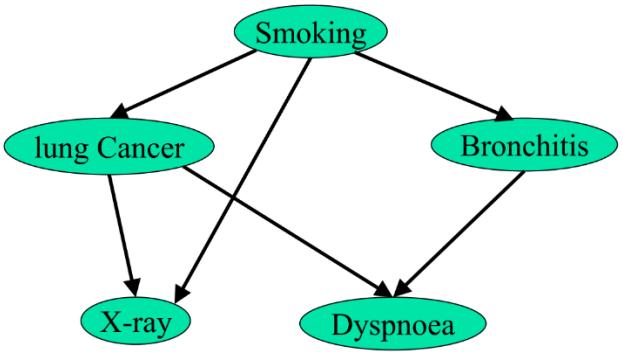
CMP9794M Advanced Artificial Intelligence – Workshop Week 2

UNIVERSITY OF LINCOLN

Summary: In this workshop you will carry out calculations related to Bayesian networks with exact inference using an implementation by the delivery team, which is available via Blackboard under the workshop materials of week 2. A class diagram of such an implementation can be found in file `workshop-w2/doc/CMP9794_BayesNets_ClassDiagram.pdf`. Briefly, the **NB_Classifier class** (discussed in week 1) will be used here to reuse some of its functionality such as reading data files in csv format. The **CPT_Generator** class fill be used to generate Conditional Probability Tables (CPTs). The class **BayesNetExactInference** will be used to do probabilistic inference using the algorithm 'inference by enumeration'. The remaining classes (**BayesNetReader** and **BayesNetUtil**) will be used for (i) reading configuration text files and (ii) specifying low-level behaviours.

Task 1: **Number of parameters in Bayesian Networks**

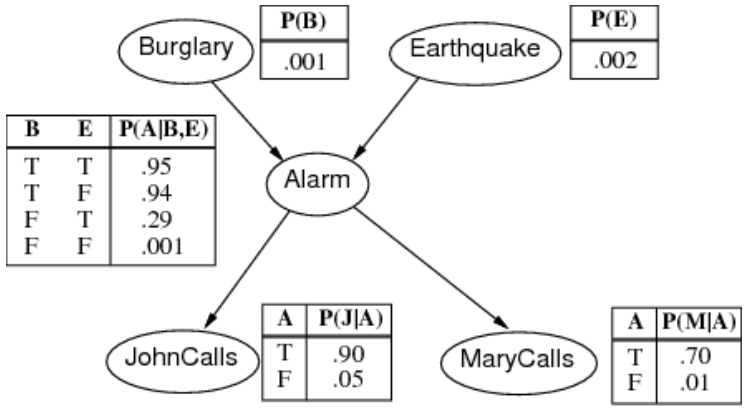
Given the following Bayesian Network with binary random variables,



a. How is the joint distribution expressed? $P(S, C, B, X, D) =$
b. What is the number of parameters (i.e. probabilities)?
c. What is the number of parameters assuming that random variable S=Smoking has three values instead of two (and the other random variables remain binary)?

Task 2: **Probabilistic inference via 'Inference by Enumeration'**

Use the Burglary Bayes net discussed during this week's lecture to calculate $P(B|j, m)$ and $P(E|j, m)$ using the program `BayesNetExactInference.py`



Download the program and data from blackboard to compare the manual calculations of your homework. Run the code from the command line (or from an IDE environment) as follows:

- `Python BayesNetExactInference.py ../config/config-alarm.txt "P(B|J=true,M=true)"`
- `Python BayesNetExactInference.py ../config/config-alarm.txt "P(E|J=true,M=true)"`

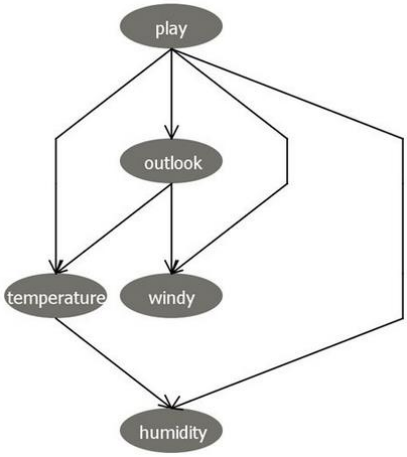
Here we assume that we know the Bayes net structure and its parameters (conditional probabilities), which are specified in the file `config-alarm.txt`. You may want to inspect this text file to see how the Burglary Bayes net has been specified in that file. This file specifies the following information:

- name of your Bayesian Network
- random variables involved (short name inside round brackets) separated by semi-colons,
- structure (conditional probabilities in your network) separated by semi-colons, and
- probabilities (Conditional Probability Tables, CPTs, in the case of discrete Bayes nets).

In your own time you may want to inspect the class `BayesNetExactInference.py` (and related files) to see how the algorithm `Inference by Enumeration' was implemented.

Task 3: **Parameter learning for Bayesian networks**

Given the following Bayesian Network, with data discussed last week, estimate its conditional probability tables using the program `CPT_Generator.py`



Using the workshop materials of this week, run the program from the command line or from an IDE:

- `python CPT_Generator.py ../config/config-playtennis.txt ../data/play_tennis-train.csv`

The txt file above assumes an initial manual definition of the name, random variables, and structure. The list of random variables and structure (sequence of conditional probabilities) assumes the following: start from variables without parents, continue with those having an increasing number of parents, until the variable with the largest number of parents. See examples in * - withoutCPTs.txt

Once the command above is executed, it will rewrite the txt file by completing it with the corresponding CPTs. For small datasets and Bayes net structures, the execution will take a small fraction of time. Just be cautious that this is not the case for large datasets and/or Bayes nets.

You may want to compare file `config-playtennis.txt vs. config-playtennis –withoutCPTs`. The latter does not have any CPTs yet and can be created (manually) prior to generating CPTs. It is recommended to keep a copy of a file without CPTs due to the rewriting step.

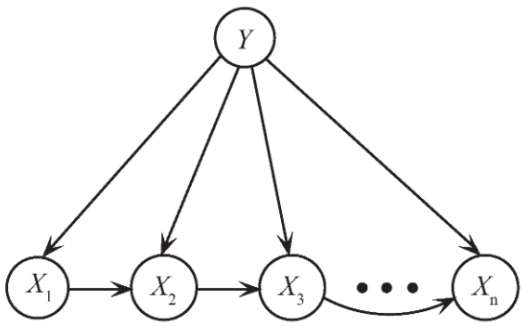
Once you have generated your config file with CPTs, use it for probabilistic inference! – as in task 2. For example:

```
python BayesNetExactInference.py ../config/config-playtennis.txt
"P(PT|W=strong,O=rain)"
```

What is the probability distribution of $P(PT|W = strong, O = rain)$?

Task 4: **Homework**

4.1 Apply the code of tasks 2 and 3 to the task of language detection, where you should first estimate the parameters of your Bayes Net and then use them to carry out probabilistic inference. Consider the following structure for your Bayes net, where X_i are input features (characters in a sentence in this task) and Y is the target variable corresponding to the language to predict (English, Dutch or Spanish in our case):



4.2 Implement the algorithm `Variable Elimination' discussed in the previous lecture and compare the inference results against those of task 2.