Hugin Exercises

The following exercises are meant to get practical experience with a program, called Hugin, for the presentation of and reasoning with probabilistic networks. At the end, you will find a proposal for your own project. The project assignment requires the writing of a short report that describes (i) the network including the relevant probability tables, (ii) the results of your calculations, (iii) an interpretation of your findings.

Exercise 1 (Basics)
Follow the Hugin tutorial [see hugin help/hugin GUI (tutorial) → Bayesian Networks Tutorial and How to Build a Bayesian Network]. Make a directory in which your own network can be stored. Study the built in *Apple Tree Example*. Learn how pieces of evidence can be accommodated in a network. Make (from scratch) your own copy of the *Apple Tree Example*. That is, get the necessary data from the built-in example and rebuilt the network. If you think this is silly, you are free to makeup your own network of comparable complexity, but try to get familiar with the relevant parts of the user's guide.

Exercise 2 (Bayes' rule)
Use Hugin as a probability calculator for the problem of exercise 2.3 (a taxi in Athens). Introduce a Boolean variable **B** (*the taxi was blue*) with the given prior probability, and a Boolean variable **LB** (the taxi looked blue). Check that updating for the variable **B** with the observation that the taxi looked blue.

Exercise 3 (Monty Hall Puzzle)
Use Hugin as a probability calculator for the Monty Hall Puzzle.

Exercise 4 (Local Semantics and Independence)
Build probabilistic networks with the following underlying structure (make up the probabilities yourself, assume three states for each random variable):
1. A → B → C (serial connection)
2. B ← A → C (diverging connection)
3. B → A ← C (converging connection)

Investigate how the nodes can influence each other, that is, how evidence is propagated through these structures. Compare your findings with what you have learned about independence (local semantics of a network)

Exercise 5 (Asia)
Run the Asia example (described in the Hugin Modelling Examples; to run it open *asia.net* in the samples folder)

Final assignment (Hugin assignment)

The following example (vacation example) gives one proposal only.

Feel free to construct your own scenarios using belief networks, for instance

(a) for predicting whether you will pass or fail the course *Reasoning with Uncertainty*
(b) for predicting whether you are in danger to become an alcoholic
(c) for predicting what are your chances to get a desired job after finishing your studies.

The minimum requirements for the network are (a) at least 15 nodes and (b) some nodes are included with noisy OR characteristics.

> **Vacation example**
> Build a probabilistic network for predicting whether you will have a nice vacation or not. Such a network at least contains a variable V(acation) with the instances: nice, not nice. This matter is influenced (directly or indirectly) by, among others: the length of the vacation, the weather, whether you will have passed all your exams or need to do them over in August, the money you can spend, whether you succeed in getting a (well-paid) job for the holidays.
>
> The network should principally be applicable to your own situation, but it should also be sufficiently general to be applicable to other people, say fellow-students, as well. Incorporate at least the aspects mentioned above, but feel free to add some aspects which are essential in your situation (having a holiday affair, getting a driver license, …). Carefully consider the structure of the network!
>
> Proceed by instantiating the network to describe your present situation. What is your chance of a nice vacation? To what extent can you still influence this chance? What is the best way to increase this chance? Are the obtained results intuitively correct? If not, are you inclined to correct your intuitions or the network.

Write a report that contains at least the following three things:
(i) the structure of the network including the relevant probability tables; make use of the noisy OR (If that’s not possible for some reason, please, give an explanation why it’s not possible)
(ii) the results you obtain by instantiating the network
(iii) discussion of the results including the aspects mentioned above (are the obtained results intuitively correct? If not, what should be corrected?)