T6 Bayesian Networks Practical - Genie

In this practical you will learn to use some features of the software package *Genie* to build a basic Bayesian Belief Network.

SMILE (Structural Modeling, Inference, and Learning Engine) is a fully portable library of C++ classes implementing graphical decision-theoretic methods, such as Bayesian net-works and influence diagrams, directly amenable to inclusion in intelligent systems. Its Windows user interface, GeNIe is a versatile and user-friendly development environment for graphical decision-theoretic models. Both modules, developed at the Decision Systems Laboratory, University of Pittsburgh, have been made available to the community in July 1998 and have now several thousand users worldwide. [from http://genie.sis.pitt.edu/].

This practical basically consists of a number of exercises about building some BBNs from some data and playing with them. Next, you may wish to further explore the program to build and work with more complex BBNs.

Please note that the basic use of Genie is very simple. The installed application has an Examples folder you may wish to autonomously explore after the practical. Please, remember that you can not save your examples or anything in the same example folder, as it is part of the installation and is local to every computer. You should save your BBNs in a suitable folder in your home space, under the H drive.

Running Genie – Launch the application from the *All Programs* menu under the *Start* button. Follow the explanation by the demonstrator that is being projected in the screen. Build the same examples.

Example 1: Asia. This example has been discussed in lectures and the related BBN is provided in the example folder of the Genie distribution.

Example 2: Glandular Fever. Build a BBN for the complete example about the Glandular Fever seen in lectures (you can find it in the lecture notes). Once the BBN is ready, use it to answer all the questions we have answered to in lectures by making all the required calculations by hands. Check you are obtaining the same results.

Some basic hand made networks

Example 3. Build a two node network that models the probability of picking a card of each suit. The first node represents the first card, the second node represents the second card. Link the two so that the second node models the probabilities of picking each suit given the suit already seen in the first.

Create two nodes called Card1 and Card2. Add four values for Heart, Spade, Club and Diamond. It’s quicker to make the node for Card1 and then cut and paste to make the second node. You add values in the window that you get when you double-click the node, as seen.

For Card1, put the expected probabilities in the table you get when you right click the node, as seen.

For Card2, enter the conditional probabilities in the table provided. What is the probability of a suit if the first card was of that same suit? And the probability of getting a suit if the first card was of a different suit? Make sure you understand why before checking the solution * at the bottom of the page!

Fill in the table and compile the network.

Calculate the following probabilities:

- Heart in card 2 given card 1 is a heart
- Spade in card 2 given card 1 is a heart
- If card 2 is a heart, what are the respective probabilities that card 1 was a heart, spade, club or diamond?

Example 4. Now imagine you take 10 cards from a deck:

- Ace, two and three of spades (black)
- King and queen of diamonds (red)
- Four, five, king, ace of clubs (black)
- Five of hearts (red)

You shuffle the cards. Based on those cards only, build a network with the following nodes:

- Colour of first card
- Value of first card
- Suit of first card

Consider the different ways you might join these three nodes and try a few of them. Try a chain from value to colour to suit, then try putting value and colour converging into suit. Be careful because some combinations might not be possible…

*The probabilities are respectively 12/51 = 0.2353 and 13/51 = 0.2549.