AI ASSIGNMENT 3 — Uncertainty, Bayesian Nets, HMM and Kalman Filtering
Deadline : 11:59 PM, 24/11/2024

Total Marks : 90 Marks **Weightage: 10%**

Instructions:

1. Assignments are to be attempted individually.
2. Submit the assignment as a single zipped folder (**A3_⟨RollNumber⟩.zip**) containing a pdf file `Report.pdf` for all the theory questions, graphs, and code analysis, and relevant python code files for programming questions as per the format provided in the Coding section of the assignment.
3. Please read the instructions given in the questions carefully. In case of any ambiguity, post your queries on Google Classroom at least a week before the deadline. **No TA will be responsible for responding to the queries after this.**
4. A part of the assignment evaluation involves automatic testing of your submitted code on private test cases. **Please make sure that you do not change the structure of the methods provided in the boilerplate code.**
5. All the TAs will strictly follow the rubric provided. **No requests will be entertained related to scoring strategy.**
6. **The use of generative tools (such as ChatGPT, Gemini, etc.) is strictly prohibited.** Failure to comply may result in severe consequences related to plagiarism.
7. **Extension and Penalty clause:**
   - Even a 1 minute late submission on google classroom will be considered as late. Please turn-in your submissions atleast 5 minutes before the deadline.
   - Not explaining the answers properly will lead to zero marks.

Theory (30 marks)

1. (10 marks) Based on the following dataset answer the below questions:
   - About 80.0% of people prefer to travel by air or train.
   - Of the people who prefer air travel, 20% travel for business and 30% travel for leisure.
   - Given that a person travels by train, the chance they are traveling for leisure is 0.400, rounded to 3 decimal places.

- About 25% of people prefer to travel by car and have reported feeling stressed during their travels.
- There is a 0.015 probability that a person prefers to travel by bus and has a low-stress level.
- Given that a person prefers traveling by bus, the probability that the person is traveling for business is about 0.350, rounded to 3 decimal places.
- The probability that a person feels stressed and prefers air travel is 0.065.
- About 70% of people travel for either leisure or business.
- There is a 60% chance that a person prefers air travel given that they are feeling stressed.
- There is a 50% chance a person prefers train travel whether they travel for business or leisure.

(a) (2.5 marks) Compare direct sampling, rejection sampling, and Gibbs sampling in the context of estimating probabilities from the given travel dataset. Compare their strengths and weaknesses.

(b) (2.5 marks) Suppose you want to estimate the probability of a person traveling by train for leisure, where you know that the chance of leisure travel given train preference is 0.400. If you sample 100 people randomly and 30 of them prefer train travel, how many should you expect to accept as travelers for leisure based on this probability? Provide your calculations.

(c) (2.5 marks) Given that 80% of people prefer air travel, and of those, 20% travel for business, calculate the probability that a randomly selected person prefers air travel and travels for business. Show your calculations and round to **three decimal places**.

(d) (2.5 marks) How does increasing the sample size affect the accuracy and precision of estimates obtained through direct sampling? Discuss the implications for the given dataset.

2. (10 marks) Given the following statements below, answer the questions. Round off the probability values computed to 3 decimal places. A statement may have more than one proposition.

- About 91.0% of people either read books or access academic journals regularly.
- Of the people who read books, 40% also access academic journals, and 60% only read books.
- Given that a person reads books, the probability that they participate in book clubs is 0.320, regardless of whether they access academic journals or not.
- About 22.7% of people access academic journals but do not read books.
- There is a 0.090 probability that a person neither reads books nor accesses academic journals.

- Given that a person does not read books, the probability that they access academic journals is 0.716.
- The probability that a person participates in book clubs and accesses academic journals is 0.088.
- About 63.1% of people either participate in book clubs or access academic journals.
- There is a 40.0% chance that a person accesses academic journals, given that they participate in book clubs.
- There is a 50.0% chance that a person accesses academic journals, whether or not they read books.
- Given that a person does not read books, the probability that they participate in book clubs is 0.0044, regardless of whether they access academic journals or not.

  (a) (2.5 marks) Identify the random variables in the statements above and write each statement using symbols for random variables, logical connectives where necessary, and conditional probability notation.

  (b) (2.5 marks) Verify that these propositions create a valid probability distribution. List the set of axioms that they satisfy.

  (c) (2.5 marks) Populate the full joint probability distribution table.

  (d) (2.5 marks) Use the joint distribution table and check for conditional independence between all the random variables that you have identified.

3. (10 marks) In the context of adversarial machine learning, consider a machine learning model used for image classification. Two different types of adversarial attacks can cause the model to misclassify an input: **adversarial perturbations** (small, imperceptible modifications to input data) and **backdoor attacks** (where the model is trained to misclassify inputs that contain a specific trigger). Both types of attacks can trigger a misclassification, leading to a "misclassification alarm" being raised.

   Now, suppose you observe a misclassification alarm after querying the model with an input. Initially, adversarial perturbations and backdoor attacks are considered independent events. However, you come across a report that backdoor triggers have been increasingly present in recent datasets. How does this new information about the prevalence of backdoor attacks change your belief regarding the likelihood of adversarial perturbations causing the misclassification?

   (a) (5 marks) Formulate this problem using Bayesian inference.

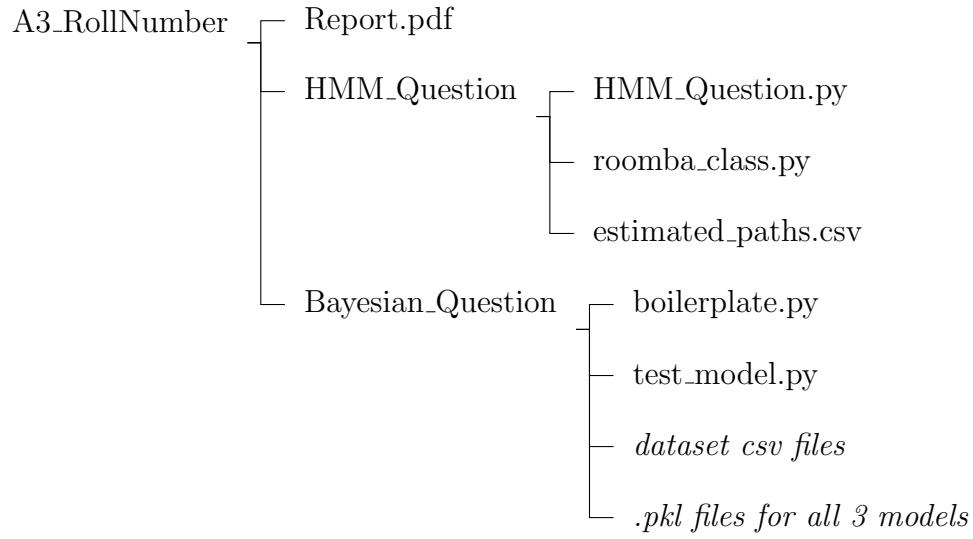
   (b) (2.5 marks) Define the probabilities involved (prior, likelihood, and posterior).

   (c) (2.5 marks) Explain how conditioning on the detection of a backdoor attack (from recent reports) changes your belief about the role of adversarial perturbations in causing the misclassification.

   Hint: Consider how the observation of the common effect (the misclassification alarm) influences your belief about the independent causes (adversarial perturbations and backdoor attacks).

Coding (60 marks)

Use the provided `requires.txt` file to setup your environment for both the coding questions. Your code should execute without any errors in this environment; otherwise, you will not be marked. Follow the below folder structure for submission.



4. (30 marks) **Bayesian network for fare classification** In this assignment, you will develop a Bayesian network model for fare classification using a public transportation dataset. The dataset contains information about different bus routes, stops, distances, and fare categories. Your task is to build an initial Bayesian network, then apply pruning techniques to improve the model's efficiency, and optimize it further using structure refinement methods. All 3 models need to be evaluated on the validation set provided. The models will be tested on the private test set for final evaluation. Return the `.pkl` files for each model along with your report.

   Boilerplate code and dataset can be accessed here

Dataset Features

You are expected to use the following features for constructing the Bayesian network:

- **Start Stop ID (S)**: The stop ID where the journey begins.
- **End Stop ID (E)**: The stop ID where the journey ends.
- **Distance (D)**: The distance between the start and end stops.
- **Zones Crossed (Z)**: The number of fare zones crossed during the journey.
- **Route Type (R)**: The type of route taken (e.g., standard, express).
- **Fare Category (F)**: The fare category for the journey, classified as `Low`, `Medium`, or `High`.

The objective is to classify the fare for a journey between a given start and end stop as one of the following categories: `Low`, `Medium`, or `High`. You will build a Bayesian network based on other features and use it to predict the fare category.
Hint: use the `bnlearn` library imported in the boilerplate code for constructing, training and testing the bayesian network. Some examples showcasing how to use the library can be found here.

Testing

The code for evaluation is provided to you along with the boilerplate code in `test_model.py`. Test your models on the validation subset and report accuracies in the assignment report. You will have to write your own code for calculating runtimes for each network construction (initialization and training). For evaluation purposes, DO NOT FORGET TO RETURN the `.pkl` FILES along with your code and report.

Tasks

1. **Task 1: Construct the initial Bayesian Network (A) for fare classification.** (10 Marks)
   (a) Build the Bayesian network using the provided features.
   (b) Ensure that the structure includes dependencies between all possible feature pairs.
   (c) You should provide a visualization of the initial Bayesian network in the assignment report.
2. **Task 2: Prune the initial Bayesian Network (A) to enhance performance.** (10 Marks)
   (a) Apply pruning techniques such as Edge Pruning, Node Pruning, or simplifying Conditional Probability Tables (CPTs).
   (b) Clearly explain the pruning method applied and how it improves the model's efficiency (time taken to fit the data) and/or prediction accuracy.
   (c) Provide a visualization of the pruned Bayesian Network (B) with fewer edges or simplified structure.
3. **Task 3: Optimize the Bayesian Network (A) by adjusting parameters or using structure refinement methods.** (10 Marks)
   (a) Apply optimization techniques such as structure learning (e.g., Hill Climbing) to refine the Bayesian network structure.
   (b) Compare the performance of the optimized Bayesian network with the initial network (A) and explain how the optimization improves the model's accuracy and/or efficiency.
   (c) Provide a visualization of the optimized network.

5. (30 marks)**Tracking a Roomba Using the Viterbi Algorithm** Imagine you have a Roomba robotic vacuum cleaner that autonomously cleans your home while you're away. The Roomba operates based on specific movement policies specified in the Roomba class member functions.

   You've installed sensors that provide noisy observations of the Roomba's location at discrete time intervals. Due to sensor limitations, these observations are not always accurate. Your goal is to model the Roomba's movement using a Hidden Markov Model (HMM) and implement the Viterbi algorithm to track its most likely path based on the noisy sensor observations. **Boilerplate code can be accessed here**

   - **Environment: (Present in boilerplate code)**
     - Your home is represented as a grid of size 10×10.
     - Possible headings the Roomba can take are: North (N), East (E), South (S), West (W).
     - Only obstacles are the 4 walls of your home.
   - **Roomba Movement Policy:**
     1. Random Walk Policy
        - The Roomba takes one unit per time step in either direction.
        - After each step it randomly selects a new heading from the available set of directions for the next step.
        - It continues moving until it reaches the final destination or runs out of time.
     2. Straight Until Obstacle Policy
        - The Roomba moves one unit per time step in its current heading unless an obstacle blocks its path.
        - Upon encountering an obstacle, it randomly selects a new heading from the available directions and continues moving.
        - The movement is deterministic unless an obstacle triggers a heading change.
   - **Sensor Observations:**
     - At each time step, sensors provide a noisy observation of the Roomba's location.
     - Observations are modeled as the true position plus Gaussian noise with mean zero and standard deviation $\sigma = 1.0$.
   - **Given Code:**
     - The Roomba's two movement policies are implemented inside the `Roomba` class. The path using both policies is simulated for T=50 time steps and is already provided.
     - Noisy observations for both policies based on the Roomba's true positions are already calculated for you.

- **Tasks:**
  (a) Model the problem as a Hidden Markov Model, specifying the state space, transition probabilities, and emission probabilities.
  (b) Implement the Viterbi algorithm to estimate the most likely path of the Roomba given the observations.
  (c) (10 marks per seed value) You can change the seed value in the `setup_environment()` function. This will generate new observations. For at least 3 different seed values do the following:
    – Use given code to setup environment, get the true path and noisy observations for the specified seed value.
    – Estimate the Roomba's path given the noisy observations generated previously using the Viterbi Algorithm. Save the estimated path for evaluation.
    – Compare the estimated path with the true path using the given `evaluate_viterbi()` function to compute the tracking accuracy.
    – Analyze which policy is more accurate and why.
    – Plot the true path, observed positions, and estimated path using the given `plot_results()` function. Include plots in the assignment report.
  (d) State your selected seed values clearly in the assignment report. Your code will be executed for multiple seed values for evaluation so your viterbi algorithm should work without any errors in the given environment. Any possible exceptions should be handled carefully.
  (e) Create a `estimated_paths.csv` file with your chosen seed values and the corresponding estimated path. This will be compared later with the estimated path generated by your code for evaluation so DO NOT MAKE ANY FORMATTING CHANGES. First column should be the seed value, second column the policy name and third column the estimated_path variable.