Introduction to Artificial Intelligence - LAB 3

ISEP –December 4th, 2024

Instructions: Prepare an **individual** report including the **source code in Python** and the **numerical results**. Submit the report on Teams. Please respect the deadline of **December 11** to submit your report. Any delay will result in penalties. The use of existing modules on the internet is permitted provided that you cite the **bibliographic resources.**

Introduction

In this laboratory, we will analyze the use of the independence between random variables, Bayesian inference, and Bayesian networks. The work should be deposited on Teams

Part A

Bayesian Networks have shaped complicated issues with few knowledge and resources. It is incorporated into the most cutting-edge technologies of the modern day, including Artificial Intelligence and Machine Learning.

Question 1

Why is independence between random variables mandatory for Bayes' Theorem?

Question 2

Give the difference between independence and conditional independence.

Question 3

You have three coins in your pocket:

1. Coin 1 is a fair coin that comes up heads with probability $\frac{1}{2}$.
2. Coin 2 is a biased coin that comes up heads with probability $\frac{1}{4}$.
3. Coin 3 is a biased coin that comes up heads with probability $\frac{3}{4}$.

1. Suppose you pick one of the coins uniformly at random and flip it three times. If you observe the sequence HHT (where H stands for heads and T stands for tails), what is the probability that you chose Coin 3?

2. Suppose X and Y are independent random variables over the domain $1, 2, 3$ with $P(X = 3) = 1/6$. Given the following partially specified joint distribution, what are the remaining values? Write your answers as simplified fractions in the blanks.

$$P(X = 1, Y = 1) = 1/4 \quad P(X = 2, Y = 1) = 1/6 \quad P(X = 3, Y = 1) = ??$$
$$P(X = 1, Y = 2) = 1/16 \quad P(X = 2, Y = 2) = 1/24 \quad P(X = 3, Y = 2) = ??$$
$$P(X = 1, Y = 3) = ?? \quad P(X = 2, Y = 3) = ?? \quad P(X = 3, Y = 3) = ??$$

Part B

Bayesian Statistics over Bayesian Networks and Inferencing them using Pgmpy Python library: `!pip install pgmpy`

Theorem of Bayes

Bayes' theorem is a fundamental concept in Bayesian statistics, as it is used by Bayesian methods to update probabilities, or degrees of belief, following the acquisition of new data. The conditional probability of A occurring given that B occurs is represented as follows:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

The law of total probability can be used to calculate the probability of the evidence $P(B)$. If $\{A_1, A_2, \ldots, A_n\}$ is a partition of the sample space, which is the collection of all experimental results, then:

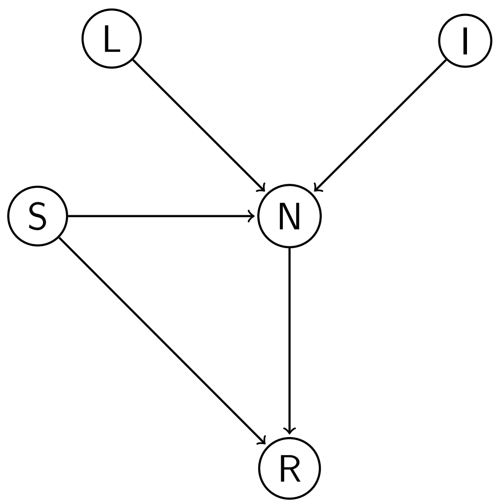
$$P(B) = \sum_i P(B|A_i) \cdot P(A_i)$$

Bayesian Network

A Bayesian network is a probabilistic graphical model, a kind of statistical model that utilizes a directed acyclic graph (DAG) to describe a set of variables and their conditional relationships. Bayesian networks are perfect for analyzing past occurrences and estimating the probability that any of numerous known causes had a role.

We can use Bayes' rule and the total probability theorem to infer probabilities in a Bayesian network. Let's consider an example: students are required to submit a project for Introduction in AI, and the Professor attempts to predict if students copied (plagiarized) the content. For this, the Professor would like to create a statistical method that can take preemptive measures based on the information given. Let's assume we have the following information:

- L: A prediction from a NLP ML (Natural Language Processing Machine Learning) model that can read the content of the Project and give a score (signalled) (probability) that this content is copied/plagiarized.
- I: Another student (colleague) marks the material as inappropriate/plagiarism.
- S: The Project was suspended before for any bad remarks.
- N: Score (Probability) that the Project should not be considered (not scored).
- R: Score (Probability) that the Student should be restricted for this lecture (ratrappage).



I	T	F
$P(I)$	0.21	0.79

L	T	F
$P(L)$	0.08	0.92

S	T	F
$P(S)$	0.12	0.88

R	T	F
$P(R\|NS)$	0.38	0.62
$P(R\|\neg S)$	0.08	0.92
$P(R\|\neg N, S)$	0.08	0.92
$P(R\|\neg N, \neg S)$	0.05	0.95

N	T	F
$P(N\|L, S, I)$	0.92	0.08
$P(N\|L, S, \neg I)$	0.88	0.12
$P(N\|L, \neg S, I)$	0.79	0.21
$P(N\|L, \neg S, \neg I)$	0.73	0.27
$P(N\|\neg L, S, I)$	0.22	0.78
$P(N\|\neg L, S, \neg I)$	0.08	0.92
$P(N\|\neg L\, negS, I)$	0.17	0.83
$P(N\|\neg L, \neg S, \neg I)$	0.03	0.97

Using Pgmpy Python Library

We will use the pgmpy Python library to compute the probabilities. You can follow the tutorial and analyze the existing functions at:

```
https://pgmpy.org/models/bayesiannetwork.html
Import the libraries :
from pgmpy.models import BayesianModel
from pgmpy.factors.discrete import TabularCPD
from pgmpy.inference import VariableElimination
import numpy as np
```

Question 1

1. Create the Bayesian network using: `bayesNet = BayesianModel()`
   - For all nodes: `bayesNet.add_node("Name_Node")`
   - For all edges: `bayesNet.add_edge("Name_Node1", "Name_Node2")`
2. Add CPDs to each node using the function `TabularCPD()`.
3. Check if the model is correctly created using `bayesNet.check_model()`.
4. Create a solver that uses variable elimination internally for inference: `solver = VariableElimination(bayesNet)`

Question 2

1. Compute the probability of "Project should be not considered (not scored)" manually and using the pgmpy library.
2. Compute the probability of "Project should be not considered (no scored) given the ML model signaled it" manually and using the pgmpy library.

Question 3

Find (in)dependencies between the variables using the function `get_independencies()`.