Preliminary Introduction

The project aims to build a hybrid recommendation model for Netflix. Recommendation systems are ubiquitous in daily lives and a primary revenue-generating instrument for digital marketing niches. The Netflix competitions in the past two decades, which awarded each winner one million US dollars, exemplified the demand for sophisticated recommendation and its evident benefits for business development and user experience utilization. A hybrid model is built upon two dominantly popular methods in recommendation, collaborative filtering and content-based approaches. The preliminary literature review below examines the two methods, with the goal of incorporating the advantages of both methods in a hybrid system.

Literature Review

Collaborative filtering(CF) is a technique in recommendation system in which the recommendations are dependent on feedback or past behaviors from each user, including explicit ones (e.g., the user's rating) and implicit ones (e.g., the user clicks on a link, listens to a song, or purchases an item). However, in collaborative filtering, explicit ones can reflect users' performance more accurately than implicit ones. Therefore, we focus more on explicit feedback.

Article [1] suggests that CF can be further categorized into two classes: neighborhood-based and model-based approaches. Neighborhood-based algorithms predict ratings for the user-item pairs based on their neighborhoods, while model-based algorithms build a model to learn the user-item interactions with their latent characteristics to predict.

The neighborhood-based algorithms are the traditional ways with the advantage of simplicity and efficiency for providing recommendations. Basically, there are two types of neighborhood-based approaches: User-based Collaborative Filtering (UBCF) and Item-based Collaborative Filtering (IBCF). For the User-based Collaborative Filtering (UBCF), the main idea is to identify similar users who displayed similar ratings on different items. Then, the ratings provided by these similar users are used to provide recommendations. For the Item-based Collaborative Filtering, the main idea is to identify similar items which have similar ratings among different users. Then the ratings received by these similar items from the user are utilized for recommendations.

When implementing the neighborhood-based algorithms, we need to consider some factors which could have significant impact on the recommendation process, including rating normalization method, computation of similarity values and neighborhood selection. For the real-world application, sparsity of the user-item matrix is a challenge which makes the process of finding a neighborhood very difficult. One solution is to define the neighborhood by utilizing the demographic information of users [2]. Another way is to replace missing values with default values such as a user's average rating [1, 3]. Another challenge is that the distribution among items is highly skewed, i.e., only a small proportion of items are popularly rated. It is hard to provide recommendations in the items rarely rated [4,5]. In addition, when new users or new items are added to the system, it's hard to provide recommendations by traditional collaborative filtering methods. Hybrid methods which leverage information of users and items can deal with this problem [6,7]

For model-based CF, one of the most popular algorithms is matrix factorization, the basic approach behind which is singular value decomposition. There are lots of sophisticated algorithms beyond that, including SVD and its variants. Since the traditional SVD can only be applied to dense matrices with no missing value, the underlying idea of its variant is to determine the user performance by linearly combination of users and items vectors mapped into a low-dimensional latent space. In 2008, Salakhutdinov and Mnih [8] presented a Probabilistic Matrix Factorization(PMF) model, which scales linearly with the number of observations and can perform well on the large, sparse, and very imbalanced Netflix dataset. Another variant that achieved great success in Netflix competition is Funk-SVD [9], it can help make recommendations with a very sparse matrix, by computing latent factors only using the factors we know. Based on Funk-SVD, adding bias for users and items, BiasSVD [10] is also a widely used model in recommendation systems. Moreover, by including users' implicit feedback, SVD++ [11] , an enhanced model of BiasSVD, was proposed. It is also one of the most effective models in the recommendation system. There are also researches combining collaborative filtering with neural networks. A two-layer undirected graphical model, Restricted Boltzmann Machines(RBF) [12] , is a mainstream algorithm for recommendation systems and successfully applied to Netflix data sets.

Content-based filtering focuses on the source item content and makes recommendations based on similarity counts. In particular, content-based methods directly examine tags of source items rated by the user and features of items to be recommended. Many such approaches can be categorized as classification of the items based on user preference. Content-based filtering is adequate for constructing independent user profiles and analyzing items not yet rated by any user. Yet the requirement for a large dataset often imposes challenges for finding a good representation of items and a reliable classifier.

Most studies represent each individual item with candidate tags directly extracted from the content source. In the Netflix case, for example, this phase can be accomplished by directly analyzing the HTML source of Netflix and other public databases. Popular classification algorithms, such as the Winnow algorithm which works particularly well on text classification, are used to assign weights to the candidate tags and identify the important ones.

[1] Breese, John S., David Heckerman, and Carl Kadie. "Empirical analysis of predictive algorithms for collaborative filtering." *arXiv preprint arXiv:1301.7363* (2013).
[2] Pazzani, Michael J. "A framework for collaborative, content-based and demographic filtering." *Artificial intelligence review* 13.5 (1999): 393-408.
[3] Deshpande, Mukund, and George Karypis. "Item-based top-n recommendation algorithms." *ACM Transactions on Information Systems (TOIS)* 22.1 (2004): 143-177.
[4] Park, Yoon-Joo, and Alexander Tuzhilin. "The long tail of recommender systems and how to leverage it." *Proceedings of the 2008 ACM conference on Recommender systems*. 2008.
[5] Yin, Hongzhi, et al. "Challenging the long tail recommendation." *arXiv preprint arXiv:1205.6700* (2012).
[6] Park, Seung-Taek, and Wei Chu. "Pairwise preference regression for cold-start recommendation." *Proceedings of the third ACM conference on Recommender systems*. 2009.
[7] Lam, Xuan Nhat, et al. "Addressing cold-start problem in recommendation systems." *Proceedings of the 2nd international conference on Ubiquitous information management and communication*. 2008.
[8]Mnih, Andriy, and Russ R. Salakhutdinov. "Probabilistic matrix factorization." *Advances in neural information processing systems* 20 (2007).
[9] Friedman, Jerome H. "Greedy function approximation: a gradient boosting machine." *Annals of statistics* (2001): 1189-1232.
[10]Koren, Yehuda, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems." *Computer* 42.8 (2009): 30-37.
[11]]Koren, Yehuda. "Factorization meets the neighborhood: a multifaceted collaborative filtering model." *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2008
[12]Salakhutdinov, Ruslan, Andriy Mnih, and Geoffrey Hinton. “Restricted Boltzmann machines for collaborative filtering." *Proceedings of the 24th international conference on Machine learning*. 2007.