual algorithms differ, since there are different ways of transforming an algorithm to code.
For example, it can be seen that Hugin Lite creates the values for the whole network and they change once an action is detected, in the python script the exit values are just the ones that the user asked for and anything else is hidden from the user, including the actual values of the network.
Also the Hugin Lite is used in real life applications for decision making because the main purpose of the software is to exploit the advantages of a bayes network to help solve problems and take decisions. In contrast the script made for the lab has the main purpose of understanding and creating a bayes network, which reduces the overall possibilities of applying it to a real solution.
Taking this into consideration we would use Hugin Lite for a real life application, but given the circumstances of not being able to purchase a license or that the provided features are more than what is needed, it would be better to create a custom implementation that represents a bayes network and write only the features that are needed.
Hugin lite while being a very useful tool, it still manages to be easy to use, its GUI keeps it very simple to learn for any user that already has some programming skills, it is helpful in various business areas.
At the end we learned how to program different methods to solve Bayesian networks and we also learned about a useful application, which is Hugin lite and it might come in handy in the future by solving some decision making problems.