Assignment 5

Subject

Topics of this session :

1. Conditional probability and independence.
2. Bayes's theorem.
3. Belief Networks.

This assignement is graded and must be submitted (individually) on Moodle before next week's class.

For each exercise, detail your reflexion steps :
- We are mostly interested in your actual thinking process.
- Even if you are unable to solve an exercise, write out what were you reflexion steps.
- For each attempted exercise, a written feedback will be provided (if time alllows it).

Reference material : Artificial Intelligence, A Modern Approach, Chapters 6, 7.

For coding :
- Noto (Online Jupyter NoteBook).
- Any other python coding environment you prefer using.

Exercise 1

Let's consider the N-meteorologist problem. Assume there are $N = 3$ weather stations, which predict what is the chance that it's going to rain or not. In the beginning we believe equally each weather station forecasts.

Tasks :

1. Represent the problem as a graph, where nodes are the events, and edges represent relationships between events.
2. Express the probabilities of rain for the first day, using our initial belief, based on the following forecasts :

	Station 1	Station 2	Station 3
Rain Probability	0.5	0.7	0.05

TABLE 1 – Forecasts for the first day.

3. Using Baye's Theorem, update our belief about stations being correct, knowing that it *rained* on the first day.
4. Assuming that the weather is independent of the past, update our belief about stations being correct, knowing that it also *rained* on the second day, with the following forecasts :

	Station 1	Station 2	Station 3
Rain Probability	0.3	0.2	0.2

TABLE 2 – Forecasts for the second day.

Exercise 2

We are given a coin, but we do not know its bias toward heads or tails. Instead, we model our belief about the probability of heads, denoted as , using a Beta distribution :

$$\theta \sim \text{Beta}(\alpha, \beta)$$

where θ represents the probability of heads, and α and β are the parameters representing our belief.

Reminders :

— The posterior distribution for the Beta Distribution remains a Beta distribution with updated parameters :

$$\theta \mid \text{data} \sim \text{Beta}(\alpha + k, \beta + (n - k)).$$

where in our case, k would be the number of heads and $n - k$ the number of tails observed in the data.

— The mean of a Beta distribution Beta(α, β) is given by :

$$E[\theta] = \frac{\alpha}{\alpha + \beta}.$$

Tasks :

1. Assume we start with a uniform prior, meaning we have no prior knowledge about the coin's bias. What values should we set for α and β ?
2. Assuming coin tosses are independent, if we initially set $\alpha = 2$ and $\beta = 2$, and after 10 coin flips we observe 7 heads, compute the posterior distribution of θ.
3. Compute the mean of the posterior distribution and interpret its meaning in the context of estimating the coin's bias.
4. Implement `BetaConjugatePrior` in `coin_toss.ipynb` and observe how the belief changes as observations come. What can you say about the model after many observations ?

Project – Part 3

This whole section must be done in groups of 2-3 people.

This week, we will assume that some observations are stochastic. Our goal is to perform **Bayesian inference** to update our belief about the true model of the environment.

1 Guided Project

Dungeon Gridworld We will once more update the assumptions about the world, adding randomness in observations.

New Assumptions :

— Falling into a hole causes the game to end immediately.

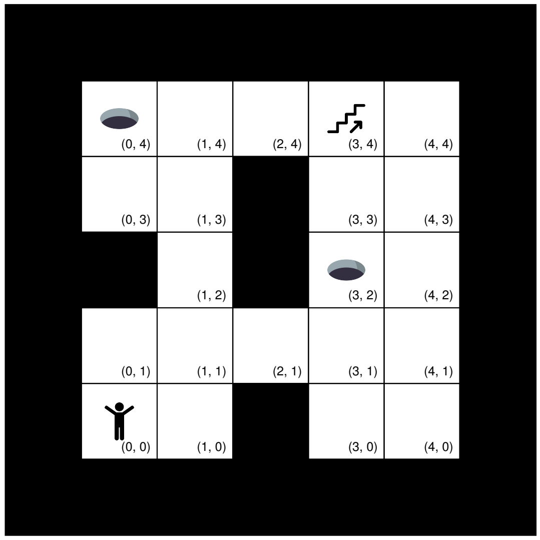

FIGURE 1 – A simple dungeon room with holes. The agent starts at position (0, 0).

— The goal is to reach the stairs (the number of steps we take does not matter this time).
— The agent **has partial and noisy observations** :
  - Whenever the agent is adjacent to a hole, it hears an *Echo* from it with probability $p_E = \frac{1}{3}$.
  - We assume we can observe perfectly our position.
  - For simplicy, we assume we do not sense anything else (no *Bump* and *Light*).
  - We have an initial belief that all tiles have a probability 0.1 to be a hole.

Tasks :

1. In `dungeon_inference.ipynb`, implement the logic for updating the belief state.
2. Ensure that the agent properly learns about the presence of holes for each tile. **Note :** Since the agent **does** not observe whether it falls into a hole, it **cannot** update its belief using that information.
3. (Optional) Try improving the agent (implement `smarter_agent()`) with a simple rule, such as : "*If I believe that a tile has no hole with probability at least 0.9, I can go there ; otherwise, I avoid it.*".

2 Personal Project

For your project, follow these steps :

1. Introduce **randomness** to certain observations and implement the corresponding changes in your environment.
2. Code a simple **random agent** that selects actions uniformly at random.
3. Define an initial belief for your agent about the environment.
4. Implement a simple **Bayesian inference method** to update the agent's belief about the environment, using data collected by the random agent.