고급 심층 강화학습 기법을 이용한 추천 시스템 구현

펭소니 [1], 싯소포호트 [1], 일홈존 [1], 김대영 [1,*], 박두순 [1,*]
[1]순천향대학교 소프트웨어융합학과

peng.sony61@gmail.com, parkds@sch.ac.kr, dyoung.kim@sch.ac.kr

Implementation of a Recommendation system using the advanced deep reinforcement learning method

Sony Peng[1], Sophort Siet[1], Sadriddinov Ilkhomjon[1], DaeYoung Kim[1,*], Doo-Soon Park[1,*]
[1]Dept. of Software Convergence, SoonChunHyang University

요 약

With the explosion of information, recommendation algorithms are becoming increasingly important in providing people with appropriate content, enhancing their online experience. In this paper, we propose a recommender system using advanced deep reinforcement learning(DRL) techniques. This method is more adaptive and integrative than traditional methods. We selected the MovieLens dataset and employed the precision metric to assess the effectiveness of our algorithm. The result of our implementation outperforms other baseline techniques, delivering better results for Top-N item recommendations.

1. Introduction

In the flood of information era, vast amounts of information are generated in milliseconds from myriad sources, leading to an information overload. With the explosion of information, recommendation algorithms are becoming increasingly important in providing people with appropriate content, improving their online experience.

Recommendation systems were developed to counteract this by saving users time and aiding businesses. They did this by providing a list of favorable items to targeted users or customers. From the customers' perspective, recommendation systems are beneficial as they reduce the time spent searching for items of interest. For the business industries, these systems increase revenue by showcasing products that customers are likely to purchase.

Recent years have seen considerable advancements in recommendation models. The transition has been from traditional recommendation mechanisms to models based on deep learning [1]. Traditional systems include techniques such as collaborative filtering (CF), content-based filtering (CBF), and a combination of both—referred to as the Hybrid method [2]. Despite their age, these mechanisms continue to draw academic interest, with efforts directed at enhancing their efficiency and addressing real-world challenges. While item retrieval from online sources might be straightforward, finding the most suitable items for users remains complex.

Deep learning is a model specializing in learning from complex information, making it a suitable option for current recommendation algorithms, particularly in the Big Data era. Like a black box, it processes input data through hidden layers to provide relevant outputs [3].

Reinforcement learning, another technique, has also garnered interest among scholars. It encompasses elements such as states, actions, environments, agents, and more. Deep learning and reinforcement learning have established themselves as influential subfields within artificial intelligence. Though their foundational principles and applications vary, they have shared factors contributing to their popularity and success.

Furthermore, both deep learning and reinforcement learning have played pivotal roles in the evolution of recommendation systems [4-5]. This study seeks to spotlight the potential of Deep Reinforcement Learning (DRL) within recommendation systems, contrasting it with traditional models and discussing challenges encountered during its application. Notably, our research does not exclusively focus on DRL; we also delve into classical recommendation mechanisms like collaborative and content-based filtering. We then present our findings using the Movielens 1 million rating dataset for evaluation [6].

* Corresponding author: Doo-Soon Park, DaeYoung Kim
Acknowledgments: This research was supported by the National Research Foundation of Korea (No. NRF-2022R1A2C1005921) and BK21 FOUR (Fostering Outstanding Universities for Research) (No.5199990914048)

2. Related work

Recommendation systems have received a lot of interest in academia. Several studies have been conducted to investigate the evolution of recommendation systems. The traditional recommendation is made up of two parts: collaborative filtering and content-based filtering. The most prevalent approach is collaborative filtering, based on implicit and explicit user-item interaction [7]. User profiles are created using content-based filtering based on user preferences and experience [8].

Ref. [9] designed movie recommendation system utilizing the benefit of social networking to mitigate cold-start problem in collaborative filtering. To handle data sparsity and cold-start, author in ref. Author in [10] provided the solution to combine user and items mechanism, namely hybrid recommendation system.

Meanwhile, significant study has been conducted on recommendation systems, ranging from conventional recommendation to deep learning, then deep learning to Reinforcement learning. The Model has been upgraded occasionally, and the troubles have also increased due to the dynamic nature of the data.

Deep learning emerged as a frontrunner in recommendation algorithms as a result of the Big Data revolution. [11] presents a thorough examination of how deep learning, with its ability to process large and complex datasets, became important in modern recommendation systems. Deep learning's black-box nature, as shown in this paper, demonstrates its capacity to produce meaningful content suggestions without explicit programming.

Parallel to deep learning, reinforcement learning makes an impression on the recommendation environment. Ref. [12] investigated the fundamental notions of states, actions, and agents in the field of reinforcement learning, their implications in recommendation systems.

The combination of deep learning and reinforcement learning opened up new opportunities for study into recommendation systems. Many researchers outlined the common characteristics that made both strategies powerful on their own and even more so when combined. Their examination highlighted the distinct strengths that each brought to the table, laying the groundwork for the investigation of Deep Reinforcement Learning (DRL).

While the aforementioned studies have significantly shaped the domain of recommendation systems, the field remains ripe for exploration. This study aims to further the understanding of DRL in recommendation systems, building upon the rich tapestry of research that precedes it.

3. DRL in Recommendation System

This section details a complex deep learning structure comprising user embedding, movie embeddings, and deep reinforcement learning. Below are the details of the proposed mechanism, as illustrated in Figure 1 in section 3.1 and their sub sections. More importantly, the details set up will be described in Section 4.

3.1 Recommendation Architecture

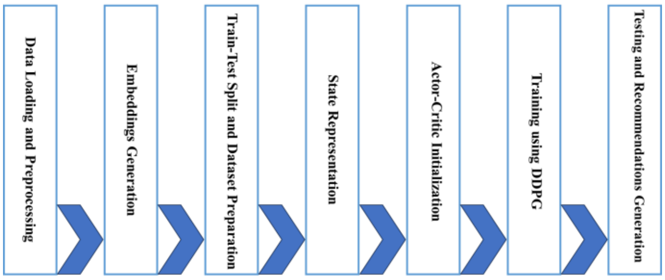

Figure 1 System Architecture

3.1.1 Embedding Layer:

User Embeddings: A dense embedding layer for the users that captures user preferences. The number of embeddings equals the number of users in the dataset. Movie Embeddings: A dense embedding layer for the movies. The number of embeddings equals the number of movies in the dataset.

3.1.2 State Representations:

s_drlavg: Direct concatenation of user embeddings with each movie's embeddings to produce a single dense vector representation.

s_drlu: Element-wise multiplication of the user embeddings with each movie's embeddings. Additionally, pairwise movie embeddings multiplication is included.

s_drlap: Only pairwise movie embeddings multiplication is considered to represent the state.

3.1.3 Deep Reinforcement Learning Component:

- Actor Network:
  - Input: State representation (from any of the methods mentioned above).
  - Output: Recommended action (a movie in this context).
  - Architecture: Multi-layered dense neural network with activation functions, and dropout layers for regularization.
- Critic Network:
  - Input: State representation and action (recommendation).
  - Output: Q-value, estimating the expected reward of taking an action given a state.
  - Architecture: Multi-layered dense neural network with activation functions, and dropout layers for regularization.

3.1.4 DDPG Component:

- Exploration and Exploitation:
  - A noise mechanism, such as Ornstein-Uhlenbeck noise, is added to the actor's output actions to promote exploration of the action space.
  - Replay Buffer: A data structure (usually a circular buffer) that stores experiences and

allows random sampling.
  - Target and Online Networks: Maintain two sets of networks (actor and critic) for both online and target. The target networks are softly updated using the online networks' weights.
- Training Process:
  - Train the Critic network using the loss between predicted Q-values and target Q-values derived from the Bellman equation.
  - Update the Actor network to maximize expected rewards by using the gradient ascent method.
  - Store experiences in the Replay Buffer and periodically sample to train both networks.
  - Employ soft updates to update the target networks.
  - Output: The Actor network will provide a recommendation for a movie based on the current state of user preferences and the relationships between movies.

4. Evaluation and Result

4.1 Experimental Setup

The current study utilizes several tools for supporting the building process, including Python version 3.9, Torch, Pandas, Numpy, and Matlibplot for visualization. Additionally, we create the entire Model with Ubuntu OS, with the capability RAM of 32Gb. Movielens 1M size is used for the evaluation. It consists of 1,000,209 anonymous ratings with the total of 3900 movies and the total of 6040 users who joined movielens in 2000. We select the users who positively rate movies greater than 3, and the rating counting is greater than 10, with the selection of an embedding size of 100 for movielens datasets. For initial parameters, dropout is set to 0.5, with a hidden size of 256, both Actor and Critic. Adam optimization is involved in deep learning with a learning rate of 0.0001. We use gamma (discount factor) equal to 0.9 to determine the importance of future rewards.

4.2 Evaluation

To evaluate the performance and to analyze the result. Movielens is selected for the recommendation system. The dataset comprises of 3 valuable files for recommendation systems and movie-related research. A common metric like precision is selected. Precision@k is used to evaluate recommendation systems. Precision@k measures how many of the top-k recommended items are relevant or true positive. Precision at k will be abbreviated as $P@k$ and form as below:

$$P@k = \frac{\#\ of\ relevant\ items\ among\ the\ top-k\ recommendations}{k}$$

4.3 Result Analysis

This part will display the result of recommendation for our Model and baselines. As shown in table 1. We select baseline methods such as Collaborative Filtering, Support Vector Machine, Non-Negative Matrix Factorization and Centered KNN. These baselines are from Surprise library using Python language with the help of other additional libraries including Pandas, NumPy, Torch, Tensorflow, and Matplotlib.

Table 1. Baseline Comparison

Models	P@5	P@10
Our Model*	**0.7537**	**0.7297**
CF	0.6033	0.6721
SVD	0.4934	0.4763
NMF	0.4500	0.3909
Centered KNN	0.4897	0.3801

Table 1 evaluates how many of the top k recommendations are relevant. A higher P@k value indicates a greater number of relevant recommendations in the top-k items, making the Model more effective in its suggestions. The table shows that "Our Model*" has the highest precision at both the top-5 and top-10 levels, indicating its better performance in delivering relevant recommendations compared to the other models. Furthermore, to highlight the advantage of "Our Model*" over the baseline models, we calculated the improvements by comparing them to the other baselines. As depicted in Figure 2, any negative values emphasize the baseline models' inferior performance compared to "Our Model*".

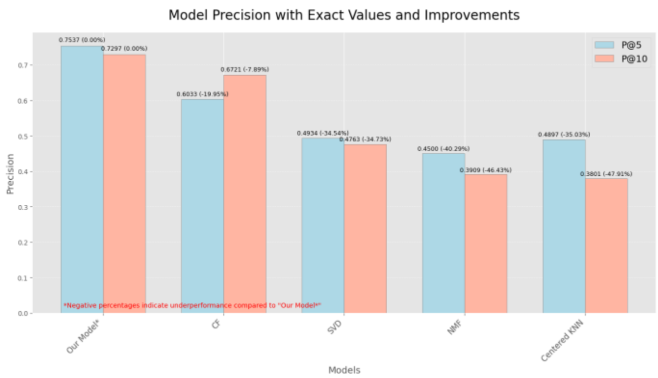

Figure 2 Model Performance

5. Conclusion

This study incorporated traditional and cutting-edge techniques to create a more effective recommendation system. To evaluate the model performance, we use the MovieLens dataset, comparing other baseline techniques such as CF, SVD, NMF, and Centered KNN. In the baseline implementation, we apply a recommendation library like Surprise. The baseline model uses the same dataset and keeps its default setting (parameters) available in the surprise library [13].

The results reveal that our novel approach, which utilizes Deep Reinforcement Learning (DRL), outperforms conventional techniques, particularly when recommending the best movie to target users. This indicates that advanced techniques such as DRL (deep reinforcement can significantly improve how we recommend items to users in the future. Following upward, we intend to investigate further combining additional emerging technologies with DRL to improve recommendation effectiveness. Furthermore, we will test our Model on various datasets to ensure its flexibility and robustness across domains.

참고문헌

[1] Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., ... & Nahavandi, S. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. Information fusion, 76, 243-297, 2021.

[2] Geetha, G., Safa, M., Fancy, C., & Saranya, D. A hybrid approach using collaborative filtering and content based filtering for recommender system. In Journal of Physics: Conference Series, 1000, 012101, 2018.

[3] Dong, S., Wang, P., & Abbas, K. A survey on deep learning and its applications. Computer Science Review, 40, 100379, . 2021.

[4] Zhang, S., Yao, L., Sun, A., & Tay, Y. Deep learning based recommender system: A survey and new perspectives. ACM computing surveys (CSUR), 52(1), 1-38, . 2019.

[5] Afsar, M. M., Crump, T., & Far, B. Reinforcement learning based recommender systems: A survey. ACM Computing Surveys, 55(7), 1-38, 2022.

[6] Harper, F. M., & Konstan, J. A. The movielens datasets: History and context. Acm transactions on interactive intelligent systems (tiis), 5(4), 1-19, 2015.

[7] Wang, Z., Xu, Q., Yang, Z., Cao, X., & Huang, Q. Implicit feedbacks are not always favorable: Iterative relabeled one-class collaborative filtering against noisy interactions. In Proceedings of the 29th ACM International Conference on Multimedia , pp. 3070-3078, 2021.

[8] Liao, M., & Sundar, S. S. When e-commerce personalization systems show and tell: Investigating the relative persuasive appeal of content-based versus collaborative filtering. Journal of Advertising, 51(2), 256-267, 2021.

[9] Xinchang, K., Vilakone, P., & Park, D. S. Movie recommendation algorithm using social network analysis to alleviate cold-start problem. Journal of Information Processing Systems, 15(3), 616-631, 2019.

[10] Sharma, S., Rana, V., & Malhotra, M. Automatic recommendation system based on hybrid filtering algorithm. Education and Information Technologies, 1-16, 2022.

[11] Fan, W., Zhao, Z., Li, J., Liu, Y., Mei, X., Wang, Y., ... & Li, Q. Recommender systems in the era of large language models (llms). arXiv preprint arXiv:2307.02046, 2023.

[12] Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. A brief survey of deep reinforcement learning. arXiv preprint arXiv:1708.05866, 2017.

[13] Hug, N. Surprise: A Python library for recommender systems. Journal of Open Source Software, 5(52), 2174, 2020.