RecoStyle

March 23, 2024

Problem Statement

Develop a Deep RL-based product recommendation system that maximizes user engagement and conversion rates on Amazon's ecommerce platform, while respecting user-defined budget constraints.

Dataset - Amazon Fashion Review Dataset

1. **Size:** 142.8 million reviews (out of which 828,699 rows considered) spanning May 1996 - July 2014.
2. **Description:**
    a. Reviews (ratings, text, helpfulness votes),
    b. Product metadata (descriptions, category information, price, brand, and image features)
        i. Product information, e.g. color (white or black), size (large or small)
        ii. package type (hardcover or electronics), etc.
        iii. Product images that are taken after the user received the product.
        iv. Bullet-point descriptions under product title.
        v. Technical details table (attribute-value pairs).
    c. links (also viewed/also bought).

3. **Data Format:** One-review-per-line in json

{"image":
["https://images-na.ssl-images-amazon.com/images/I/71eG75FTJJL._SY88.jpg"],

"overall": 5.0,

"vote": "2",

"verified": True,

"reviewTime": "01 1, 2018",

"reviewerID": "AUI6WTTT0QZYS",

"asin": "5120053084",

"style": { "Size:": "Large",

"Color:": "Charcoal" },

"reviewerName": "Abbey",

"reviewText": "I now have 4 of the 5 available colors of this shirt... ",

"summary": "Comfy, flattering, discreet--highly recommended!",

"unixReviewTime": 1514764800 }

Abstract

This project aims to develop a product recommender system using Deep Q-Network (DQN) algorithm. The system is designed to recommend products to users in an e-commerce platform, optimizing long-term rewards based on users' interactions with the recommended items. The Markov Decision Process (MDP) is modeled within the OpenAI Gym environment, providing a standardized framework for RL experimentation and evaluation.

OpenAI Gym is utilized to define a custom environment, named RecommendationEnv, which simulates user-product interactions. Within this environment, the states represent the users' current context, such as their browsing history or preferences, and actions correspond to the products recommended to the users. The transition dynamics are modeled such that recommending relevant products leads to positive rewards, while irrelevant recommendations yield lower rewards.

The DQN algorithm is implemented to learn an optimal policy for recommending products. The agent, represented by the DQNAgent class, learns to maximize cumulative rewards by iteratively interacting with the environment, selecting actions, observing rewards, and updating its Q-values. Experience replay and target networks are employed to stabilize training and improve convergence, while epsilon-greedy exploration balances exploration and exploitation.

The algorithm is trained over multiple episodes and epochs, all of which are recorded in “Weight and Biases”, with each episode consisting of interactions between the agent and the environment. During training, the agent learns to recommend products that align with users' preferences, aiming to maximize long-term rewards such as user engagement (here we consider click-through-rate) or purchases.

RL Techniques

1. DQN

   It utilizes deep neural networks (DNNs) to approximate the Q-function, which estimates the expected future rewards of taking a particular action in a given state.

2. Experience Replay

   The agent stores past experiences (state, action, reward, next state) in a replay memory buffer.

3. Epsilon-Greedy

   The agent employs an epsilon-greedy policy to balance exploration and exploitation.

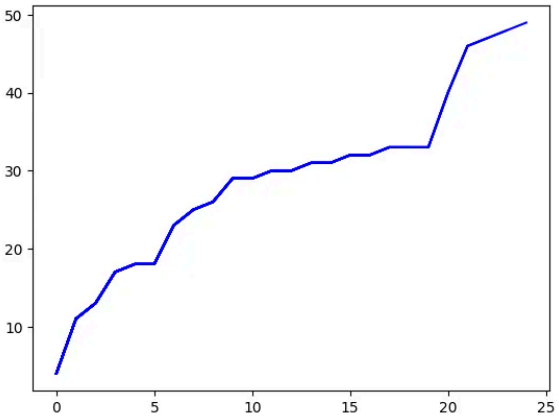
4. Reward Shaping

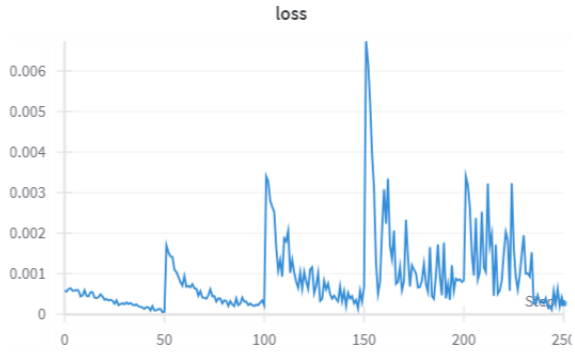
   The agent assigns rewards based on the user's interactions with the recommended products.

   Higher rewards are given for recommending products that are subsequently purchased by users or if the click-through-rate is high.

Experimental Results

1. In the first phase of the project, we have successfully proved that we have better optimized and personalized recommendations as our rewards have steadily increased with the episodes (Reward vs Episodes).





2. **Novelty score** was found to be 0.167 and Serendipity score = 0, owing to the dataset containing only fashion items.
3. Precision was found to be 1, since the agent was run several times and the agent solely relies on the past history of an user, thereby suggesting only highly likely items that will make it to the cart.

Future Work

1. Budget Constraints- Time based and Expense based
2. Comparison with Policy Gradient Environment