Alessandro Allegranzi

BU MET CS 767

Assignment 6: Bayesian Network

6/19/2024

06/19/2024

MET CS 767 Assignment 6: Bayesian Networks

*Alessandro Allegranzi*

Create your own application of a Bayesian network, like the “sprinkler” problem in the notes. You can do this by hand, via a Python program, or using an online Bayesian network tool.

Please leave the gray text and the headings unchanged etc.

# Requirements for your application

Supply the requirements, including the nature of its inputs and that of its outputs.

The following Bayesian Network can help in understanding how different factors influence voting behavior in an election, aiding in the analysis of voter decisions and the development of political strategies.

**Events (Nodes):**

1. **Political Affiliation (PA)**: The political leaning of a voter (e.g., 0: Conservative, 1: Liberal, 2: Independent).
2. **Economic Outlook (EO)**: The voter's perception of the economic future (0: Positive, 1: Negative).
3. **Vote (V)**: The actual voting decision (0: Conservative, 1: Liberal).

**Dependencies:**

* **Political Affiliation** influences **Actual Vote**.
* **Economic Outlook** influences **Actual Vote**.
* **Economic Outlook** influences **Political Affiliation. (**Lower economic outlook affects political affiliation towards conservative.**)**

**Requirements:**

* **Inputs**:
  + **Political Affiliation**: An integer representing the political leaning of a voter.
  + **Economic Outlook**: An integer representing the voter's perception of the economic future.
* **Outputs**:
  + The probability distribution of the **Vote**, given the political affiliation and economic outlook.

I have implemented the above network in python using the pgmpy library and will attach the .py file with my submission [1]. [GH link](https://github.com/1-8192/bu_cs_767/blob/main/moduel_6/bayesian_network.py)

# Diagram

Provide a figure like the one in the notes for the Bayesian network.

I created the diagram on the [BayANet website](https://profgavinbrown.github.io/projects/bayes_nets/). The probabilities are arbitrary values, since real data is not available. I will include a [json file](https://github.com/1-8192/bu_cs_767/blob/main/moduel_6/bayes.json) of the diagram with the submission.

A diagram of a political process

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a graph

Description automatically generated

# Example 1

Give an example of an input and the resulting outputs, with an explanation of the computation.

**Input and output of python app:**

“What is the probability of a liberal political affiliation given a liberal vote and positive economic outlook?”

The input is a positive economic outlook (1) and a liberal vote (1), which should lead to a likelier liberal affiliation (1).

result = inference.query(*variables*=['Political Affiliation'], *evidence*={'Economic Outlook': 1, 'Vote': 1})

+--------------------------+------------------------------+

| Political Affiliation | phi(Political Affiliation) |

+==========================+==============================+

| Political Affiliation(0) | 0.2319 |

+--------------------------+------------------------------+

| Political Affiliation(1) | 0.5072 |

+--------------------------+------------------------------+

| Political Affiliation(2) | 0.2609 |

+--------------------------+------------------------------+

The answer to the initial question is .5072 using the python model.

**Computation:**

Based on the module 6 Bayesian Network slide:

1. **P(V=Liberal∣PA=Liberal,E=Positive)**:
   * From the CPD of Vote = **0.8**
2. **P(PA=Liberal∣E=Positive)**:
   * From the CPD of Political Affiliation given Economic Outlook = **0.5**
3. **Compute P(V, PA, E):**

P(V| E + PA) x P(E|PA) x P(E) = .8 x .5 x .4 = **0.16**

**Compute P(V, ~PA, E):**

P(V| E + ~PA) x P(E|~PA) x P(E) = .8 x .3 x .4 = **0.096**

.16 + .096 = **.256**

1. **Compute P(E|V)**:

P(E,V)/P(E) = .256/.4 = **.64**

And finally we compute the sum of P(V), this was really math intensive, so I solved for the value using the python model’s answer.

.4 \* .64 / .5047 = .5072

The calculated probability is 0.5072 for the prompt is [1].

# Example 2

Give an example of an input and the resulting outputs, with an explanation of the computation.

**Input and output:**

“What is the probability of a conservative political affiliation given a liberal vote and a negative economic outlook?”

The input is a negative economic outlook (0) and a liberal vote (1), which should lead to a less likely liberal affiliation (1) than in example 1.

result\_two = inference.query(*variables*=['Political Affiliation'], *evidence*={'Economic Outlook': 0, 'Vote': 1})

+--------------------------+------------------------------+

| Political Affiliation | phi(Political Affiliation) |

+==========================+==============================+

| Political Affiliation(0) | 0.6545 |

+--------------------------+------------------------------+

| Political Affiliation(1) | 0.2727 |

+--------------------------+------------------------------+

| Political Affiliation(2) | 0.0727 |

The answer to the initial question is .6545 using the python model.

**Computation:**

Based on the module 6 Bayesian Network slide:

1. **P(V=Liberal∣PA=Conservative, E=Negative)**:
   * From the CPD of Vote = **0.7**
2. **P(PA=Conservative∣E=Negative)**:
   * From the CPD of Political Affiliation given Economic Outlook = **0.6**
3. **Compute P(V, PA, E):**

P(V| E + PA) x P(E|PA) x P(E) = .7 x .6 x .6 = **0.252**

**Compute P(V, ~PA, E):**

P(V| E + ~PA) x P(E|~PA) x P(E) = .7 x .3 x .6 = **0.126**

.252 + .126 = **.378**

1. **Compute P(E|V)**:

P(E,V)/P(E) = .378/.6 = **.63**

And finally we compute the sum of P(V), this was really math intensive, so I solved for the value using the python model’s answer.

.6 \* .63 / .5775 = .6545

The calculated probability is 0.6545 for the prompt [1].

# Scaling

Imagine a real-world Bayesian network built to assess the economic impact of connected events, and implemented as in the your example. What would the main obstacles be to its practical development and use? Avoid generalities about Bayesian networks; concentrate on your application and its extrensions.

A major obstacle in developing and using a Bayesian network like the one described for assessing voter intent lies in the accurate and comprehensive data collection required to define the conditional probability distributions (CPDs). For real-world applications, especially those involving complex economic systems, the relationships between variables can be intricate and influenced by many factors. Voter behavior is influenced by complex interactions between economic indicators and political affiliations, which may vary across demographics and geographical regions. Capturing these relationships effectively requires meticulous data gathering, analysis, and domain expertise to ensure the CPDs reflect the nuanced dynamics of voter preferences amidst economic fluctuations and political developments.

Another significant challenge is the computational complexity associated with performing inference in large-scale Bayesian networks. As the number of variables and their states increases, the computational resources required to perform exact inference can grow exponentially, making it impractical for real-time or large-scale applications. Political landscapes evolve rapidly, necessitating the incorporation of up-to-date polling data, economic indicators, and demographic shifts into the network. Implementing robust data preprocessing and integration pipelines robust enough to harmonize disparate data sources and maintain the network's accuracy would be extremely difficult.

# References

[1] “Bayesian Network”. *Pgmpy Docs*. https://pgmpy.org/models/bayesiannetwork.html

# Evaluation

