CMSC 421 Programming Exercise 2

Due Date: Tuesday, November 18 before 5PM

This is an exercise in knowledge engineering using belief networks. **You are allowed to submit this exercise in groups of two to three.** The task is to construct a Bayesian network that models some domain that you think might be interesting. To build this network, you will use a BN software package called Hugin, described below. Note that this is purely a knowledge engineering assignment; no programming is required.

A Bayesian network typically has (among others) some *evidence variables* and some *query variables*. The evidence variables are ones whose value you often get to observe (e.g., phone call or the disease symptoms in our example from class). Query variables are ones whose probability you often want to query (e.g., burglary and earthquake; disease). Networks also often contain hidden variables, whose value is neither of particular interest and may not be observed directly, but is relevant to defining a correct model (e.g., alarm). Note that the variables you choose to put in a network depend on the task you are trying to accomplish with it.

You should pick a domain that you think is interesting, but here are some examples to get you started thinking about this:

- Modeling the viewing behavior of the home audience of a local television station, perhaps in order to allow advertisers to target their ad campaigns.
- Deciding whether to grant automobile insurance to a driver.
- Deciding what's wrong with your computer (running Windows XP).

For example, in the TV domain, the query variables might indicate whether the viewer watches a particular television program or type of program. Other variables might include some viewer demographics, some of which might be observed and others hidden.

What to Hand In

To fulfill the basic requirements, the network you construct should have around 8–10 nodes, and at least 3 layers. Note that binary variables substantially reduce the amount of parameter estimation you need to do, so you might try to stick to those except where it really doesn't make sense.

We expect you to put in a reasonable amount of effort designing your network. You should document the design decisions that you have made. Your network will be evaluated based on whether it is a reasonable representation of the domain. That doesn't mean that we expect you to hand in a network which is a perfect network. In terms of the structure, your links should go in the right direction. They should also correspond to a reasonable notion of probabilistic dependency. If you drop a link which we might reasonably expect to be there, you should explain your decision. Similarly if you add a link representing a non-obvious dependence.

We also do not expect you to come up with completely realistic conditional probability parameters. However, your conditional probabilities should have the right order of magnitude (e.g., the probability that an infant enjoys watching Congressional hearings wouldn't be 50%). They should also be qualitatively correct relative to each other, e.g., people who typically sleep late won't watch a lot of morning news programs.

You should hand in the following:

1. Description of the example application you had in mind.

2. An explanation of the meaning of each variable and of the values that it takes. It is important that your variables pass the *clarity test*, i.e., they should be sufficiently well-defined that an omniscient being (one who can see the future and observe everything) should be able to determine its value unambiguously. For example, in the automobile insurance application, the variable "Car accident within the next year" passes the clarity test, whereas "Risk of accident" does not.

3. For each node, whether it is always observable (e.g., on a marketing survey form), sometimes observable, or never observable.

4. Your network, submitted electronically as specified below.

5. A brief justification for any non-obvious links you chose to put in or to omit. When deciding whether to explain something, remember that what is obvious to you might not be obvious to us.

6. Description of how you would use the network to guide your decisions in the task that motivated its creation.

7. Four or five reasonable test cases. For each case, pick a set of observations that you might get in real life, instantiate them as evidence, and (have the system) compute the probabilities of a number of reasonable query nodes (these do not have to be one of the designated query nodes). You should hand in the description of the evidence, and the probabilities you got as an answer. If your probabilities don't make sense to you, you should probably go back and revise your network. Your cases should test diagnostic/evidential reasoning, causal reasoning, and intercausal reasoning (explaining away), with at least one data case incorporating more than one of these forms of reasoning. For each case, specify which types of reasoning it covers.

Meeting the basic description above will receive 80and writeups that display thorough and thoughtful design will receive additional credit.

Submitting Your Network

To do this project, you will be using a demo version of the HUGIN software, downloadable from the class web page
`http://www.cs.umd.edu/class/fall2003/cmsc421-0101/resources.html`. There are windows, sun and linux versions of the software available. The windows version is likely to be the most stable version.

When you are finished with your network, you should save it as a `.net` file. You should mail the net file to the ta, lg421075@umd5.umd.edu, and turnin a physical copy of your writeup.