DSE 3260 ARTIFICIAL INTELLIGENCE LABORATORY

6th Semester BTech Data Science & Engineering

SECTION B

AI Laboratory Manual

WEEK 1: Tutorial on OpenAI Gymnasium

1. Take the tutorial: Getting Started With OpenAI Gym: The Basic Building Blocks https://www.gymlibrary.dev/content/tutorials/ https://blog.paperspace.com/getting-started-with-openai-gym/
2. Use the CartPole-v0 environment and write a program to :

a. Implement the CartPole environment for a certain number of steps
b. Implement the CartPole environment for a certain number of episodes
c. Compare and comment on the rewards earned for both approaches.
d. Plot the cumulative reward of the games and note down the results.

WEEK 2: Problem Solving Agents & SEARCH, POLICY

QUESTION 1: The Game :

According to the "Six Degrees of Kevin Bacon" game, anyone in the Hollywood film industry can be connected to Kevin Bacon within six steps, where each step consists of finding a film that two actors both starred in. To solve the problem, find the shortest path between any two actors by choosing a sequence of movies that connects them. For example, the shortest path between Jennifer Lawrence and Tom Hanks is 2: Jennifer Lawrence is connected to Kevin Bacon by both starring in "X-Men: First Class," and Kevin Bacon is connected to Tom Hanks by both starring in "Apollo 13."

Problem Solving Agent:

Given two actors nodes in the graph we need to find the distance (shortest path) between the nodes. Write a python program to determine how many "degrees of separation" apart two actors are. Find the distance or the degree of separation., using

a. Breadth first search

b. Depth first search

Distribution Code: Data & Download the distribution code from:
https://cdn.cs50.net/ai/2020/x/projects/0/degrees.zip
The distribution code contains two sets of CSV data files: one set in the large directory and one set in the small directory. Use the small dataset for ease of testing and experimentation. Each dataset consists of three CSV files.

1. **small/people.csv**: each person has a unique id, corresponding with their id in IMDb's database. They also have a name, and a birth year.

2. **small/movies.csv**: You'll see here that each movie also has a unique id, in addition to a title and the year in which the movie was released.

3. **small/stars.csv**: This file establishes a relationship between the people in people.csv and the movies in movies.csv. Each row is a pair of a person_id value and movie_id value. For example: The first row (ignoring the header), for example, states that the person with id 102 starred in the movie with id 104257. Checking that against people.csv and movies.csv, you'll find that this line is saying that Kevin Bacon starred in the movie "A Few Good Men."

4. **degrees.py**: At the top, several data structures are defined to store information from the CSV files. The names dictionary is a way to look up a person by their name: it maps names to a set of corresponding ids (because it's possible that multiple actors have the same name). The people dictionary maps each person's id to another dictionary with values for the person's name, birth year, and the set of all the movies they have starred in. And the movies dictionary maps each movie's id to another dictionary with values for that movie's title, release year, and the set of all the movie's stars. The load_data function loads data from the CSV files into these data structures. The main function in this program first loads data into memory (the directory from which the data is loaded can be specified by a command-line argument). Then, the function prompts the user to type in two names. The person_id_for_name function retrieves the id for any person (and handles prompting the user to clarify, in the event that multiple people have the same name). The function then calls the shortest_path function to compute the shortest path between the two people, and prints out the path.

Code the shortest_path function using
a. Breadth First Search
b. Depth First Search

QUESTION 2: Use the CartPole-v0 environment, the status of the system is specified by an "observation" of four parameters (x, v, θ, ω), where
x: the horizontal position of the cart (positive means to the right)
v: the horizontal velocity of the cart (positive means moving to the right)
θ: the angle between the pole and the vertical position (positive means clock wise)
ω: angular velocity of the pole (positive means rotating clock-wise)
Use gym and write the following policies for 100 episodes :
a. Theta policy: if the pole is tilted to the left (θ<0), then push the cart to the left and vice versa.
b. Omega policy : when the pole is moving away from the vertical position (ω<0) then push the cart to the left and vice versa.
c. Plot the cumulative rewards (based on steps) for each policy , write down the average rewards with standard deviation.
d. Comment on performance of policies.

WEEK 3: Gaming Agent & Negamax search

EasyAI is an artificial intelligence framework for two-players abstract games. Read through the documentation.

http://zulko.github.io/easyAI/index.html

Write a python program to define and implement a tic—tac-toe game with one human player. Solve the game using the built in algorithms and compare the solutions.
a. Iterative Deepening
b. Depth first search

WEEK 4: Travelling Salesman Problem and Local Search

QUESTION 1:

The Traveling Salesman Problem (TSP) is: *"Given is a list of cities and distances between each pair of cities - what is the shortest route that visits each city and returns to the original city?"* Consider the symmetric variant of the TSP (sTSP), where the cost from city j to city i equals the cost from city i to city j.

1. Download the TSPLIB dataset, which contains graphs with cities as nodes & the distances between them, either in tsp or XML format
   http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/
   The documentation is available in :
   http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/tsp95.pdf

2. Consider the following datasets: **rd100.tsp, eil101.tsp, a280.tsp, d198.tsp and ch150.tsp**. For each dataset:
   a. compute the distance matrices between cities.
   b. Assess the performance of three Hill Climbing Algorithms in solving the TSP.
      i. Simple Hill Climbing
      ii. Stochastic Hill Climbing
   c. Tabulate results and comment on the best local search method for all datasets.

WEEK 5 – MULTI-ARMED BANDITS – AD OPTIMIZATION

Consider the given dataset "Ads_clicks" containing data about which add was clicked in each time step. Suppose an advertising company is running 10 different ads targeted towards a similar set of population on a webpage. We have results for which ads were clicked by a user Each column index represents a different ad. We have a 1 if the ad was clicked by a user and a 0 if it was not.

a. Write down the MAB agent problem formulation in your own words.

b. Compute the total rewards after 2000-time steps using the ε-greedy action. a. for ε=0.01, ε= 0.3

c. Compute the total rewards after 2000-time steps using the Upper-Confidence-Bound action method for c= 1.5

d. For all approaches, explain how the action value estimated compares to the optimal action.

WEEK 6 – MULTI-ARMED BANDITS- MOVIE RECOMMENDATION

a. Write down the “Movie Recommendation” as a Reinforcement Learning problem formulation. Use comment lines for proper documentation.
b. For the environment, use the built-in TF-agent agent, MovieLensPyEnvironment (non-per-arm)
https://github.com/tensorflow/agents/blob/master/tf_agents/bandits/environments/movielens_py_environment.py
c. Compute the regret using the built-in metric in TF-agents
https://github.com/tensorflow/agents/blob/master/tf_agents/bandits/metrics/tf_metrics.py
d. Plot the regret against 20,000-time steps using the built-in agents:
    I. LinUCB
    II. LinTS
    III. NeuralEpsilonGreedy

    And identify the best agent for the movie recommendation.
e. Write the Recommendation policy; given a new observation request (i.e., a user vector), the policy will produce actions, which are the recommended movies.