Phase 2: Movie Recommendation

(Q-learning)

Habiba Arafa 202201684

Shahd Hossam 202100936

Bosy Ayman 202202076

1. Introduction

In this report, we introduce the implementation of a movie recommendation system using Q learning. The idea is to suggest the most similar movie by exploring and exploiting cosine similarity between each input and the other states to provide a suitable reward in each new state.

2- Design

State: movies vectors where each movie has its vector, including zeros and one which is stored in a JSON file representing genre, and rate.etc.

Action: Toggle a bit in the vector to go into another state (movie).

Cosine Similarity: Being used as the primary measure of similarity between movies.

Reward: The reward is the similarity score between the next movie's vector (after the action) and the current movie's vector.

Q-table: A table where each movie's vector has associated Q-values for each action (bit toggling). The Q-values are updated as the agent explores and exploits.

Alpha (Learning Rate): The rate at which the Q-value is updated based on new experiences. Set to 0.1.

Gamma (Discount Factor): The factor that discounts future rewards.

Epsilon (Exploration Rate): The probability of taking a random action instead of exploiting the current best-known action.initial value is 0.1.

We applied 1000 episodes, and the Q table is updated during each episode and step, At each step, the agent either explores (chooses a random action) or exploits (chooses the best-known action based on the Q-table).

3-Q learning setup

Training Phase:

First, the Q table is initialized by zeros, the actions are toggles (based on selected index). Then, a random movie is selected as a starting point, and the process continues until the specified episodes are reached.

The agent explores at each step by either choosing a random action (explore) or choosing the best action (exploit) based on the current Q-table.

an action is performed by toggling a bit in the movie's vector, generating a new movie vector.

After applying the action, the system calculates the cosine similarity between the new movie vector and all other movie vectors.

-The movie with the highest similarity is selected as the next recommendation.

The reward is the similarity score of the recommended movie compared to the original movie vector.

The following rule is used to update the Q-table after each action:

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma \, maxQ(s',a') - Q(s,a))$$

r is the reward, s' is the next state, and a' is the next action.

The exploration stops when all movies has been visited or when the maximum number of episodes is reached, and the **q_table_trained.json** is performed in the training phase.

Testing Phase:

Once the training was completed, the Q-learning model underwent testing using various input movies to assess its effectiveness. The system aimed to recommend a sequence of movies that closely matched the test input over 10 steps. The testing process involved the following steps:

1. **Setup**:
   - A test movie was chosen as the starting point.
   - The corresponding vector for the selected movie was retrieved from the dataset stored in the JSON file.
2. **Recommendation Process**:
   - At each step, the agent determined an action based on the Q-table by selecting the action with the highest Q-value (exploitation).
   - The chosen action modified a bit in the movie's vector, resulting in a new state.
   - Cosine similarity was calculated between this new vector and the vectors of all other movies in the dataset.
   - The movie with the highest similarity score, excluding previously visited movies, was identified as the next recommendation.
3. **Path Tracking and Evaluation**:
   - The sequence of recommended movies was recorded for analysis.

- ○ Cosine similarity scores were tracked to verify the relevance of each recommendation and how well the system maintained high similarity across steps.

4- Results

Episodes:

Multiple episodes (1000) were constructed on which the system was trained

Convergence:

It gave Q-values that were gradually converged, indicating that the agent learned effective actions to maximize similarity-based rewards.

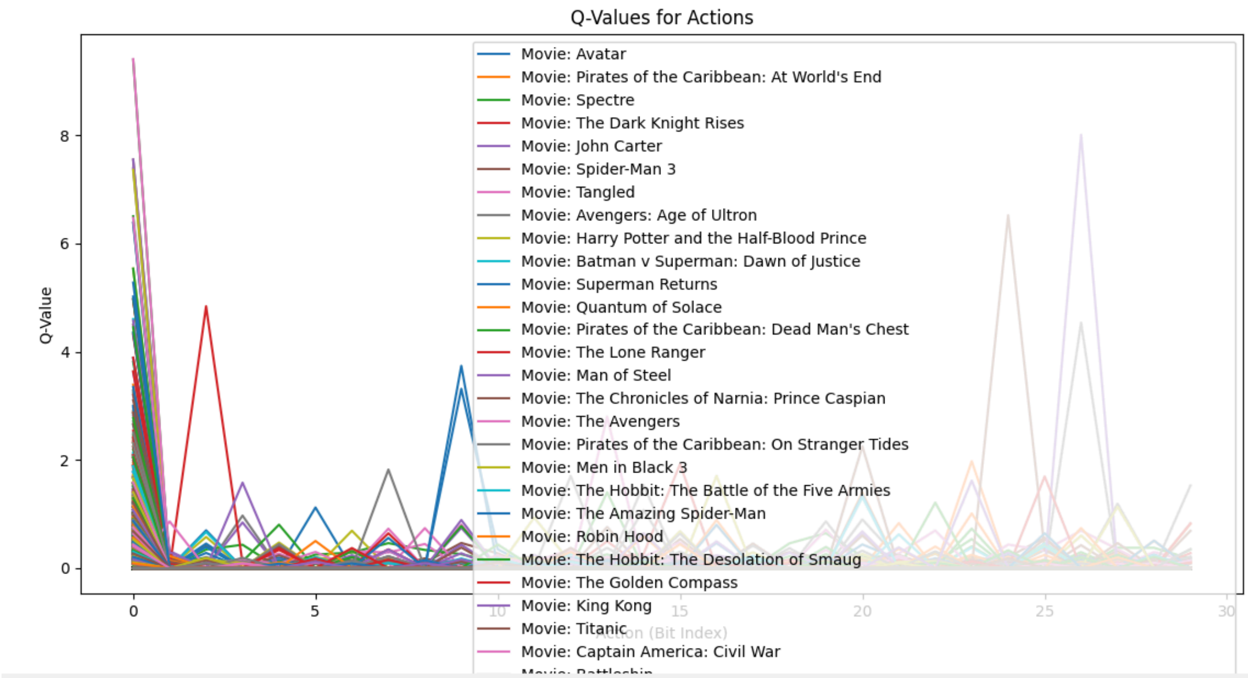
Policy Evaluation:

The trained Q table ensures a successful recommendation based on the highest cosine similarity score available over successive steps to maximize the best reward.

5- Performance Monitoring/ Analysis

Visualization

- **Q-Values Over Actions :** The final Q-values show how the agent prioritizes actions for each movie.



Side by side comparison of the different algorithms