Bayesian Network for

Patient Outcomes

NIRALEE KOTHARI

INTRODUCTION AND PROBLEM STATEMENT

The project aims to develop a Bayesian Network (BN) that can predict patient outcomes based on a range of variables including the patient's current condition, diagnostic test results, and treatment decisions. Bayesian Networks are probabilistic models that represent variables and their conditional dependencies; they are particularly useful for decision-making under uncertainty.

ENVIRONMENT SETUP

Install **pgmpy** using pip

Import **BayesianNetwork** , **TabularCPD**, and **VariableElimination** from **pgmpy**

IMPLEMENTATION DETAILS

In the implementation of the Bayesian Network for this project, the following steps and details were adhered to:

The BN was implemented in Python using the `pgmpy` library. The nodes were defined and conditional probability distributions (CPDs) were established for each node. Inference was performed using the Variable Elimination method.

Defining the Bayesian Network Structure

- The structure of the Bayesian Network was defined as a Directed Acyclic Graph (DAG) where nodes represent the variables, and edges represent the dependencies between these variables.

- Four nodes (A, B, C, D) were established to represent the Patient Condition, Test Result, Treatment Decision, and Outcome respectively.

- The network was constructed with the following relationships: `A -> B`, `A -> C`, `B -> C`, and `C -> D`.

Conditional Probability Distributions (CPDs)

- Independent CPD for Node A: Defined the prior probabilities for the patient's condition.

- Conditional CPDs for Nodes B and C: Established probabilities based on the condition of the patient and the test result.

- Conditional CPD for Node D: Set the probabilities of the outcome based on whether treatment was administered or not.

CPD Tables

The CPD tables were constructed as follows:

- `cpd_A` for Node A was a simple table with probabilities [0.7, 0.2, 0.1] for states [good, fair, poor].

- `cpd_B` for Node B was a 2x3 table accommodating the probabilities of test results given the patient's condition.

- `cpd_C` for Node C was a 2x6 table that included probabilities for the treatment decision given both the patient's condition and the test result.

- `cpd_D` for Node D was a 2x2 table that detailed the probabilities of outcomes based on the treatment decision.

Inference Engine

- The Variable Elimination algorithm was used for performing inference on the network. This method computes the posterior distribution for a subset of variables, integrating out the others.

- Queries were set up to calculate various probabilities such as the likelihood of patient outcomes under certain conditions.

Inference Queries

- The inference engine was queried to calculate the distribution of the patient's condition without any evidence.

- Probabilities of test results given specific conditions of the patient were determined.

- The likelihood of treatment decisions given both the patient's condition and test results was computed.

- Predictions of patient outcomes under various evidence combinations were obtained.

Challenges and Resolutions

- The initial challenge was ensuring the correct setup of CPDs, especially for nodes with multiple parent nodes.

- A careful check of the dimensions and probabilities within CPDs was necessary to avoid inconsistencies in the results.

- Another challenge was ensuring that evidence is accurately represented when performing queries with the inference engine.

- The discrepancy in the expected outcome for treated patients necessitated a review of the CPDs and inference queries to ensure correct implementation.

.

Logic

The logic of the Bayesian Network implemented in Python using the `pgmpy` library is to model and infer the relationships between a patient's condition, diagnostic test results, treatment decisions, and outcomes. CPDs define how variables are probabilistically related, and the `VariableElimination` method is used to compute the likelihood of outcomes given certain evidence. The implementation steps include setting up the environment, defining the network structure, establishing CPDs, and performing queries to predict outcomes.

EVALUATION RESULTS AND INSIGHTS

Treatment Decision and Outcome:

- The probabilities associated with the treatment decision and patient outcomes after treatment did not align with the expected probabilities stated in the project's specifications.

Probability Distribution of Patient Condition:

- The probability distribution of the patient condition was consistent with the initial problem statement.

Probability Distribution of Test Result Given Patient Condition:

- The test result probabilities given a poor patient condition matched the expected values from the problem statement.

Treatment Decision Given Patient Condition and Test Result:

- The treatment decision probabilities given a fair patient condition and a negative test result were correct.

Outcome Given Specific Patient Condition, Test Result, and Treatment Decision:

- The outcome probabilities for a patient in a fair condition with a positive test result who was not treated were as expected.

Most Likely Outcome and Treatment Decision:

- Outputs for the most likely outcomes and treatment decisions were not explicitly included in the provided screenshots.

Sensitivity Analysis of Outcome to Treatment Decision:

- The sensitivity analysis showed reversed probabilities contrary to the expected values, suggesting a potential error in the implementation.

Outcome Distribution Under Different Conditions:

- The probabilities of a positive outcome for treated patients did not match the expected value, indicating a discrepancy.

OUTPUTS

The following outputs are a selection of computations from the implemented Bayesian Network:

Query-1

```
Query 1 Result:
+---------+----------+
| A       |   phi(A) |
+=========+==========+
| A(good) |   0.7000 |
+---------+----------+
| A(fair) |   0.2000 |
+---------+----------+
| A(poor) |   0.1000 |
+---------+----------+
```

Query-2

Query 2 Result:

B	phi(B)
B(positive)	0.3000
B(negative)	0.7000

Query-3

Query 3 Result:

C	phi(C)
C(treated)	0.3000
C(not treated)	0.7000

Query-4

Query 4 Result:

D	phi(D)
D(positive)	0.2000
D(negative)	0.8000

Query-5

```
Query 5 Result:
 {'D': 'positive'}
```

Query-6

```
Query 6 Result:
 {'C': 'treated'}
```

Query-7

```
Query 7 Result (for treated):

Treatment: treated, Outcome: {'D': 'positive'}
Treatment: not treated, Outcome: {'D': 'negative'}
```

Query-8

```
Query 8 Result under varying conditions:
Condition: good, Test Result: positive, Treatment: treated, Outcome: {'D': 'positive'}
Condition: good, Test Result: positive, Treatment: not treated, Outcome: {'D': 'negative'}
Condition: good, Test Result: negative, Treatment: treated, Outcome: {'D': 'positive'}
Condition: good, Test Result: negative, Treatment: not treated, Outcome: {'D': 'negative'}
Condition: fair, Test Result: positive, Treatment: treated, Outcome: {'D': 'positive'}
Condition: fair, Test Result: positive, Treatment: not treated, Outcome: {'D': 'negative'}
Condition: fair, Test Result: negative, Treatment: treated, Outcome: {'D': 'positive'}
Condition: fair, Test Result: negative, Treatment: not treated, Outcome: {'D': 'negative'}
Condition: poor, Test Result: positive, Treatment: treated, Outcome: {'D': 'positive'}
Condition: poor, Test Result: positive, Treatment: not treated, Outcome: {'D': 'negative'}
Condition: poor, Test Result: negative, Treatment: treated, Outcome: {'D': 'positive'}
Condition: poor, Test Result: negative, Treatment: not treated, Outcome: {'D': 'negative'}
```

CONCLUSION AND RECOMMENDATIONS

Based on the evaluation, there are inconsistencies in the output when compared to the expected probabilities. These inconsistencies are critical as they can significantly impact the model's reliability and its application in a healthcare setting. It's recommended to:

- Re-examine the implementation of CPDs, especially for the node representing the outcome, ensuring they reflect the correct probabilities.
- Ensure that the inference engine is applying evidence properly and the calculations are accurately reflecting the CPDs.

This report highlights the need for rigorous testing and validation of Bayesian Network models before deployment in sensitive areas such as healthcare. The insights gained call for a careful review of the model's logic and a thorough understanding of the domain knowledge to ensure its efficacy and safety.