Extra Credit Homework

Abde Manaaf Ghadiali (G29583342)

Q1 - Carl's Diner

Disclaimer:

1. ***While executing the code for a higher cutoff of inventory size and max coffee bags to order, I was facing resource allocation error as the state space exploded to more than 20000 states.***
2. ***I discussed a few points with Gehna Ahuja specifically on the state space definition, transition probability matrix creation and algorithm concepts of Value Iteration.***
3. ***I have created a .py file but would suggest looking at my .ipynb notebook as it is more comprehensive with the outputs already there.***

Our State Space (S) is a combination of all possible inventory sizes and number of customers for the day (ranging from 0 to 400 both inclusive).

- Our inventory size is the number of (**Fresh**) coffee bags at the start of the day ranging from 0 to 25 (both inclusive).
- We then get our State Space Matrix by one-hot encoding the state space.

Our Action Space (A) is how many coffee bags we need to order for the next day based on today's state (day end inventory and number of customers that came in today).

Creating our Reward Vector for each action in the Action Space

Algorithm:

- For each action in the A do the following:
    - Calculate the Action Cost (Number of coffee bags ordered x Cost of a Single Bag).
    - Calculate the number of bags that don't meet the standard (Using Binomial Distribution) (this will be a random number in the Binomial Distribution).
    - Get the Usable Coffee Bags from the total number of coffee bags ordered.
    - Now for each state in the S do the following:
        - Calculate New Inventory = current inventory + Usable Coffee Bags
        - Calculate Cups Sold, Unsatisfied Customers
        - Based on above, Calculate revenue reward
        - Finally subtract the action cost from revenue (This becomes our reward for this state)

This creates our Reward vector which we perform for all the actions in A and stores them in a dictionary.

Calculating Transition Probabilities

Algorithm:

- For each action in the A do the following:
  - Random probability of coffee bags that do not meeting the standard
  - For all states (inventory, customers) in the S do the following:
    - New Inventory = Old Inventory + Action - Bags not meeting the standard (Using Binomial Distribution) (Similar to the calculation in reward)
    - Calculate Leftover Bags
    - Calculate the Stale Probability of the coffee bags counting from 0 to leftover bags
    - For each next day customer, do the following:
      - Get the Customer Probability
      - Get next states (After subtracting Leftover with stale bags)
      - Calculate the Probability as: Customer Probability x (1 - Stale Probability) x (1 - Not Standard Probability)

This creates our Transition Probability Matrix which we perform for all the actions in A and stores them in a dictionary.

After this we perform our Value Iteration and Policy Iteration Algorithms to get the most optimal policy for Carl to order coffee bags for the next day.

Q2 - Feedback

Homework 4 – Question 4 (Inference By Variable Elimination)

I found the description in the Homework Question to be a bit incomplete. Though the description of the function in the provided code was a bit useful but still a bit misleading when it says we can refer to the previous function (Inference By Enumeration) as that function does the elimination very differently.

Question Description:

Implement the Inference by variable elimination algorithm to compute marginal probabilities of certain variables given evidence. Systematically eliminate variables by joining and marginalizing factors associated with each variable in a predefined elimination order.

Integrate provided evidence by filtering the factors, ensuring that the computations respect the observed evidence. After processing all variables, normalize the resulting factor to produce a normalized probability distribution, representing the marginal probability of the target variables.

Algorithm:

- Get all conditional probability tables (CPTs) from the Bayesian Network that are consistent with the given evidence dictionary.
- For each variable in the elimination order:
  - Store factors involving the current variable
  - Store factors that do not involve variable
- For each factor in the list of factors:
  - If the current variable var is present in the factor's variable set, add the factor to a list (say list_1)
  - Otherwise, add the factor to another list (say list_2)
- Join all factors in the list_1 using the current variable, resulting in a joined factor.
- If the joined factor has more than one unconditioned variable:
  - Eliminate the current variable from the joined factor to simplify it.
  - Add the simplified factor to list_2.
- Replace the factors list with the updated list_2
- Join all remaining factors into a single factor
- Normalize this final factor to ensure it represents a valid probability distribution