Artificial Intelligence and Decision Systems

Assignment #3

Professor Luís Manuel Marques Custódio

Students: Henrique Carvalho - 70327 Gonçalo Ribeiro - 73294

1 Usage

The program is called using the following command :

```
python3 bn_inference.py [-h] [--verbose] bayesnet query
```

The input BN and query files are mandatory, the verbose flag allows to ask a step-by-step solution. A help option also exists.

2 File Parsers

2.1 Bayes Network Input

The Bayesian network parser, `BNParser`, is thoroughly protected. It has 24 points of failure (`raises`), each with a costum error message. It is well commented and the error messages also help understanding what the code does. Some features of the `BNParser` class are the following:

- Comments are allowed at the end of any line
- Variables' names are case sensitive
- Error if some declaration other than `VAR` and `CPT` are found (case insensitive)
- Error if more than one value is given to `name`, `alias` or `parents` in a VAR definition
- Error if `name` or `values` are missing in a `VAR`
- Error if the `CPT` contains a value not allowed for the variable
- Error if a `CPT` refers to an undefined `VAR`
- Error if the number of entries in a `CPT` is not consistent with its parents' domains' cardinality and the cardinality of the domain of the variable itself.

2.2 Query Input

For this part, we defended the parser from the following situations:

- A variable is not defined in the Bayes Network
- There is more than one variable in the variables to query (the `QueryParser` does not allow this, but the `BayesNet.ppd()` method *does* support multiple query variables)
- The expected `EVIDENCE` word is not found
- The number of evidence variables in the file is negative
- The number of evidence variables does not match the one stated
- The evidence is for a variable that does not exist in the Bayesian network
- The value of an evidence variable is not in the domain for that variable
- The values of the domain are case insensitive, so that both `T` and `t` correspond to the same value in the domain.

3 Data Structures

Each of the structures referred in the following subsections are implement as a class. The internals of those classes are described.

3.1 Bayes Network Parser – `BNParser`

This parser stores all the information retrieved in a dictionary that is later used by `BayesNet`. All the aliases appearances are converted to to the names of the variables. A dictionary converting aliases to names is stored in the parser object so that alias-name conversion can later be performed by the `QueryParser` object.

The use of aliases instead of names in the program could speed up the dictionaries' hashing, but but the names were preferred in favour of the aliases because they carry more information about the problem.

The method that performs the exact inference is called `ppd()`. The name stands for 'posterior probability distribution.' The Variable Elimination algorithm is performed in this method. It is interesting to note that our implementation supports multiple query variables.

3.2 Bayesian Network – `BayesNet`

A dictionary was used to implement the Bayesian network. It is indexed with each variable's name and the values of the dictionary are information regarding each variable, namely its parents, domain and CPT. The parents are stored in a list and so are the possible values for the variable (its domain). The CPT is stored as a dictionary whose keys are tuples corresponding to the combination of values of the parents and the variable itself. The values of this dictionary are the conditional probabilities.

3.3 Factors – `Factor`

The `Factor` class stores the variables for that factor and the table with probabilities corresponding to those variables. The class provides two important methods: `join()` and `eliminate()`. The first performs the product of an arbitrary number of factors; the second eliminates a single variable from an existing factor. These methods are used in `BayesNet.ppd()` to perform the Variable Elimination algorithm.

4 Output File

To write the solution, there is also a class that receives the name of the query file. To print the solution file, is required a solution (CPT for the query variable), a string carrying the query (read from the query file), an evidence string (also read from the query file) and, if the verbose option was selected, the step-by-step solution string in the `BayesNet` class after inference was done with `ppd()`.

The output file starts by presenting the query variable, the evidence and the CPT for the solution. If the verbose option is selected, a step-by-step solution will be presented. For each step, in the Variable Elimination algorithm, first all factors will be printed in one line. Then follows the join part, which shows the factors to join and the CPT for each one of them. After joining, comes the variable to eliminate and the factor and respective CPT resulting from it.

References

[1] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach.* Prentice Hall, 3rd edition, 2009.