Homework Probabilistic Reasoning

Bayes Networks (10 pts)

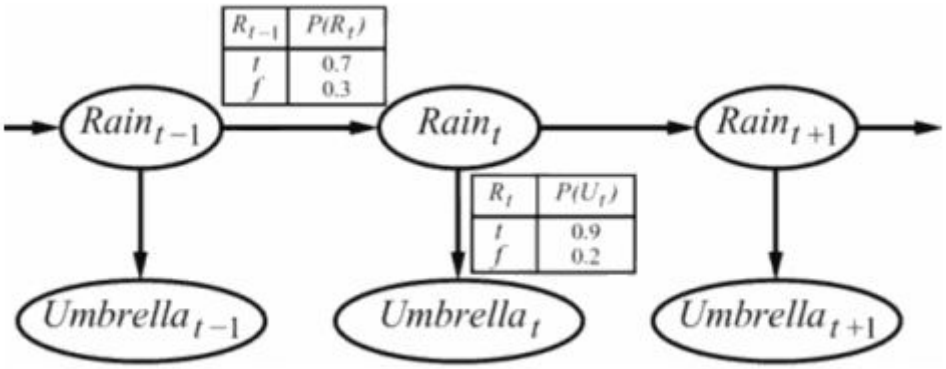
Problem:
At school, there is an alarm that rings at the top of every hour. The alarm is started by the custodian, who sometimes falls asleep and sometimes goes out to run quick errands. Hence, sometimes the alarm does not ring. More rarely, the alarm is just broken.

Assignments:

1) Define the variables that are needed to model the problem in the form of a Bayes Network. Please work ONLY with Boolean variables.
2) Draw a Bayesian network for this problem domain.
3) Suppose that the probability that the alarm works correctly when started is x if not broken (y if the alarm is broken). Write the conditional probability table for the fact that the alarm rings, conditioned on all parents in your Bayes network.
4) Suppose that the alarm always works correctly except when it is faulty, in which case it never sounds. Write the conditional probability table for the fact that the alarm rings, conditioned on all parents in your Bayes network.
5) Suppose we know that:
    a) the alarm is working properly
    b) custodian is working (not sleeping, and not running errands)
    c) alarm sounds

    Write an expression for the probability that we are at the top of the hour, showing all your steps.

Hidden Markov Models (10 pts, 1 bonus)



Problem:
Consider the depicted Hidden Markov Model, with transition and observation model as specified in the network tables.

Assigments:

1) Implement the model in C++ (allowed library for matrix computation: Eigen) or Python (allowed library for matrix computation: numpy).
2) Implement in the chosen programming language direct sampling, as studied in class, and use it to sample at least 15 sequences of length 20.
   Note: store both the sequence of states and observations.
3) Implement in the chosen programming language the basic version of the forward-backward algorithm, as studied in class, and test it over one or more sequences that you generated.
   Note: verify how good your estimate of the state is using the state sequence stored in point 2.

Bonus Points (1 pt): Extend the implementation of the backward-forward algorithm using one of the improvements presented in class.

Markov Decision Processes (10 pts, 3 bonus)

Problem:
Choose one of the following environments from OpenAI Gym Atari [https://gym.openai.com/envs/#atari] with the proposed simplifications:

- Breakout-v0
- MsPacman-v0 (with static ghosts)
- SpaceInvaders-v0 (with static invaders)

Assignment:
Implement the chosen environment (using the same interface as the one in Gym [https://gym.openai.com/docs/#environments]) in C++ or Python according to the following specifications:

- The environment is FULLY observable.
- No graphical rendering is needed.
- Use ONLY discrete states (i.e., instead of images, you will return the real environment state as observation).
- Use a reduced state space (e.g., the state of the grid should be small - at most 10x10). Adapt the number of walls, ghosts or invaders accordingly.
- Use ONLY discrete actions.
- Except the agent (i.e., the player) everything is STATIC.
- Write a simple test program where you create an instance of the enironment and you execute a sequence of at least 5 random actions, printing (or showing in any form you prefer) the observation you get out of it.

Bonus Points (3 pts): Expose the transition model from the environment to the agent, implement the value iteration algorithm and run it on your environment.