CS4244 Project 2 Report

- He Yuxiao A0268714W
- Lyu Xingyu A0268704X

Problem

P2 Encoding of Bayesian Networks

Representation of Bayesian Network

The Bayesian network is represneted in an array of nodes. Each node has three members: name, parents and CPT:

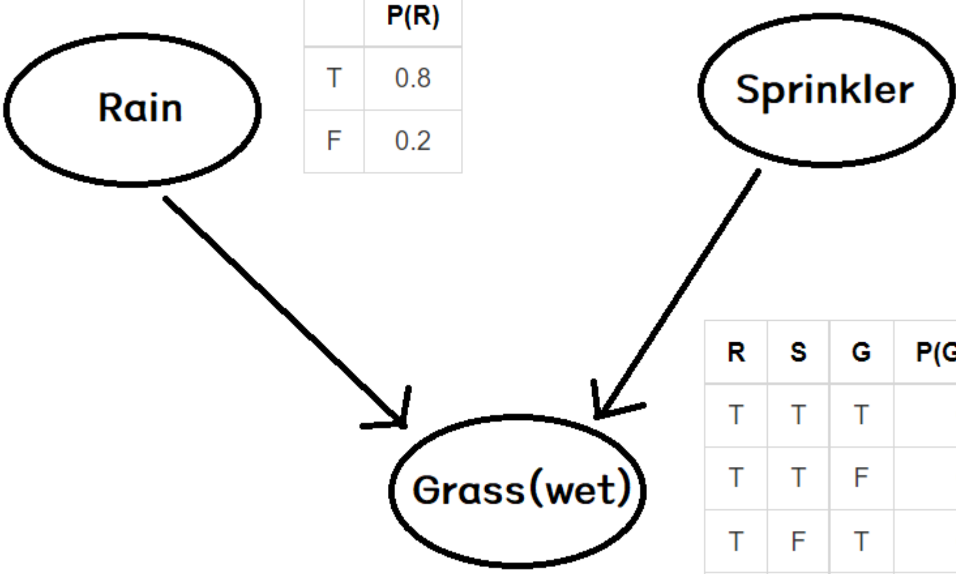
```
class BayesianNetwork{
    vector<Node*> nodes;
};

class Node{
    string name;
    vector<Node*> parents;
    unordered_map<string, int> cpt;
};
```

Here the CPT is represented using an unordered map, which maps an assignment to its probability. For example, if the assignment is "A = 0, B = 1, C = 0, D = 1" and the corresponding probability is 0.8, then the pair `<"0 1 0 1 ", 0.8>` will be an element in CPT.

Note that order of the string in CPT is the same with the order of `parents`.

CNF Variables and their Weight

The encoding method of Bayesian network is based on the method professor taught in class. Basically, it creates one variable for each line of CPT table. For example, if we have the following Bayesian network:



	P(R)
T	0.8
F	0.2

	P(R)
T	0.6
F	0.4

R	S	G	P(G | RS)
T	T	T	0.8
T	T	F	0.2
T	F	T	0.7
T	F	F	0.3
F	T	T	0.6
F	T	F	0.4
F	F	T	0.3
F	F	F	0.7

Then for the CPT of "Rain", we have new variables C_R and $C_{\bar{R}}$. For the CPT of "Grass(wet)", we have new variables $C_{RSG}, C_{RS\bar{G}}, C_{R\bar{S}G}, C_{R\bar{S}\bar{G}}, C_{\bar{R}SG}, C_{\bar{R}S\bar{G}}, C_{\bar{R}\bar{S}G}, C_{\bar{R}\bar{S}\bar{G}}$.

The weight of each variable is the weight in the corresponding CPT line. For example, $W(C_{R\bar{S}G}) = 0.7$ and $W(C_R) = 0.6$.

The weight of negation of each variable is always 1. For example, $W(C_{R\bar{S}G}) = 1$ and $W(C_S) = 1$.

CNF Formula

The CNF formula consists of two parts. The first part is the formula constructed from evidence, and the second part states that we can only choose exactly one variable that represents one line from each CPT.

Construction of the first part

For each node in the Bayesian network, if it is observed with a certain truth assignment, the other truth assignment will not be considered. For example, if S is observed to be true, then only $C_S, C_{RSG}, C_{RS\bar{G}}, C_{\bar{R}SG}, C_{\bar{R}S\bar{G}}$ will be considered.

For each CPT, a clause will be generated. Consider the CPT of node "Grass(wet)", if S is observed to be true, the clause corresponding to this CPT will be $(C_{RSG} \vee C_{RS\bar{G}} \vee C_{\bar{R}SG} \vee C_{\bar{R}S\bar{G}})$.

Finally, all the clauses will be conjunctured to form the first part of the CNF.

Construction of teh second part

The second part states that we can only choose exactly one variable representing one line from each CPT. For example, we can only choose exactly one variable from $C_R, C_{\bar{R}}$, also exactly one variable from $C_{RSG}, C_{RS\bar{G}}, C_{R\bar{S}G}, C_{R\bar{S}\bar{G}}, C_{\bar{R}SG}, C_{\bar{R}S\bar{G}}, C_{\bar{R}\bar{S}G}, C_{\bar{R}\bar{S}\bar{G}}$.

Use $\mathrm{ExactlyOne}(A_1, A_2, \ldots)$ to represent this process. We have to find a CNF representation of $\mathrm{ExactlyOne}()$. First, we should choose at least one variable from the variable list $A_1, A_2, \ldots$, thus we have a clause $(A_1 \vee A_2 \vee \ldots)$.

Second, at most one variable can be chosen, i. e. each pair of the variables contains at least one variable that is not chosen. Thus we have clauses $(\neg A_1 \vee \neg A_2), (\neg A_1 \vee \neg A_3), \ldots, (\neg A_2 \vee \neg A_3), \ldots$.

Finally, all the clauses will be conjunctured to from the second part of the CNF.

Structure of the Source Code

The source code has 4 parts:

- **BayesianNetwork**

This part is explained in first part of this report. It is the inner Bayesian network data structure.

- **CNF**

This is the CNF data structure, which has methods related to CNF representation and weight/CNF file I/O.

- **Converter**

This converter converts .uai file to the inner Bayesian network data structure.

- **CNFConstructor**

This CNF constructor is the main part of the algorithm. It encodes the Bayesian network using the algorithm discussed above.

Replace `TEST_FILE` to specify the .uai and evidence file, then execute the main function in `Main.cpp`. Then a .cnf file and .weight file representing the generated CNF and the weight of each literal will be generated under the same folder containing .uai and evidence file.

Self-Evaluation

In this project, we implemented an algorithm encoding the Bayesian network based on evidence.

The advantages of our work are:

- A clear data structure representing Bayesian network.
- The algorithm is relatively direct and easy to understand.

Outcomes

Through this project, we have a better understanding of Bayesian network and its encoding algorithm. Computing probabilistic inference queries is a fundamental problem in Bayesian networks, but traditional inference algorithms can be time-consuming for large and complex networks. By encoding these queries as weighted counting problems, existing efficient counting algorithms can be leveraged to speed up computations.