

Lab 4: Bayes Networks

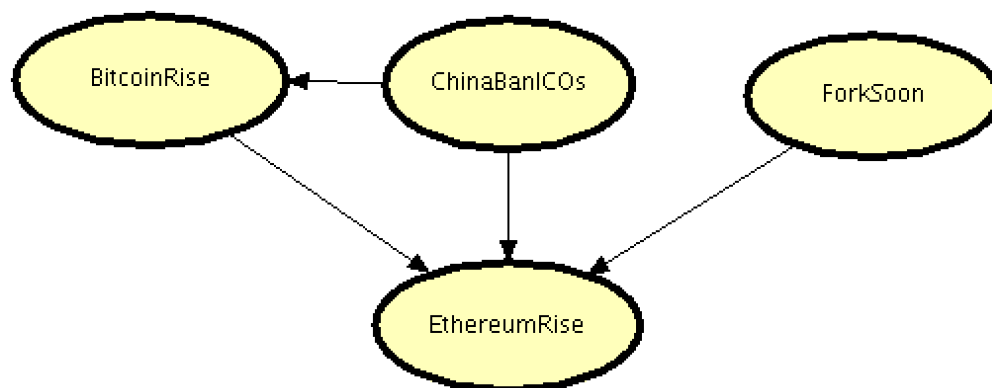
Artificial Intelligence

Eduardo Vaca

A01207563

In this lab we had to implement a program capable of creating a Bayesian Network given a set of nodes with their probability tables for a later querying.

After implementing we had to come up with our own Bayesian Network, for which I decided to create one simple representation of the rise of crypto currency Ethereum with some of the variables that influence in this event. The diagram is as follows:



The probability tables of the network are represented in the following statements:

+ForkSoon=0.07
+ChinaBanICOs=0.65
+BitcoinRise|+ChinaBanICOs=0.05
+BitcoinRise|-ChinaBanICOs=0.35
+EthereumRise|+BitcoinRise,+ChinaBanICOs,+ForkSoon=0.64
+EthereumRise|+BitcoinRise,-ChinaBanICOs,+ForkSoon=0.82
+EthereumRise|-BitcoinRise,+ChinaBanICOs,+ForkSoon=0.27
+EthereumRise|+BitcoinRise,+ChinaBanICOs,-ForkSoon=0.4
+EthereumRise|-BitcoinRise,-ChinaBanICOs,+ForkSoon=0.4

$+EthereumRise|-BitcoinRise,+ChinaBanICOs,-ForkSoon=0.03$
 $+EthereumRise|+BitcoinRise,-ChinaBanICOs,-ForkSoon=0.57$
 $+EthereumRise|-BitcoinRise,-ChinaBanICOs,-ForkSoon=0.2$

We had to represent this network in our program and also in a tool called Hugin. In both we had to make some queries and compare the results. The following table represents the queries with the results:

Query	My Program	Hugin Tool
+EthereumRise	0.1630987	16.31%
+EthereumRise +ForkSoon	0.378975	37.90%
+EthereumRise +BitcoinRise,+ForkSoon	0.7822581	78.23%
+ForkSoon +EthereumRise	0.1626515	16.27%
+ChinaBanICOs -EthereumRise,-BitcoinRise	0.7669908	76.70%
+ChinaBanICOs	0.65	65%
-EthereumRise +ChinaBanICOs	0.9347	93.47%
-EthereumRise +BitcoinRise	0.4482919	44.83%

What are the differences between what they generate?

The results are the same.

Do they use the same algorithms?

I implemented my program using Bayes Networks, specifically the *Enumeration* algorithm. I dug through the website of the Hugin tool and as expected they didn't document much about their implementation but they inform that they use Bayesian Networks as well:

"...HUGIN software is based on Bayesian networks and influence diagram technology, an advanced artificial intelligence technique widely used for supporting decision-making under uncertainty."

What are their common bases?

Both use Bayesian Networks. However, Hugin website mentions in its description that they use also *"influence diagram technology"* which from Google's definition is: *an intuitive visual display of a decision problem*. So I'm pretty sure is just for advertising purposes cause it's basically the definition of a Bayes Network.

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

I really like my program because of its easy use. Basically you can run it through a command line and the input format is very generic, with a Python script you could easily transform some data to the input format expected. However, the Hugin tool has had support for more than two years which means they have an engineer team totally focused on improving performance and general aspects of the tool. Thus, I would use my program for school projects or simple subjects that would be interesting to model and obtain real probabilities. Hugin tool I would firstly use it for education purposes due to its very visual UI and also for business purposes, it would be very easy and elegant to present a project to a company board with that tool, and in engineering aspects it should be very efficient on memory and speed. My program for example wouldn't hold a network for predicting diseases, the number of nodes involved must be gigantic, so also for this kind of big problems, involving thousands of variables in the network, I believe Hugin is the correct choice. The only disadvantage is that you would likely need to pay to obtain the whole functionality.