Artificial Intelligence – Week 12 (Netflix RL)

Instructor: Teerapong Panboonyuen

Course Repository: **https://github.com/kaopanboonyuen/SC310005_ArtificialIntelligence_2025s1**

🎯 Objective

This week, you will explore **reinforcement learning (RL)** using the `RL_Netflix_Users.csv` dataset. Your goal is to build an agent that **recommends a movie genre** to a user and **learns from feedback** (reward = 1 if the recommended genre matches the user's favorite genre, 0 otherwise).

You will experiment with multiple RL algorithms and compare their performance to understand **policy learning**, **exploration vs. exploitation**, and evaluation metrics.

📚 Dataset

Path: RL_Netflix_Users.csv

https://github.com/kaopanboonyuen/panboonyuen_dataset/blob/main/public_dataset/RL_Netflix_Users.csv

RL Concept Mapping:

- **State:** User_ID (or user features like Age, Watch Time)
- **Action:** Recommend a genre (from `Favorite_Genre` unique values)
- **Reward:** 1 if the recommended genre matches the favorite, 0 otherwise

---

🏗️ Assignment Tasks

1️⃣Environment Setup

- Design a **custom RL environment** similar to RetailEnv from last week.
- The environment should use `User_ID` for states and `Favorite_Genre` for actions.
- Each episode can be a single recommendation per user.

2️⃣Implement RL Algorithms

Students are required to **implement at least 3 algorithms**:

1. **Q-Learning** (tabular)
2. **Actor-Critic** (model-free policy gradient)
3. **PPO** (proximal policy optimization)

Optional: Implement a model-based algorithm (e.g., Dyna-Q) for extra credit.

3️⃣Training

- Train each RL algorithm on the environment.

- Tune hyperparameters such as learning rate, epsilon (for Q-learning), and the number of episodes.

4️⃣ Evaluation

Compare all algorithms using **RL-specific metrics**:

- Average reward per episode
- Max / Min reward
- Stability (standard deviation of rewards)
- Success rate (%)
- Convergence speed (episodes to learn effective policy)

Optional (advanced):

- Policy accuracy against an "optimal" policy if known

Summarize results in a **table and/or plots** to identify which algorithm performs best.

5️⃣ Inference / Recommendation

- Load trained model or Q-table.
- Pick **random users** from the dataset and generate **genre recommendations**.
- Evaluate **reward/accuracy** for multiple users.

6️⃣ Deliverables

- Colab Notebook including:
  - Environment class definition
  - Training for all algorithms
  - Evaluation metrics & comparison table
    Inference examples

- Extra points for:
  - Implementing Dyna-Q or other model-based algorithms
  - Exploring hyperparameter variations
  - Clear visualizations of reward trends

💡 Tips for you guys

- Clearly define **state, action, and reward**.
- Compare **model-free** vs **model-based** algorithms.
  Experiment with **different hyperparameters** and **episode counts to optimize performance**.
- Include **plots, tables, and concise explanations** of results.

NETFLIX