Adaptive Spotify Music Recommendation using Reinforcement Learning

1 Problem to be Solved

The goal is to build an adaptive Spotify music recommendation using RL that dynamically suggests songs based on the user's listening behavior. Unlike static recommendation models, the system will use Reinforcement Learning (RL) mainly Deep RL to continuously improve recommendations based on user feedback.

2 Uncertainties Involved

User preferences changes over time : User taste in music can evolve based on mood, time of day, and external influences.

Feedback is not accurate : Skipping a song does not always mean dislike, and listening to a song does not always mean preference.

Exploration vs. Exploitation tradeoff : Should the system play familiar songs to keep engagement high or explore new songs to learn better preferences.

Sparse and Delayed Rewards : The system only receives feedback after a song is played, and playlists influence long-term engagement.

As user will be interacting with environment, this will lead to uncertainty.

3 Why the Problem is Non-Trivial?

Huge Action Space : There are 1.2 million unique songs, making traditional Q-learning infeasible.

Uncertain User Behavior : There is no explicit function to predict how a user will respond to a song.

Sequential Decision-Making : The system must learn from past interactions and adjust recommendations over time.

4 Existing Solution Methods

Collaborative Filtering : Learns from similar users choices. Fails for new users (cold start problem) and does not adapt to real-time user behavior.

Content-Based Filtering : Uses song features (genre, tempo, etc.) to make recommendations and therefore it struggles with exploration always recommends similar songs.

Multi-Armed Bandits - MABs : Learns to recommend based on immediate rewards and does not consider long-term playlist optimization.

Deep Q-Networks (DQN): As it is a value-based algorithm and has to approximate all Q-values for each action on song, the network becomes too big and not feasible as action space is huge. For eg 10k songs with 5 action possible on each song the output layer will have 50k nodes.

5 Plan for Modeling and Solving the Problem

State Space (s): The system represents the user's listening history & song features and audio features in form of a vector.

Action Space (a): The system selects the next song from possible songs list. Instead of explicitly choosing, the policy samples were from a learned probability distribution over song embeddings.

Observations (o): The system observes user feedback after recommending a song (*skip, like, listening duration*) and this feedback updates the state space.

Reward Function (r): User likes the song and listens to the full song (positive reward), if user skips the song(negative reward) and if user skips immediately(high negative reward)

Algorithm to be used: Proximal Policy Optimization (PPO) with Actor-Critic architecture (to estimate value functions and improve policy). Since we have a large action space due to the number of songs, a policy-based algorithm will be the most feasible option to work with.

6 Conclusion

Therefore music recommendation problem is a classic Sequential Decision Problem as it involves repeated interactions with the user. The system must adapt based on feedback over time and there is always an uncertainty in user preferences that changes after each recommendation.