영화 추천 시스템을 위한 연구: 한계점 및 해결 방법

초느에진랏[1], 마리즈아길랄[1], 무함마드 필다우스[2], 강성원[2], 이경현[3]
[1]부경대학교 정보보호학과
[2] 부경대학교 인공지능융합학과
[3]부경대학교 컴퓨터공학부
(chocho1612, marizaguilar, mfirdaus) @pukyong.ac.kr, jsm2371@hanmail.net, khrhee@pknu.ac.kr

Survey for Movie Recommendation System: Challenge and Problem Solution

Cho Nwe Zin Latt[1], Mariz Aguilar[1], Muhammad Firdaus[2], Sung-Won Kang[2], Kyung-Hyune Rhee[3]
[1]Dept. of Information Security, Pukyong National University
[2]Dept. of Artificial Intelligence Convergence, Pukyong National University
[3]Division of Computer Engineering, Pukyong National University

Abstract

Recommendation systems are a prominent approach for users to make informed automated judgments. In terms of movie recommendation systems, there are two methods used; Collaborative filtering, which is based on user similarities; and Content-based filtering which takes into account specific user's activity. However, there are still issues with these two existing methods, and to address those, a combination of collaborative and content-based filtering is employed to produce a more effective system. In addition, various similarity methodologies are used to identify parallels among users. This paper focuses on a survey of the various tactics and methods to find solutions based on the problems of the current recommendation system.

Keywords: Movie recommendation system, Content based filtering, Collaborative filtering, Matrix factorization

1. Introduction

A recommendation system is a system that is widely used in many areas such as e-commerce. For example, if a customer goes to a store and buys a plate, then the seller asks, "Mr./Ms., did you not buy tea? The tea is very good today. It has been brewed." They could easily provide advice and guidance on the customer's purchases. But, if the customer buys an item online, how can they provide effective service in offering the right product? If we can present products based on the customer's preference, online shopping sales will surely grow significantly. Therefore, we proposed a system that will equip recommendations. This paper focuses on the movie recommendation system, so our main purpose is to give the most relevant movies suitable for the user. Here, we used *matrix factorization* and *candidate generation techniques* to achieve our expected result.

This paper is divided as follows; Section 2 will discuss the related work in the recommendation system. Section 3 and 4 will explain the challenges and problems in the current movie recommendation system. Section 5 will present the solution. And finally, the concluding remark in Section 6.

2. Related Work

The term recommendation system covers a broad spectrum of topics that are applied in various fields. It is employed in a wide range of real-world applications, including entertainment, e-commerce, services, and social media [1]. In the entertainment industry, it is extensively utilized when watching movies or listening to music. The following related works are some of the other applications that use a recommendation system.

- Movie Recommendation: *Netflix* uses algorithms for recommending movies according to their interest.
- Music Recommendation: *Pandora* uses the properties of a song or the artist.
- News Recommendation: Various applications that provide news recommendation can be Google News or Apple news.

3. Challenges in the current system

The recommendation system faces numerous challenges, such as cold start problem, data sparsity, and scalability.

- *A cold start problem* is when the system is not able to recommend items to the users. As for every recommender system, it is required to build a user profile by considering users' preferences. So, the user profile is developed based on the activities and behaviors they perform in the system [2].
- *Data sparsity* is the term used to describe the phenomenon of not observing enough data in a dataset. Also, difficulty of finding enough reliable similar users, since in general, active users rated only a small portion of the items.
- *Scalability* is where the amount of data used as input in the recommendation system is growing quickly as more users and items are added [2].

3.1. Technical Review

There are two techniques mostly used in recommendation systems; Collaborative filtering and Content-based filtering.

3.1.1. Content-based Filtering

Content-based filtering is a type of system that attempts to guess what a user may like based on its activity. It uses attributes such as genre, director, description, actors, etc. to make suggestions [3]. In figure 1, the idea of content-based filtering is to recommend an item based on a comparison between the content of the items and the user profile. For example, a user might receive a movie recommendation according to the description of other movies. Therefore, it is easier for a content-based approach to recommend new items.

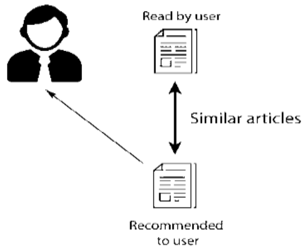

Figure 1: Content-based Filtering

Also, it provides a content feature that helps explain why the recommendation is being made. However, finding a particular feature like images or movies of any genre can be a problem. In general, this is called an over-specialization problem. The users have never recommended outside the user profile, so it is easy to miss recommending items since there is not enough information about it [4].

3.1.2. Collaborative Filtering

Collaborative filtering finds users whose likings are similar to another user. Then, it recommends an item or product, assuming that the other user would also consider the suggestion with a comparison with another user who also shares the same preferences. It means this method matches users with the same interest and gives recommendations based on their likes [5].

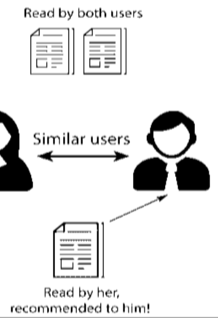

Figure 2: Collaborative Filtering

The theory behind collaborative filtering is to work in collaboration with the user or movie id. For example, in figure 2, there are two users Alice and John. Alice like movies A, B, C, and D, while John likes movies C and D. Since movies C and D are common to Alice and John, movies A and

B will be recommended to Alice [5].

4. Problems in the current Recommendation System

Many open-source recommendation systems are available on the internet. Most of them have their own functions such as Candidate Generation, where the current customer selects the product that the customer will like out of the tens of thousands of products. Another is the selection of tens of items from hundreds of products and advising the customer, this is called the Ranking.

Basically, recommendation systems differ based on the functions used. In this paper, we emphasize a movie recommendation system that is designed to support both of the features mentioned above to address problems such as; (1) information overload for the users due to number of choices sets with irrelevant videos that' s affecting the quality of the video recommendations. (2) And user-item dataset that is so sparse that the recommendation system cannot properly learn the user-item relationships.

5. Solution to the problems in the current Recommendation System

We can see the Candidate Generation feature utilized in the proposed system. So, to build a candidate generation model we implemented a Neural network-based matrix factorization type. The main function of the matrix factorization type is to model which function to predict, so input data is important when performing this function.

5.1. Candidate Generation

Figure 3 explained the general structure of our recommendation system. Two neural networks are used in the system: one for *candidate generation* and the other for *ranking*. The candidate generation network uses events from the user' s movie history to get a tiny subset (hundreds) of videos from a vast corpus. These candidates are meant to be highly precise while being broadly relevant to the user. Only collaborative filtering allows the candidate generation network to deliver extensive customization. The main input data used for movies is the movie ID, search query, and demographic information [8]. To identify relative importance among candidates with strong recall, a fine-level representation is required when presenting a few "best" recommendations in a list.

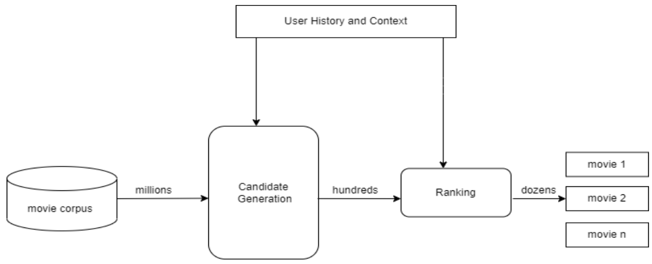

Figure 3: Recommendation System Architecture

Meanwhile, the ranking network achieves this by giving each video a score based on a specified objective function and a large number of characteristics that describe the video and the user. The user is shown the highest-scoring movies, which are ranked by their score. The two-stage approach to recommendation allows us to make recommendations from a big corpus of movies (millions) while trusting that the limited number of movies that appear on the device is personalized and entertaining to the user.

5.2. Matrix Factorization

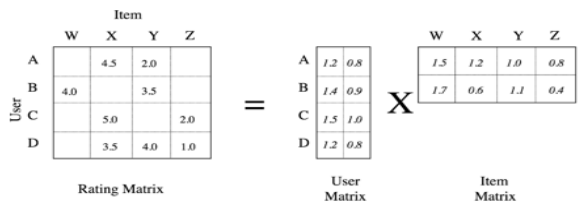

Figure 4: Matrix Factorization

Figure 4 explains that the most important output from matrix factorization is the scoring of items for the user, as well as learned user and item embedding. Additionally, learn embedding can be used for other transfer learning, also in a re-ranking section [8].

As a result of our training using Candidate Generation, we found that the strategies used in the offline evaluation were different from those used in online serving. According to the paper, the offline evaluation focuses on precision training and model training. While, it is said that A/B testing is used to model online evaluation. Although, offline and online results were not directly affected [9].

5.3. SoftMax Formula

We present the recommendation as extreme multiclass classification to address the accuracy in prediction problem wherein it classifies a video watch w_t at time t among millions of videos i from examples V based on a user U and a context C, as expressed below [10].

$$P(w_t = i|U,C) = \frac{e^{v_i u}}{\sum_{j\in V} e^{v_j u}}$$

In the above equation, embedding is simply mapping sparse entities into a dense vector. So basically, the deep neural network's task is to learn user embedding u as a function of the user's history and context that can be used to categorize between videos using a SoftMax classifier. Additionally, we use the implicit feedback mechanism to train the model that is based on watches instead of the explicit mechanism which is already been used in a streaming platform such as YouTube. The reason is, it is available and can generate recommendations in situations where explicit feedback is very rare.

When we calculate the scoring, we use the SoftMax formula [10]. SoftMax is a formula for calculating the probability of the relevant class for multi-classification; adding the probability of all these classes adds up to 1. Therefore, the probability of this formula can be used as scoring. The result of this Candidate Generation is user, content embedding, and movie embedding, so if we can input the user and content pair into the model, we will get the relevant movie and scoring. We can recommend relevant movies to the user without overwhelming them with information. As a result, we can solve the above two problems utilizing the three techniques as we expected.

6. Conclusion

Movie recommendation systems have shown to be the most effective means of tackling the problem of information overload. They assist in decision-making by conserving time and energy. Future work will focus on improving the existing methods and testing used in the recommendation systems to increase prediction, scoring, and recommendation quality.

Acknowledgement

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation) (IITP-2022-2020-0-01797) and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education(2021R1I1A3046590)

References

[1] G. Suganeshwari, and S. P. Syed Ibrahim. "A survey on collaborative filtering-based recommendation system." *Proceedings of the 3rd international symposium on big data and cloud computing challenges (ISBCC–16')*. Springer, Cham, 2016.

[2] L. Sharma, and Anju Gera. "A survey of recommendation system: Research challenges." *International Journal of Engineering Trends and Technology (IJETT)* 4.5 (2013): 1989-1992.

[3] S. Reddy, S. Nalluri, S. Kunisetti, S. Ashok, and B. Venkatesh, "Content-based movie recommendation system using genre correlation," *Smart Innov. Syst. Technol.*, vol. 105, pp. 391–397, 2019, doi: 10.1007/978-981-13-1927-3_42.

[4] P. Lops, M. de Gemmis, and G. Semeraro, "Content-based Recommender Systems: State of the Art and Trends," *Recomm. Syst. Handb.*, pp. 73–105, 2011, doi: 10.1007/978-0-387-85820-3_3.

[5] J. Ben Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender systems," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 4321 LNCS, pp. 291–324, 2007, doi: 10.1007/978-3-540-72079-9_9.

[6] P. Kumar, and Ramjeevan Singh Thakur. "Recommendation system techniques and related issues: a survey." *International Journal of Information Technology* 10.4 (2018): 495-501.

[7] Arora, et al. "Movie recommendation system based on users' similarity." *International journal of computer science and mobile computing* 3.4 (2014): 765-770.

[8] Y. Koren, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems." *Computer* 42.8 (2009): 30-37.

[9] R. Mehta, and Keyur Rana. "A review on matrix factorization techniques in recommender systems." *2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA)*. IEEE, 2017.

[10] H. Lee, and Jungwoo Lee. "Scalable deep learning-based recommendation systems." *ICT Express* 5.2 (2019): 84-88.