Reinforcement Learning based Recommendation System

(Practical Project)

Vaibhav Saraf, Rohit Thakur

28 October 2021

1 Introduction

The vast majority of traditional recommendation systems consider the recommendation procedure as a static process and make recommendations following a fixed strategy. A user interacts with recommendation engine in a sequence of exchanges of recommendations and provides feedback on them. Hence, we should also try to incorporate the feedback of the user at each time step while recommending items at the next time step. The idea is that previous interaction influence the later ones and the importance of the sequence of interactions can be modeled using Markov decision processes and solved by reinforcement learning.

2 Problem Statement

To formulate the recommendation system as Markov Decision Process, we consider the Recommendation System as Agent. The user interacts with the agent via environment, and obtains a reward and the new state. Formally, the definitions are:

- State Space: A list of N last interacted items before time t.

$$s_t = \{s_t^1, s_t^2, ...s_t^N\}$$

- Action Space: Action in the context of recommendation is the item to recommend at time t. Hence, the action space is equal to the total number of unique products available.
- Reward R: The recommendation agent executes action a_t and state s_t, and obtains the reward r_t in terms of product rating. Hence, the product rating can be binary or can be available rating options [1 to 5]. In our project, each item is represented by an embedding of product in vector space. The embedding is obtained by using Word2Vec like algorithm on list of product items.

3 Methodology

In the recommendation system, the number of actions is usually very high which is total number of unique products available. Usual Q-learning methods can estimate Q-value for each (s, a) pair, however, it becomes infeasible as the action space is very large (argmax operation over large action space can be costly and inefficient).
Hence, we decided to try Policy Gradient Method which are designed for continuous action space to optimize policy directly instead of focusing on value functions. We are focusing on Deep Deterministic Policy Gradient (Actor-Critic technique) which concurrently learns Q-function and a policy.
The input to this algorithm will be user's historic sessions of interacting with the system, and the expected output is the learned near-optimal policy which maximizes the cumulative reward (cumulative expected ratings).

4 Expected Outcome

Our expected outcome of the project is understanding and implementing the Deep Deterministic Policy Gradient in Recommendation systems, and understanding the effects of various parameters on the performance of the agent. In the end, our learned policy will take user's current history as an input, and provide K recommended items to the user based on the history. We also think that reinforcement learning methods can provide more diverse recommendation than classical supervised/unsupervised ML algorithms, and we will be trying to produce some quantifiable results on diversity of recommendations. This project provides us an opportunity to understand difficulties in implementation level for Reinforcement Learning algorithms in practical business applications.

5 Implementation Details

We will be learning about Policy gradient methods, specifically, DDPG algorithm in detail. The project will be implemented in Tensorflow. The dataset that we'll be using is Amazon Reviews Dataset. We will be sampling 100K ratings given by user and the metadata about the products(around 2000 products). We do not have the pre-implemented environment, hence we will be creating our own environment.

6 Estimated Week-by-Week Plan

– Week 1(29 Oct - 5 Nov): Understanding DDPG and Data Preprocessing

– Week 2(6 Nov - 13 Nov): Implementing the Environment (will be included in **Milestone Report**)

– Week 3(14 Nov - 21 Nov): Implementing DDPG Algorithm

– Week 4(21 Nov - 28 Nov): Iterative Qualitative and Quantitative Evaluation of the model based on different parameters

The overall tasks will be divided amongst the two of us. As both DDPG and Environment creation is new for both of us, we will be working together on most parts of the project. However, high level distribution of tasks is as follows:
Vaibhav - Data Cleaning and Creation of Embedding for environment creation (state and action), DDPG Implementation (Critic Part)
Rohit - Creating the environment and Implementing DDPG Algorithm (Actor Part), Evaluation of the trained policy

References

[1] M. Mehdi Afsar, Trafford Crump, and Behrouz Far
Link: Reinforcement Learning based Recommender Systems: A Survey

[2] Anton Dorozhko, Evgeniy Pavlovskiy
Link: Reinforcement learning for long-term reward optimization in recommender systems

[3] Xiangyu Zhao, Liang Zhang, Long Xia, Zhuoye Ding, Dawei Yin, Jiliang Tang
Link: Deep Reinforcement Learning for List-wise Recommendations

[4] Link: Deep Deterministic Policy Gradient Documentation

[5] Link: Amazon Reviews Dataset