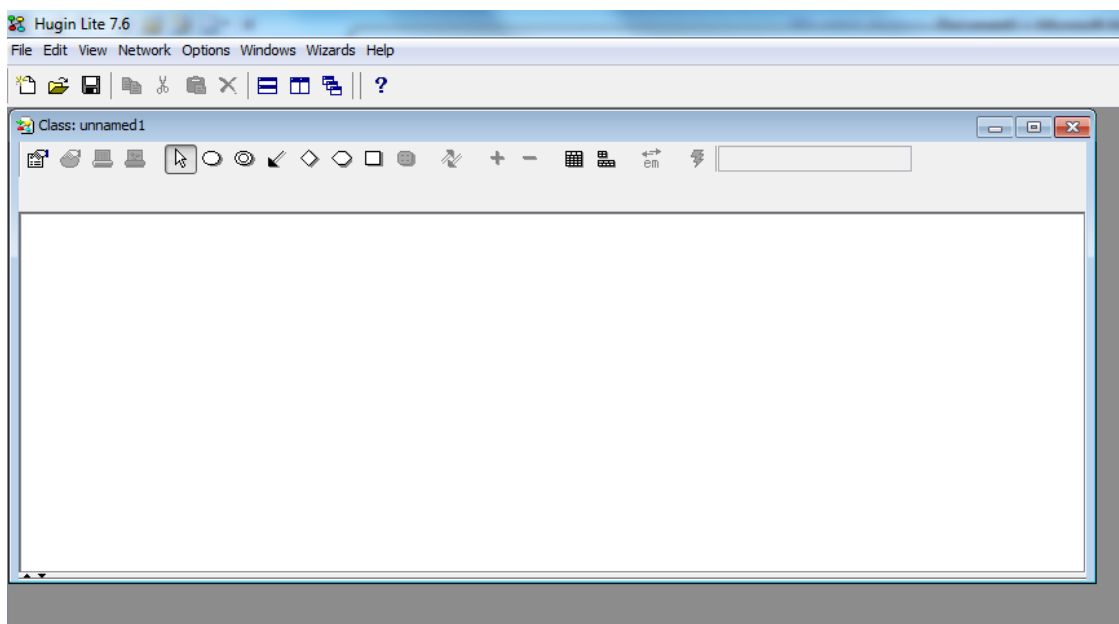

INTRODUCTION TO HUGIN SOFTWARE (WITH EXERCISES)

Laura Uusitalo 2006-2019, with input from Susanna Jernberg 2019

This is a tutorial designed to walk you through the first steps of using the Hugin software and build your first models. The blue boxes tell you what to do in the Hugin software.

GETTING STARTED:

Open Hugin software. It should look like this:



In the model window inside the program (the one titled “Class: unnamed1” in the figure), you have the tools to create a network: different node (variable) types, arcs (arrows) to link them, etc. If you hover the mouse in top of them, you’ll get a tool tip telling you what each tool does.



First of all, there are two “modes” in Hugin: editing mode and inference mode. You can get from edit to inference mode by clicking the lightning bolt, and from inference to edit mode by clicking the pencil icon.

In edit mode, you can create and change the model; in inference mode you can run the model and see how it behaves.

“BASIC” DISCRETE VARIABLE

Let's start with the discrete variable. You can create it by clicking the simple oval in the tool ribbon, and then clicking anywhere on the white space on your model window.

TRY:

Click on the discrete chance node (the simple oval), then click anywhere on the white space in the window. You should get a discrete variable node.

The possible values such a variable can have come from a discrete set, i.e. separate, distinct values. They can be numbers such as 2, 4, 6, 8, ..., colours, species, seasons... In Hugin there are four different discrete **node types**:

- Labelled
 - each state has a label, and the labels don't (necessarily) have any numerical interpretation
 - “Fully”; “Partially”; “None”
 - “red”; “green”; “white”
 - “1”; “2”; “3”, “4”
- Boolean
 - has two states: false and true (cannot be renamed)
- Numbered
 - exact, discrete values
 - don't need to be integers
 - 1; 2; 3; 4; 5; 6 (*the possible results of a die roll*)
 - 0; 1.5; 3; 4.5; 6 (*e.g. the possible doses of a medicine*)
- Interval
 - consists of intervals on a continuous scale
 - starting and end points can be freely chosen, but the intervals have to cover all of the values in between
 - the break-points don't need to be integers
 - positive and negative infinity can be used as the endpoints (inf and -inf)
 - 0-1; 1-2; 2-4; 4-10; 10-100; 100-inf
 - -inf-0; 0-0.33; 0.33-0.5; 0.5-1; 1-inf

Note that you can give numbers as state names also in the labelled mode. The difference between this and numbered or interval modes is, however, that in the numbered and interval modes (together called “numeric”) you can perform calculations based on the state labels; in labelled mode, the numbers are interpreted just as names. (We'll see an example later.)

TRY:

Double-click the node so that a window pops up. Choose the Node tab (should be the one on the top), and use the dropdown selection of Type

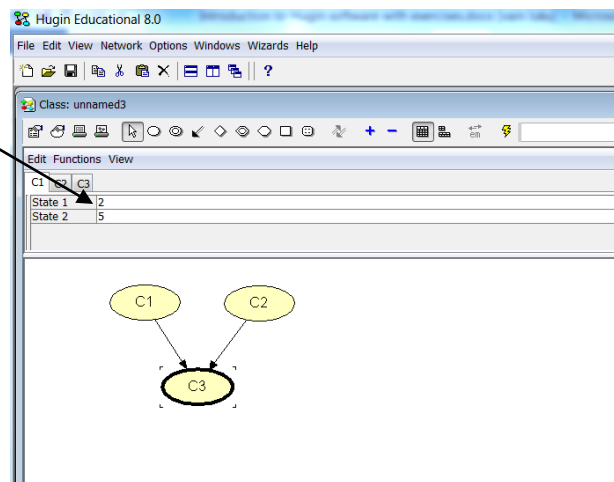
OR

Ctrl-click the node to see the **Tables pane**; then choose Functions → Set type

In the Tables pane, you can also see and edit the state names (just click on them to be able to edit) and add or remove them from the blue plus and minus signs.

Specify the probability distribution in the white area next to the labels. Note that you can write any numbers you like; they don't have to sum up to one at this stage, but they will, nonetheless, be interpreted as probabilities as the model is run.

Table's pane:
probabilities of the
variable C1 and its
two states



Table's pane:
probability of variable C3 is 4, given that the variable C2 is in state 2 and variable C1 is in State 1

C2	State 1		State 2	
C1	State 1	State 2	State 1	State 2
State 1	1	5	4	6
State 2	0.5	3.5	2	2

The diagram shows three nodes: C1, C2, and C3. C1 and C2 are parents of C3, indicated by arrows pointing to C3.

Exercise 1:

Create a model in Hugin with three variables: sky, wind and going for a picnic. Going for a picnic depends on the sky and the windiness of the weather. Fill the probability tables for the variables based on the tables below. In these tables the probabilities are real and they sum up to 1. Run the model. In the inference mode, you can set variables to a certain stages by double-clicking them. What is the probability that the wind is hard if you are not going for a picnic?

Sky	
Sunny	0.6
Cloudy	0.4

Wind	
Hard	0.35
Low	0.65

Going for a picnic			
		Wind	
		Hard	Low
Sky	Sunny	0.4	0.8
	Cloudy	0.1	0.5

OTHER VARIABLE TYPES

CONTINUOUS VARIABLE

Hugin also includes supports continuous variables, but in a very limited way. This is because the theoretic basis of how the probabilities can be updated is not developed yet.

Hugin's continuous node is depicted by the double oval, and the Hugin manual says:

"A continuous chance node (or CG node = continuous Gaussian node) represents a random variable with a Gaussian (normal) conditional distribution function. A

continuous chance node can be added to the current network by activating the continuous chance tool and clicking somewhere in the network pane.

“Currently, there are some restrictions to the use of continuous chance nodes, partly because the underlying theory is still being developed. The restrictions are listed below:

“The only continuous chance nodes currently supported represent variables with Gaussian (normal) distribution functions.

“A continuous chance node cannot be parent of a discrete chance node.

“Continuous nodes cannot be used in influence diagrams. That is, continuous nodes cannot exist in a network also containing utility nodes or decision nodes.”

DECISION AND UTILITY VARIABLES

Bayesian Networks can be enhanced by adding *decision and utility nodes*, thus turning them into *influence diagrams*. A **decision node** describes a decision, i.e. something that can and will be decided by the manager of the system. There can be several decision variables in the model, such as whether to collect more data or not; what kind of use restrictions to apply; whether to increase taxes or fees; etc etc.

In a decision node, you state all the decision options, which must be mutually exclusive. (The numbers associated to the decision options are not a probability table but they're called "initial policy" by Hugin.)

The decision nodes are depicted as rectangles in Hugin.

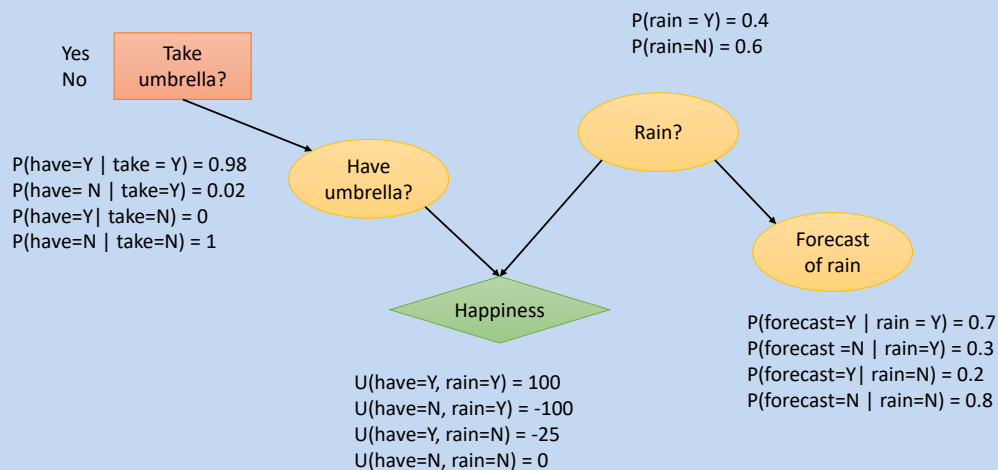
Utility nodes describe the utilities associated with all the possible outcomes. There can be several utility nodes in the network; but as the model is used to maximise the overall utility, they must be all measured in the same scale.

Utilities can be either positive or negative; for example, costs can be described as negative utilities and profits as positive utilities. Or you could use positive utility values to describe increased conservation status, and negative utilities to loss of biodiversity. Also, you could use very large negative utilities to state that certain states of nature are very undesirable (e.g. extinction of species or habitats, etc.).

Utilities are depicted as diamonds in Hugin.

TRY:

Build a BN that has utility and decision nodes using the information below:



SOURCE: [HTTPS://WWW.TECHFAK.UNI-BIELEFELD.DE/~SKOPP/LEHRE/STDKI_SS10/VL11-DBN_DECISIONMAKING.PDF](https://www.techfak.uni-bielefeld.de/~skopp/lehre/stdki_ss10/vl11-dbn_decisionmaking.pdf)

Notice that each of the decision options shows the expected utility that is related to making this decision. In other words, the decision node tells you the expected utility of making decision A, or decision B, etc., taking into account the rest of the model, i.e. both the probabilities of the different states of the variables in the model, and the utilities linked to these states.

When you set the values of the chance nodes to a certain state, for example you observe that there is rain and set the rain state to "100 % rain", the expected utilities of the different decision options might change. This is because we know more about the system's state now, and different decision options might be the best ones in different circumstances.

Can you answer these questions? 1) Should I take my umbrella? 2) What is the value of knowing the weather forecast?

ALTERNATIVE WAYS TO FILL IN THE CPTs

FUNCTIONS AND OPERATORS

You can type the probability distributions (conditional probability tables, CPT) manually. However, there is also a collection of built-in arithmetic operations and statistical distributions in Hugin that can considerably help in creating the CPTs. Before applying these, make sure that your variable is numeric (numbered or interval) and that they include of the possible outcomes of the expressions or distributions. That is, e.g. if your distribution goes from negative to positive infinity (such as Gaussian or lognormal distribution), you need to cover the whole range also in your variable; and if your operation can result in negative numbers, you need to have these negative numbers in your set of values.

The operators and functions available include:

- mathematical operators and functions such as:
 - addition
 - subtraction
 - multiplication
 - logarithm
 - exponential
 - sine
- comparison operators such as
 - equals
 - not equals
 - less than
- If-Then-Else
 - if(condition, if_true, if_false)
- logical operators
 - and
 - or
 - not

TRY:

First, ctrl-click the node to see the Tables pane. Then choose:

Functions → Expressions → Switch to expressions; then:

Functions → Expressions → Build expression → Click the expression line →
Choose from the Function category menu Arithmetic operations or if-then-else

or

Click the expression line in the tables pane and fill the expression by hand (hint:
for simple expressions the manual filling of expression may be easier, as the
expression builder may seem a bit complicated at first).

For more information about the logic behind typing the expressions can be found in the Hugin GUI help: <http://data.biotracer.hugin.com/htmlhelp/index.html>. Go to Tutorials → Table Generator Tutorial. It is recommended that you take a look at the webpage before going to the exercises.

Once you have filled the function to the expression line go to Funcions → Expressions → Show as table, and the Hugin will calculate the probabilities of the desired variable.

Exercise 2:

Build a model in which you have the number of adults and children, and define the amount of people as adults + children. Note that the maximum value of people has to be equal or higher than $\max(\text{adults}) + \max(\text{children})$.

Exercise 3:

Build a model in which the number of bird offspring depends on the stock size and on the food availability that year in the following manner: if the food availability is poor, then the number of offspring will be just 70% of (0.7 times) the stock size; otherwise, the number of offspring will be 300% of (3 times) the stock size.

DISTRIBUTIONS

You can also use in-built statistical distributions to define your probability tables. The program will compute the correct probabilities for each bin given the parameters of the distribution. Note that if you use a distribution that goes to infinity, such as the Gaussian (normal) distribution, you'll need to define your bins so that they go to infinity as well. The symbol for infinity in Hugin is *inf*; for example, a bin from one thousand to infinity would be *1000 - inf*.

Note also that the node type needs to be defined correctly. In order to be able to apply any continuous distributions, the node type needs to be *Interval*, and to apply discrete distributions, *Numbered* or *Interval*.

TRY:

First, ctrl-click the node to see the Tables pane. Then choose:

Functions → Expressions → Switch to expressions; then:

Functions → Expressions → Build expression

And then choose Distributions from the drop-down list

or write the distribution on the expression line in the tables pane.

MODEL NODES

Model node is, basically, just a more user-friendly interface to if-then-else expressions: instead of writing the complex if statement, you can split the if expression into several expressions, one for each different value of the parent. For example, in the picture below, normally you would have an if_then_else function to say “If the town is Paris, then the price of the house is ... and else if the town is Bergen, then the price is ... and else the price is ...”. The model node makes this prettier.

The screenshot shows a software window titled 'Edit Functions View' with a 'Prize' node selected. Below the title bar is a table with columns for 'Town' (Paris, Bergen, London) and 'Bedrooms' (1, 2, 3, 4). The 'Expression' row contains normal distribution formulas: 'Normal (150 + 100 * Bedrooms, 300)' for Paris, 'Normal (100 + 80 * Bedrooms, 200)' for Bergen, and 'Normal (130 + 70 * Bedrooms, 250)' for London. Below the table is a diagram showing 'Town' and 'Bedrooms' nodes pointing to a 'Prize' node.

Town	Paris				Bergen				London			
Expression	Normal (150 + 100 * Bedrooms, 300)				Normal (100 + 80 * Bedrooms, 200)				Normal (130 + 70 * Bedrooms, 250)			
Town	Paris				Bergen				London			
Bedrooms	1	2	3	4	1	2	3	4	1	2	3	4
-inf - 50	3.82188E-31	1.64718E-67	2.65912E-...	1.52738E-...	1.92107E-20	3.51796E-50	9.49698E-94	3.50998E-...	1.1908E-21	2.60337E-44	1.9403E-75	4.71644E-...

```
graph TD; Town((Town)) --> Prize((Prize)); Bedrooms((Bedrooms)) --> Prize;
```

IF YOU'RE INTERESTED

The material from this point on is not essential to this course. However, feel free to take a look; the instance nodes can be useful in some situations.

INSTANCE NODES

Instance nodes are a way to create copies, or representations, of other BNs into another BN model. In other words, you can have “sub-models” nested within your model. There are a few cases in which this can be practical:

- if you need to have several, identical elements in your main network (for example, time slices or different locations), it might be useful to create this piece once and then make instances of it into the main network
- if your model is very complex, and you need to communicate it to stakeholders, it may be useful to simplify the representation without simplifying the actual model

BNs that include sub-networks (instance nodes) are called object-oriented, or hierarchical, BNs.

The models can be nested on several levels, i.e. the model included as a sub-network can include sub-networks itself.

Building hierarchical networks requires care. An instance node should be thought of as a copy of the original subnetwork. It connects to other nodes via some of the (basic) nodes in the subnetwork; these nodes are called interface nodes. There are two types of interface nodes: output and input.

Output nodes are visible also outside of the instance, and they can be used as parents of other nodes in the main network. All of the variables that need to be parents of other variables in the main network must be defined as output nodes.

Input nodes are best thought of as placeholders for such variables outside the sub-network which are parents of variables in the sub-network. The structure of the probability tables of these placeholders must be identical to the tables in the actual node in the main network (i.e. the discretization or labels have to be identical, but the probability distributions don't.) When the input node (the placeholder node) is connected to the node outside the sub-network, the probability distributions become identical, i.e. the input node will just reflect the values of the actual variable in the main network. If the input node doesn't have parents, the default distribution of the input node will be used, and that's why they need to be considered with care.

For more instructions, open the Hugin Help files and search for “How to Build an Object-Oriented Bayesian Network”. There you find a tutorial with more technical support.

TRY:

Define the input and output nodes for one of your model files (or create a new, simple model). Open a new network file, and create instances of your sub-network. Link them together, make new variables as needs be. Experiment with the model.

NOTE:

Instance nodes are suitable if and only if we wish to make multiple exactly similar copies of a network; not only the structure but also the parameters. The network of which an instance is to be made must be open, and they can't be closed as long as the OO network is open.