A Literature Review of Collaborative Filtering Recommendation System using Matrix Factorization algorithms

Winnie Nguyen
Computer Science Department at Earlham College
Richmond, Indiana, USA
zdnguyen18@earlham.edu

ACM Reference Format:
Winnie Nguyen. 2021. A Literature Review of Collaborative Filtering Recommendation System using Matrix Factorization algorithms. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In the technology 4.0 era, with an increasing number of online shopping platforms opened, increasing user interaction and enrich shopping potential to earn revenue by utilizing recommender systems has been many eCommerce companies' ultimate goals. Recommender systems (or recommendation engines) exploit similarities between the users and their item's choices to make recommendations, allowing companies to maximize their return on investment (ROI) based on information gathered from customers through their experiences, behaviors, preferences, and interests [3]. While recommendations influence 80% of content watched on Netflix and help the company save 1 billion dollars yearly in value from customer retention, Amazon's revenue witnessed a 21.11% year-over-year growth, reaching more than $250B by the end of 2019 thanks to item-to-item collaborative filtering recommendation engines [6]. This literature review will cover a diverse set of papers on how factorization algorithms can be used in collaborative filtering recommender systems for predicting users' preferences. First, I will explain the characteristics of a data source used in collaborative filtering approaches and make details of datasets used in these research papers. Second, I will explain what matrix algorithms have been suggested by most researchers in the field. This section is divided into two parts. The first part surveys studies that developed and implemented the Singular Value Decomposition (SVD) algorithm to make recommendations. The second part explores studies that used the Alternative Least Square (ALS) algorithm to do the same. Finally, I will conclude by suggesting possible future work in the field.

2 DATASETS

This section introduces the datasets used by researchers in collaborative filtering recommender systems. While different researchers have different ways of approaching the problem and make different assumptions, they have the same objective to enhance the performance of recommender systems in terms of accuracy or running time. It is essential to understand the characteristics of data and why researchers choose specific datasets in those papers.

In general, collaborative filtering technique is based solely on the past interactions recorded between users and items, stored in an user-item interactions matrix [7]. Users' interactions have two main types: while explicit interactions include past activity, ratings, reviews and information about users' profile, implicit ones contain devices users use for access, click on a link, location, and dates [3]. For collaborative methods, the explicit feedback can reflect users' preferences more accurately than implicit ones as there is less noise in the data. In datasets used in these papers, explicit users' preference information is shown in users' rating column.

MovieLens [4] is the most common dataset among these research papers for collaborative filtering personalized recommendations - five out of six articles use MovieLens as their primary dataset. MovieLens dataset is provided by MovieLens group, a web-based research recommender system with over 20 million movie ratings and tagging activities since 1995 [4]. As all randomly selected users rated at least 20 movies, researchers can assume users care more about the product and not be less active. The dataset can be converted into a user-item matrix with about 100,000 ratings, and all unrated items have a zero value. The rating scale ranges from 1 to 5, where 1 represents dislike, and 5 illustrates a strong preference. Moreover, no demographic data is included; an id represents each user or each movie.

Besides MovieLens, other papers use different dataset: Paterek [5] used the Netflix Prize data, which contained a training.txt file with more than 100M ratings on a scale of 1 to 5 about 17,770 movies made by 480,189 customers; Aljunid and Manjaiah [1] used the Bookcrossing dataset consisting of 1.1M rating scale from 1 to 10 of 270,000 books by 90,000 users, and Zhou et al. [10] used Flixster dataset containing more than 8M ratings from 786,936 users for 48.794 movies in Flixster commercial website.

3 METHODS FOR COLLABORATIVE FILTERING RECOMMENDER SYSTEMS

This section describes the two main matrix factorization algorithms and their updated techniques, including Singular Value Decomposition (SVD) and Alternative Least Square (ALS) to perform recommender systems better. Those two methods solve three main problems suffered by using collaborative filtering approaches [4]:

- `Cold start`: refers to a problem, when for a new user or item there is not enough data to make recommendations
- `Scalability`:a large amount of computation power is required to make recommendations.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

- `Sparsity`: the missing values in users-items matrix because many users will only have rated a small subset of the overall database. Thus, even the most popular items have very few ratings.

The accuracy of the recommender system is evaluated through two popular measures: Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). The lower value of both metrics, the better performance of the recommendation algorithms.

3.1 Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) is a matrix factorization algorithm that can extract characteristics of the dataset's features by splitting the original user-item ratings matrix into three smaller matrix multiplications [5]. Therefore, SVD decreases the dimension of the utility matrix. Paterek [5] evaluated six predictors based on SVD, including regularized SVD of data with missing values, improved regularized SVD, K-means, post-processing results of regularized SVD with KNN, regularized SVD with biases, and post-processing result of SVD with kernel ridge regression, using the Netflix Prize dataset. Each predictor has a slight update compared with regularized SVD, but not all predictors can improve the performance:

- Improved regularized SVD adds biases parameters to the predictions formula for user i and movie j;
- Post-processing SVD with KNN uses prediction by one nearest neighbor using similarity formula between two items;
- Post-processing SVD with kernel ridge regression tries to get rid of all weights after training.

Decreasing the number of parameters in the SVD-based model, Paterek designed and implemented two new models whose parameters used gradient descent regularization and early stopping. Besides evaluation, the researcher made a hybrid model merging in proportion 85/15 two linear regression using different training-test sets, experimenting with 7.04% improvement in the original performance.

Zhou et al. [10] proposed an incremental algorithm called Incremental ApproSVD - the combination of Incremental SVD and Approximating SVD algorithm - to improve running time and accuracy of predicting new items entered dynamically. Compared to other clustering or data dimensionality reduction methods which solve the massive amount of data with quick response time and the sparsity problem as offline computation, Incremental ApproSVD can handle online and dynamic issues more efficiently. For the dataset, they used MovieLens and Flixster. The essential technique of Incremental ApproSVD is choosing column sampling probabilities, specifically adopting column sampling to reduce the column number then the size of the original matrix. The evaluation showed that the prediction model could predict unknown ratings when new items enter dynamically with the lower value of both RMSE and MAE, plus be a suboptimal approximation with less running time. Moreover, the paper provided an updated mathematical error analysis between the actual ratings and the predicted ones generated by the Incremental ApproSVD algorithm.

3.2 Alternative Least Square (ALS)

Alternative Least Square (ALS) is a matrix factorization algorithm, factoring the original user- item ratings matrix into the user-to-feature matrix and the item-to-feature matrix. ALS works well with sparse matrices by filling random numbers to the blank spaces in each matrix and calculating the error term [4]. Doing the filling progress back and forth until multiplying the two matrices again, researchers can fill the sparse in the original user-item matrix [2]. In these papers, ALS-based models are applied mainly in Spark ecosystem - an open-source parallel computing framework or Apache Spark, which is one of the cluster computing systems in this ecosystem.

Xie et al. [9] constructed a parallel implementation process based on the ALS and implemented a new loss function for this algorithm. The problem of ALS loss function is the difference between the similarity of users-items before and after training as losing some information while processing the user-to-feature and item-to feature matrices. The improved loss function in the paper provides the same similarities between users and items before and after the model is trained. In this paper, researchers created a Spark cluster as the experimental environment and randomly chose 10,000 out of 1M data items in MovieLens as experimental data. As a result, they succeeded in improving the accuracy performance of the proposed model after five iterations, decreasing the average RMSE by 3.7%.

Aljunid and Manjaiah [1] proposed an improved recommendation model based on ALS, applying a new function called kfoldALS, evaluating in two datasets - MovieLens and Bookcrossing in Apache Spark. In ALS's improved model, the authors implemented the kfoldALS function to split the dataset into k datasets at the beginning step, then iterate the training and evaluating process several times. The advantage of using an improved model is preventing the randomness of the random split strategy and the function to build an original ALS model to get a more precise performance score by using the entire dataset. Researchers evaluated the improvement of accuracy by RMSE metric.

Awan et al. [2] focus on improving the performance of recommendation based on the ALS by tuning selected parameters and sizing down the number of ratings. First, the authors used 100,000 ratings out of 20M data points in MovieLens dataset with the original ALS algorithm implemented in data bricks - a cloud-based platform launching optimized clusters for Spark ecosystem and evaluating by RMSE. By optimizing selected parameters, including using six latent factors, regularization = 0.2 with iteration =5, and producing predictions up to 1000 movies, the authors managed to not only obtain 97% accuracy with lower RMSE but improve runtime as well.

Winlaw et al. [8] conducted an approach of combining alternating least squares (ALS) and nonlinear conjugate gradient (NCG) to optimize the collaborative filtering recommender system. The disadvantage of using NCG separately in practice is slowly converging to an optimization problem. Therefore, instead of using NCG as an alternative for the ALS algorithm, researchers combined and used ALS as a nonlinear preconditioner for NCG. Implementing parallel ALS-NCG with the MovieLens dataset in Apache Spark, researchers managed to apply different techniques to improve the accuracy of prediction, speed up the running time, expand the number of data points being used - up to nearly 1 billion ratings, and be able to apply in any collaborative filtering model based on ALS algorithm.

4 CONCLUSION

The literature review discussed how collaborative filtering models are used for personalized recommenders in eCommerce. First, I explained the typical qualifications for datasets and mentioned specific datasets used in research papers. Second, I explained the matrix factorization algorithms researchers used for predicting users' preferences. One approach focuses on using algorithms based on SVD, and the other is techniques using ALS-based algorithms. Each paper uses improved techniques and strategies with the primary dataset MovieLens and additional datasets such as Bookcrossing, Flixster, and the Netflix Prize. Therefore, it is challenging to conclude which approach is the best to improve the performance of collaborative filtering recommender systems. In the future, applying different methods in the same dataset, using the same data split strategy and performance metrics will help evaluate the efficiency of different algorithms.

REFERENCES

[1] Mohammed Fadhel Aljunid and DH Manjaiah. 2018. An improved ALS recommendation model based on apache spark. In *International Conference on Soft Computing Systems*. Springer, 302–311.

[2] Mazhar Javed Awan, Rafia Asad Khan, Haitham Nobanee, Awais Yasin, Syed Muhammad Anwar, Usman Naseem, and Vishwa Pratap Singh. 2021. A Recommendation Engine for Predicting Movie Ratings Using a Big Data Approach. *Electronics* 10, 10 (2021), 1215.

[3] Roger Chua. 2019. A simple way to explain the Recommendation Engine in AI. Retrieved Sep 2, 2021 from https://medium.com/voice-tech-podcast/a-simple-way-to-explain-the-recommendation-engine-in-ai-d1a609f59d97

[4] GroupLens Kaggle. 2018. Introduction to recommender systems. Retrieved Sep 2, 2021 from https://www.kaggle.com/grouplens/movielens-20m-dataset/activity

[5] Arkadiusz Paterek. 2007. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD cup and workshop*, Vol. 2007. 5–8.

[6] Kaja Polachowska. 2019. Is It Worth It? ROI of Recommender Systems. Retrieved Sep 2, 2021 from https://dzone.com/articles/is-it-worth-it-roi-of-recommender-systems

[7] Baptiste Rocca. 2019. Introduction to recommender systems. Retrieved Sep 2, 2021 from https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada

[8] Manda Winlaw, Michael B Hynes, Anthony Caterini, and Hans De Sterck. 2015. Algorithmic acceleration of parallel ALS for collaborative filtering: Speeding up distributed big data recommendation in spark. In *2015 IEEE 21st International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 682–691.

[9] Li Xie, Wenbo Zhou, and Yaosen Li. 2016. Application of improved recommendation system based on spark platform in big data analysis. *Cybernetics and Information Technologies* 16, 6 (2016), 245–255.

[10] Xun Zhou, Jing He, Guangyan Huang, and Yanchun Zhang. 2015. SVD-based incremental approaches for recommender systems. *J. Comput. System Sci.* 81, 4 (2015), 717–733.

[5] [9] [1] [2] [10] [8] [3] [6] [7] [4]