Instituto Tecnológico y de Estudios Superiores de Monterrey Campus Querétaro

Implementing Bayesian Networks

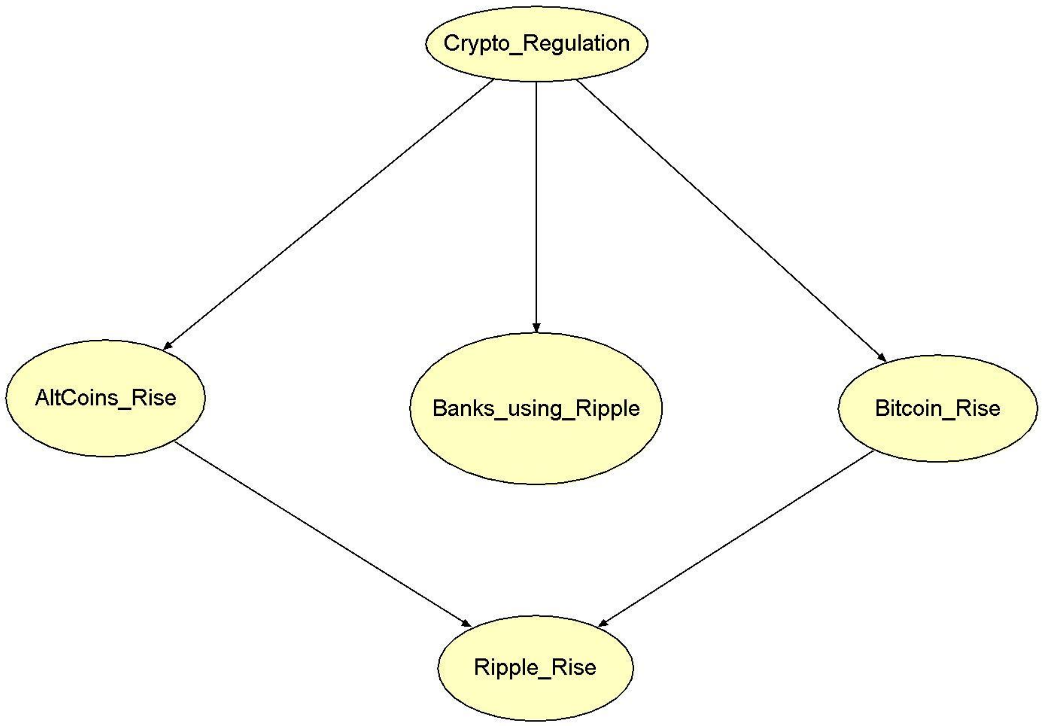
Laboratory Report

Authors:
Christian Ricardo Solís Cortés A01063685
Raúl Mar A00512318
Professor: Ruben Stranders

Artificial Intelligence

Santiago de Querétaro, Querétaro, México. Delivery Date: March 16th, 2018

Report: Ripple Rise Bayesian Network



P(+Crypto_Regulation) = 0.75

P(+Alt_Coins_Rise | +Crypto_Regulation) = 0.70

P(+Alt_Coins_Rise | -Crypto_Regulation) = 0.30

P(-Banks_Using_Ripple | -Crypto_Regulation) = 0.65

P(+Banks_Using_Ripple | +Crypto_Regulation) = 0.75

P(+Bitcoin_Rise | -Crypto_Regulation) = 0.30

P(-Bitcoin_Rise| +Crypto_Regulation) = 0.80

P(+Ripple_Rise | +Alt_Coins_Rise, +Bitcoin_Rise) = 0.85

P(+Ripple_Rise | +Alt_Coins_Rise, -Bitcoin_Rise) = 0.60

P(-Ripple_Rise | -Alt_Coins_Rise, +Bitcoin_Rise) = 0.25

P(-Ripple_Rise | -Alt_Coins_Rise, -Bitcoin_Rise) = 0.80

We represented this network in our program and also in the tool called Hugin. For both cases we made some queries to compare the results. This is the table representing the results:

Query	Hugin Tool Results
P(+Ripple_Rise | +Crypto_Regulation)	74.53%
P(+Ripple_Rise | +Alt_Coins_Rise, +Bitcoin_Rise, +Banks_Using_Ripple, +Crypto_Regulation)	85%
P(+Ripple_Rise | -Alt_Coins_Rise, -Bitcoin_Rise, -Banks_Using_Ripple, -Crypto_Regulation)	20%
P(+Ripple_Rise | -Banks_Using_Ripple)	60%
P(+Alt_Coins_Rise | -Banks_Using_Ripple, -Crypto_Regulation, -Bitcoin_Rise)	30%
P(+Ripple_Rise | -Banks_Using_Ripple, -Crypto_Regulation, -Bitcoin_Rise)	32%
P(+Crypto_Regulation | +Alt_Coins_Rise, +Bitcoin_Rise, +Banks_Using_Ripple, +Ripple_Rise)	97.56%
P(+Ripple_Rise | +Bitcoin_Rise)	80.60%

1. ***What are the differences between what they generate?***

The results are expected to be the same (and they are).

2. ***Do they use the same algorithms?***

No, the algorithm that the bayes networks that can be created with the Hugin tool is actually implementing two algorithms for structural learning, the PC & NPC. This are the main steps:

1. Test for (conditional) independence between each pair of variables.
2. Identify the skeleton of the graph induced by the derived CIDRs.
3. Identify colliders.
4. Identify derived directions.

The Hugin Tool for Learning Bayesian Networks (PDF Download Available). Available from: https://www.researchgate.net/publication/220907806_The_Hugin_Tool_for_Learning_Bayesian_ Networks [accessed Mar 16 2018].

3. *What are their common bases?*

The primarily base is of course the use of Bayesian Networks, but the software also mentions the use of "Influence diagram technology". Searching online about the technology, and according to Google: "Influence diagram technology is a compact graphical and mathematical representation of a decision situation. It is a generalization of a Bayesian network.". This definition, tells us that the influence diagram technology is technically a Bayesian Network.

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

We would use the Hugin tool for any problem that can be solved using bayesian inference. An example of where can we use a bayes network is for example if a bank wants to know if they should give a certain loan to a person. This a a complex problem that can be solved efficiently with a bayes network.

We were not able to finish the lab, so we could not compare the results with our own program, but we used a one from a library to compare the results. We will still try to finish the lab but we fully understand that we failed for this lab.

References

Madsen, Anders & Lang, Michael & B. Kjærulff, Uffe & Jensen, Frank. (2003). The Hugin Tool for Learning Bayesian Networks. Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science). 2711. 594-605. 10.1007/978-3-540-45062-7_49.