Movie Recommendation System
Tuned Asymmetric Singular Value Decomposition

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

ABSTRACT

In recent years, the need for more accurate recommender systems to improve user interaction and provide more personalized services on eCommerce platforms such as Amazon and Netflix has increased globally. The motivation results from a desire to help users find an appropriate product that fits their tastes and meets various special needs, enhancing users' satisfaction and loyalty. However, with the overload of vast amounts of customer data, recommender systems face challenges in processing data robustly and accurately. This proposal focuses on designing a movie recommendation system that takes into account both explicit and implicit ratings and performs well when new users are added to the original dataset. The base algorithm in my paper is Singular Value Decomposition (SVD), an applied matrix factorization method of the item-based collaborative filtering model. To solve scalability matters, reduce the expensive matrix factorization steps, and integrate implicit feedback in the model, the tuning of Asymmetric SVD is expected to improve prediction accuracy.

ACM Reference Format:

Winnie Nguyen. 2022. Movie Recommendation System Tuned Asymmetric Singular Value Decomposition. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In the technology 4.0 era, especially after the COVID-19 pandemic triggered stay-at-home orders and canceled all social plans, people spend most of their time on online media streaming services, leading to the rapid growth of streaming platforms [28]. According to Statista, about 62% of US adults currently subscribe to at least one streaming service [21]. Moreover, a market research report by Grand View Research states that the global streaming market was worth $42.6 billion in 2019, expected to grow more than 20% annually until reaching around $185 billion by 2027 [21]. More details about this vast and growing industry show that YouTube has 2 billion monthly active users for free services and 186 million subscribers [4]. At the same time, Netflix hits over 220 million subscribers globally in 2021 for just its streaming service [3]. Netflix has become the world's leading Internet television network and the most valued and largest streaming service [29], which motivates them to personalize users' experiences more correctly with the help of machine learning and data science. As Ted Sarandos, Netflix's chief content officer, said: "There's no such thing as a 'Netflix show.' Our brand is personalization" [12]. The mission of the recommendation system at Netflix or any online service is to suggest to users the most relevant product they love.

Along with the explosion of the streaming and eCommerce industries, various recommendation methods have been proposed to help consumers discover the product to buy or the movie that fits their tastes. Moreover, an accurate recommendation system allows online providers to maximize their return on investment (ROI) based on information gathered from users through their experiences, behaviors, preferences, and interests [7]. Besides providing a better user experience and boosting subscription rates, a recommendation system is a tool to help streaming and eCommerce companies enhance customer engagement and increase traffic to their website [24].

Researchers have developed various recommender systems intending to help online providers utilize a large amount of customer data and ease the decision-making process for users. Three most commonly applied recommendation models are collaborative, content-based, and hybrid approaches. Collaborative filtering uses the concept that "a set of users possessing similar features" will have similar interests to make recommendations. It applies matrix factorization methods to find the relations between items and users. Meanwhile, content-based filtering recommends depending on the similarity measurement between item-feature and target-feature, rather than on the user's opinions [5]. The hybrid approach combines two or more techniques to maximize the benefits while covering all chosen methods' weaknesses.

Among top online providers, Spotify, Amazon, and Netflix, the collaborative filtering model is widely used to help users navigate extensive product assortments, make decisions and overcome information overload. Spotify platform provides relevant music suggestions by combining three algorithms in different systems, including collaborative filtering, the natural language process model, and the audio path analysis model. Their recommendation system is named "Bandits for Recommendations as Treatments," or "BRaT" [16]. Meanwhile, Amazon, the world's leading online retailer, launched its item-based collaborative filtering in 1998, applied to millions of customers and millions of products [27]. Thanks to item-to-item collaborative filtering recommendation engines, Amazon's revenue witnessed a 21.11% year-over-year growth, reaching more than $250B by the end of 2019 [27]. For two decades, Amazon's

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

growth has served as confirmation of the advantage of its collaborative filtering recommendation systems: simplicity, scalability, explainability, adaptability, and relatively high-quality algorithms [27]. Netflix's successful recommendation system is combined with four complex algorithms that utilize the customer viewing data, search history, rating data, or time duration to generate recommendations [14]. In 2016, Netflix began its contest to find a better movie recommendation system to replace its current design. The winner used collaborative filtering models, specifically various versions of SVD algorithms, to upgrade the accuracy by at least 10% [10]. Nowadays, Netflix Recommendation Engine (NRE) influences 80% of content watched on Netflix and helps the company save one billion dollars yearly in value from customer retention [20].

This proposal presents a collaborative filtering recommendation model based on matrix factorization algorithms called Singular Value Decomposition (SVD). With the tremendous growth of users and products, collaborative filtering methods face two challenges: accuracy and scalability. While the first challenge is to improve the quality of recommendations to show customers their personalized preferences, the second one is that we also want to upgrade the scalability of the algorithms to handle millions of potential neighbors in real-time. Those two challenges conflict in some ways. The less time an algorithm uses to search for neighbors, the more scalable it will be, but the worse its quality will be [23]. Therefore, besides the SVD-based approach, the paper will implement a tuned Asymmetric SVD model-building to enhance the scalability and integrate implicit as well as explicit interactions while providing better predictive accuracy for the movie recommendation dataset.

This paper introduces collaborative filtering recommender systems and the challenges in applying this method. Then the report focuses on related work about different SVD-based algorithm approaches. Following this is the design of the proposed systems - tuned Asymmetric SVD, explained in detail. This section elaborates on the frameworks and covers each component of the tuned Asymmetric SVD recommender system algorithms. Then, I discuss the verification method of the systems, test plans, and significant risks, followed by the results and future works.

2 BACKGROUND & RELATED WORK

This section explains the collaborative filtering method and the challenges we are facing in detail. Then I focus on different SVD algorithms used in recommendation systems in related research. The datasets used in associated pieces of research will also be introduced. The section focuses on the details of each technique, the advantages, and the theoretical nature of the study. Additionally, the analysis and comparison of the techniques in each algorithm and my Asymmetric SVD approach are presented.

2.1 Collaborative Filtering

A collaborative filtering model is built by collecting users' interactions on different items, then creating embeddings for every user and item [22]. It recommends to a particular user based on the reactions of other users who share similar tastes. Users' interactions have two main types: explicit and implicit. Explicit interactions are users' input regarding to their interest in an item. It is often measured by ratings or ranking provided by each user using one or more ordinal or qualitative scales [17]. In datasets used in related pieces of research and this paper, explicit users' preference information is shown in users' rating column.

Meanwhile, implicit interactions are information produced after observing users' behavior, such as movie rental history. There is no requirement for users to participate or gather this data by themselves, as the system will automatically track users' preferences by monitoring the performed actions, including which items they visited, clicked, or bought [1]. In other words, the data expressed users' preferences by rating values, which items users rate, and regardless of how they rated the items (high or low).

Traditional collaborative filtering algorithms include memory-based (User-User or Item-Item-filtering) and model-based methods. While neighborhood methods use direct interactions to find a similar group and then predict unrated items, model-based methods are more widely used with the application of matrix factorization [8]. Specifically, the matrix factorization model provides a decomposition of a rating matrix into two matrices representing users and items in a latent factor space [18]. The algorithm predicts the expected rating for a user who hasn't rated or bought the item yet. To make a rating prediction for an item, we look at the previous item's rating given by users who share a similar taste to the given user. In detail, we multiply two matrices, items' and users' entities, to predict the relationship between them - how the assigned users would rate the items [6].

With the massive growth of customers and products data, the matrix factorization approach of collaborative filtering method still faces three main problems to making accurate recommendations [2, 23]:

- `Responding time`: the neccessity of improving algorithms responding time applied in a huge amount of data
- `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, and even users that are very active rate just a few items compared to the total number
- `Scalability`: the collaborative filtering fails to scale up the computing time with the massive growth of both number of new users and items when making accurate recommendations

2.2 Singular Value Decomposition (SVD)

Zhou et al. [30] states that applying data dimension reduction methods, specifically SVD, is one of the most popular solutions for the sparsity problems. 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. Given a mx matrix A (N is the number of items, M is the number of users) with rank$(A) = r$, the $SVD(A)$ is defined as:

$$SVD(A) = U \times S \times V^T \tag{1}$$

where U, S, V are dimensions $m \times r$, $r \times r$, $r \times n$. While the middle matrix S is a diagonal matrix with r nonzero entries, which are the singular values of A, matrices U and V are orthogonal [30]. U and V are known as the $left$ and $right$ singular vectors, respectively with the first r columns of U corresponding to the nonzero singular values span the *columnspace*, and the first r columns of V span the *rowspace* of the matrix A [23].

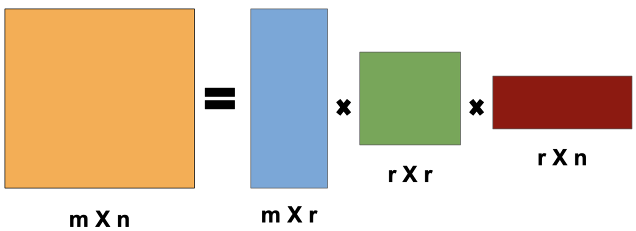

Figure 1: SVD Matrix Factorization

The accuracy of the SVD 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. SVD decreases the dimension of the utility matrix and be the best low-rank linear approximation of the original matrix A using three matrices multiplication [23].

According to Sarwar et al. [23], a dimensionality reduction is a popular approach in SVD, helping customers who rated similar products be mapped into the space spanned by the same eigenvectors. It is possible to keep the first k largest singular values in diagonal matrix S only $k \ll r$ and the remaining smaller ones set to zero by discarding other entries . The reduced matrix is denoted as S_k. Simultaneously, by deleting the corresponding $(r-k)$ columns from matrix U and $(r-k)$ rows from matrix V, we produce two reduced matrices U_k and V_k . The reconstructed SVD is represented as:

$$SVD(A_k) = U_k \times S_k \times V_k^T \tag{2}$$

which is the rank-k matrix closet approximation to the original matrix A. However, in my paper, I apply the new incremental SVD technique in the original matrix A instead of the reduced version A_k as in Sarwar's research. Applying the SVD method to a user-item matrix, a sparse rating of user u to item i can be estimated as:

$$\hat{r_{ui}} = b_u i + p_u^T.q_i \tag{3}$$

where $b_u i$ is the baseline estimate for an unknown rating $r_u i$ of user u to item i being with b_u and b_i are the average devisations of user u and item i from the average; being defined as

$$\hat{b_{ui}} = \mu + b_u + b_i \tag{4}$$

In equation (3), $p_u \in \mathrm{R}_k$ is the user u factor vector and $q_i \in \mathrm{R}_k$ is the item i factor vector.

2.3 Related Work - Different SVD-based algorithms

The time complexity of the SVD algorithm is calculated by batch and equals $O(m^2n + n^3)$ (where m,n are, respectively, row size and column size of the matrix) [30]. As SVD requires all the data be processed simultaneously, it has a challenge in not only dealing with a large dataset but also the expensive computing time when a new user or item is added to the system. Due to the limitation in running time with massive data of SVD-based algorithm, we need to experiment with diverse upgraded model-building technique for SVD to improve the scalability of recommender system.

Zhou et al. [30] 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.

Sarwar et al. [23] address this problem by designing the folding-in SVD literature, which allows new users and items to be added without affecting the existing ones. The model will use Latent Semantic Indexing (LSI) to reduce the dimensionality before applying the incremental technique folding in. Therefore, when a new user is incrementally folded-in the space, the user-item matrix is already reduced the size. As SVD decomposition using existing users and items is pre-computed, folding-in technique will take advantages to create a more scalable recommender system. Applied to the MovieLens dataset, the result shows that incremental algorithm speeds up computational time while provide comparable prediction accuracy.

Ghanzanfar et al. [9] created a based structure for the Iterative SVD algorithm, which works as a piece-wise function, finding the predicted values $r_i u$ via the combination of SVD and Expected Maximization algorithm. If a rating exists in the original rating matrix, the algorithm leaves the variable. If there is a null rating, Iterative SVD attempts to approximate it using SVD, then applies Expected Maximization to calculate the error evaluation repeatedly until the change between two iterations is less than a pre-determined threshold. This method is one of the solutions to the Netflix Prize.

Also, in Netflix prize, Koren [15] shows the SVD++ method that takes into account both implicit and explicit interactions, representing the highest quality in RMSE-optimized factorization methods. However, SVD++ doesn't work for the dynamic dataset, which means when a new user or item is added to the dataset, SVD++ needs to retrain the whole model, costing a lot of time. Moreover, Koren also constructed a new model named Asymmetric SVD that can handle new users or items added without retraining the model and estimating new parameters. The Asymmetric SVD has the advantage of both the SVD and SVD++ algorithm.

The new tuned Asymmetric SVD is reconstructed following Koren's work with optimal hyper-parameters, including the number of iterations the algorithm runs, learning rate, and the number of factors k in the diagonal matrix (sorted in descending order). Unlike SVD, SVD++, or TruncatedSVD using a built-in function in *sklearn* Python library, the new model has a customized Estimator object. The tuned Asymmetric SVD works well with dynamic datasets leading to the speeding up recommendation algorithm when new users are added, and efficiently integrating implicit feedback to improve the accuracy of prediction.

2.4 Datasets in related researches

This sub-section introduces the datasets used in collaborative filtering recommender systems. While different papers 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.

MovieLens [13] is the most common dataset among the research papers examined for this work. GroupLens provide the MovieLens dataset, a web-based research recommender system with over 20 million movie ratings and tagging activities, released 09/1997; updated 10/2016. There are various versions of MovieLens with a size range from 100,000 up to 1B ratings. The rating scale ranges from 1 to 5, where 1 represents dislike and 5 illustrates a strong preference. Other datasets are also used. Zhou et al. [30] used the Flixster dataset, containing more than 8M ratings from 786,936 users for 48.794 movies in Flixster.

3 DESIGN & IMPLEMENTATION

3.1 Dataset - MovieLens 100k

This paper uses MovieLens 100k dataset - the smallest dataset among all MovieLens versions of data, released on 04/1998 [11]. It is the recommended stable benchmark data with 100,000 ratings from 943 users on 1682 movies collected in seven months from September 1997 through April 1998. Each user has rated at least 20 films in this dataset and needs to fill in full demographic info (age, gender, occupation, zip). My algorithm mainly uses four columns, including the unique UserId (Customer Identification), the item ID (Netflix's unique movies identification code for each product), ratings (ranging from 1-5 based on customer satisfaction), and the timestamp of the rating (in UNIX time). Converting the dataset into the user-item matrix where rows represent users and columns represent items, I can apply tuning techniques and the Asymmetric SVD algorithm. The big problem of the original user-item matrix is sparsity - only 6.3% of entries are filled with ratings.

3.2 Framework of tuned Asymmetric SVD

As illustrated in Figure 2, the proposed systems intake both explicit and implicit feedback before tuning the three hyper-parameters to get the optimal values and then plugging them into the Asymmetric SVD algorithm. As Asymmetric SVD is the upgraded version from the SVD-based algorithm, modifying equation (3), the rating of user u to the item i can be estimated as:

$$\hat{r_{ui}} = b_{ui} + q_i^T (p_u + |N(u)| \sum_{j \in N(u)} y_j) \tag{5}$$

where:

- $(p_u + |N(u)| \sum_{j \in N(u)} y_j)$ is used to model a user u
- p_u is a free user-factors vector which is learnt from the given explicit ratings
- $|N(u)| \sum_{j \in N(u)} y_j$ represents the perspective of implicit feedbacks

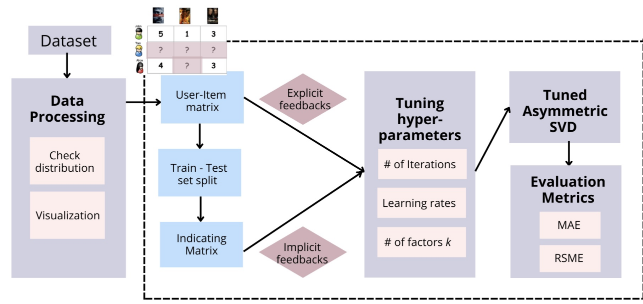

Figure 2: Framework of the tuned Asymmetric SVD recommender system

While the rating data from the user-item matrix coming from the MovieLens dataset works as explicit feedback, the implicit data is not available to us. Instead, we can reduce the rating matrix into an indicating matrix with binary values where "1" stands for rated and "0" stand for "not rated" to show users' preferences implicitly[15]. This way can count as a kind of implicit data to incorporate into the model, improving prediction accuracy. Moreover, modeling the movies, the user rated u_i as s function of binary vectors instead of fitting each u_i separately, decreasing the number of parameters. Model parameters are learned using gradient descent to minimize the associated squared error function.

To obtain optimal results, I fine-tune hyper-parameters, including learning rate γ, the number of features k, and choose the optimal number of iterations the algorithm will run. The approach is using scikit-learn's GridSearchCV to run through all the different hyper-parameters that are fed into the parameter grid and produce the best combination of hyper-parameters [25]. The values hyper-parameters were based on a 5-fold cross-validation splitting strategy with 20 jobs to run in parallel. The model runs with a learning rate γ fixed at 0.001, the number of features $k = 50$ through 200 iterations to reach the optimal performance.

Compared with regularized SVD and SVD++, Asymmetric SVD offers several benefits:

- `Fewer parameters`: While regularized SVD model has $O(Nk+Mk)$ parameters where N is the number of users, M is the number of movies, k is the number of features, the tuned Asymmetric SVD only has $O(Mk)$ parameters. The number of users is much greater than the number of items in our dataset. Therefore, when we exchange user parameters with item parameters, the model decreases the number of parameters and has lower complexity.
- `Handling new users`: Unlike SVD++ cannot work for a dynamic dataset, Asymmetric SVD handles new users added to the model well. There is no user information in the model involved; the new model doesn't need to retrain repeatedly when new users have their information input.
- `Implicit feedback integration`: While SVD cannot take into account the implicit feedback, Asymmetric SVD uses both implicit and explicit feedback for the prediction. Having implicit feedback as an additional indication of user preferences, the forecast becomes more accurate.

3.3 Evaluation Metrics

I evaluate the performance of the proposed recommendation algorithms according to accuracy metrics. I use a popular statistical accuracy measurement named Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) to show the closeness of predicted ratings to the actual ratings. The lower the MAE, the more accurately the recommendation engine predicts user ratings [26]. Meanwhile, RMSE puts more emphasis on a larger absolute error, and when RMSE is low, the accuracy of prediction is better [19]. MAE and RMSE of N corresponding rating prediction pairs are defined:

$$MAE = \frac{\Sigma_{i=1}^{N} |r_{ui} - \hat{r}_{ui}|}{N} \tag{6}$$

$$RSME = \sqrt{\frac{\Sigma_{i=1}^{N} (r_{ui} - \hat{r}_{ui})^2}{N}} \tag{7}$$

where r_{ui} represents the true rating on item i by user u, and $\hat{r}_{ui}$ shows the predicted rating on item i by user u.

4 EXPERIMENTAL RESULTS

For the experiment, I operated the regularized SVD model and then the tuned Asymmetric technique to compare the performance of the two algorithms. From that, I can evaluate the tuned Asymmetric technique implications. Graphs below show the evaluation metrics of the tuned Asymmetric SVD through 200 iterations, printing values every ten iterations.

Figures 3 and 4 show the result of the experiment. The dependence of the error metrics on the number of iterations the model runs through. We can conclude that increasing the number of iterations reduced the RMSE and MAE values, meaning better prediction accuracy.

Tables 1 and 2 shows the MAEs and RMSEs of SVD and tuned Asymmetric SVD performance. As directed, the adjusted Asymmetric SVD model performs better, giving MAE=0.7348 and RMSE=0.9324 on the test set compared with the 0.7535 and 0.9551 of the regularized SVD respectively.

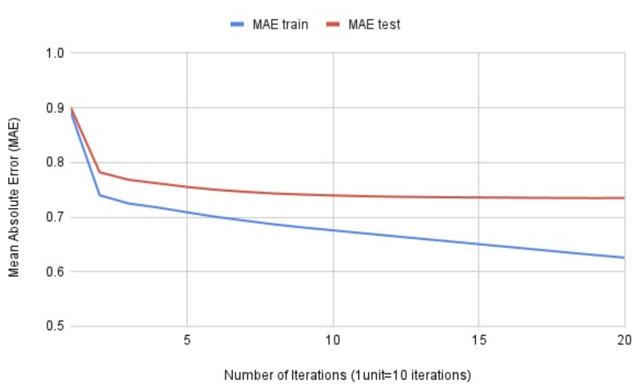

Figure 3: Dependence of tuned Asymmetric SVD MAE values on the number of iterations

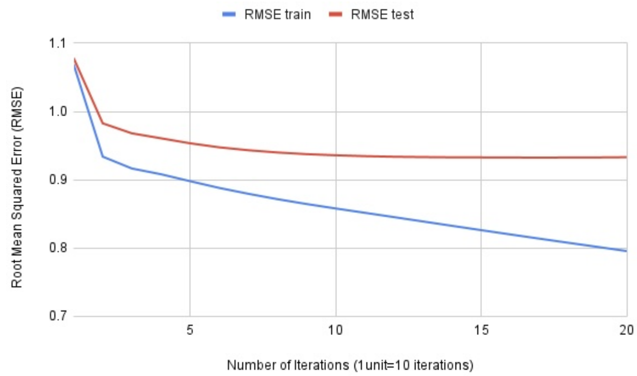

Figure 4: Dependence of tuned Asymmetric SVD RSME values on the number of iterations

	MAE train set	MAE test set
Regularized SVD	0.6709	0.7535
Tuned Asymmetric SVD	0.6257	0.7348

Table 1: A comparisons of the MAEs of SVD and tuned Asymmetric SVD on MovieLens-100K

	RMSE train set	RMSE test set
Regularized SVD	0.8912	0.9551
Tuned Asymmetric SVD	0.7955	0.9324

Table 2: A comparisons of the RMSEs of SVD and tuned Asymmetric SVD on MovieLens-100K

5 CONCLUSION AND FUTURE WORK

This work proposed improvements to one of the most popular Collaborative Filtering approaches- the SVD-based model. The upgraded SVD-based model, called Asymmetric SVD with tuned hyper-parameters, produced higher accuracy than the regularized version by modifying the rating prediction equation. The new model offers the advantages of having fewer parameters, conveniently handling new users, and efficiently integrating both implicit and explicit user feedback to the model. But again, the result comes from a small sample. Therefore, further investigation is needed to obtain more reliable accuracy. In the future, one could compare running time and accuracy trade-offs among SVD-based algorithms. Moreover, one could construct an iterative workflow for SVD-based algorithms working with dynamic datasets such as Asymmetric SVD, ApproSVD, Incremental SVD, and Appro Incremental SVD.

ACKNOWLEDGEMENTS

I would like to thank my advisors Dr. Charles Peck and Dr. David Barbella for guiding me in all aspects of the project.Thanks to MovieLens for releasing the data. Thanks to Yehuda Koren for sharing the approach to a Multifaceted Collaborative Filtering Model.

REFERENCES

[1] Zahra Ahmad. 2021. Recommender Systems: Explicit Feedback, Implicit Feedback and Hybrid Feedback. Retrieved Sep 20, 2021 from https://medium.com/analytics-vidhya/recommender-systems-explicit-feedback-implicit-feedback-and-hybrid-feedback-ddd1b2cdb3b

[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] GMI Blogger. 2021. Streaming Usage in the U.S. and Beyond – 2021 Statistics and Facts. Retrieved Mar 30, 2022 from https://cordcutting.com/streaming/2021-streaming-usage-stats/

[4] GMI Blogger. 2022. YOUTUBE USER STATISTICS 2022. Retrieved Mar 30, 2022 from https://www.globalmediainsight.com/blog/youtube-users-statistics/

[5] Chentung Chen and Weishen Tai. 2004. A User Preference Classification Method in Information Recommendation System.. In *ICEB*. 1091–1096.

[6] Denise Chen. 2020. Recommender System — Matrix Factorization. Retrieved Sep 14, 2021 from https://towardsdatascience.com/recommendation-system-matrix-factorization-d61978660b4b

[7] Roger Chua. 2019. A simple way to explain the Recommendation Engine in AI. Retrieved Sep 21, 2021 from https://medium.com/voice-tech-podcast/a-simple-way-to-explain-the-recommendation-engine-in-ai-d1a609f59d97

[8] Xiaowen Fang. 2014. *A Study of Recommender Systems with Applications*. Ph.D. Dissertation. UNIVERSITY OF MINNESOTA.

[9] Mustansar Ali Ghazanfar and Adam Prugel. 2013. The advantage of careful imputation sources in sparse data-environment of recommender systems: Generating improved svd-based recommendations. *Informatica* 37, 1 (2013).

[10] Stephen Gower. 2014. Netflix prize and SVD. *University of Puget Sound* (2014).

[11] grouplens. 1998. MovieLens. Retrieved Apr 3, 2022 from https://grouplens.org/datasets/movielens/

[12] Springboard India. 2019. Streaming in the U.S. - Statistics Facts. Retrieved Mar 30, 2022 from https://medium.com/@springboard_ind/how-netflixs-recommendation-engine-works-bd1ee381bf81

[13] GroupLens Kaggle. 2018. MovieLens 20M Dataset. Retrieved Sep 2, 2021 from https://grouplens.org/datasets/movielens/20m/

[14] Astha Khandelwal. 2021. How Does Amazon Netflix Personalization Work? Retrieved Mar 30, 2022 from https://vwo.com/blog/deliver-personalized-recommendations-the-amazon-netflix-way/

[15] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 426–434.

[16] Arkadiusz Krysik. 2021. How Does Recommendation Systems of Netflix, Amazon, Spotify, Tik Tok and YouTube Work? Retrieved Mar 30, 2022 from https://recostream.com/blog/how-does-recommendation-systems-of-netflix-amazon-spotify-tiktok-and-youtube-work

[17] Douglas W Oard, Jinmook Kim, et al. 1998. Implicit feedback for recommender systems. In *Proceedings of the AAAI workshop on recommender systems*, Vol. 83. WoUongong, 81–83.

[18] Fernando Ortega, Antonio Hernando, Jesus Bobadilla, and Jeon Hyung Kang. 2016. Recommending items to group of users using matrix factorization based collaborative filtering. *Information Sciences* 345 (2016), 313–324.

[19] Denis Parra and Shaghayegh Sahebi. 2013. Recommender systems: Sources of knowledge and evaluation metrics. In *Advanced techniques in web intelligence-2*. Springer, 149–175.

[20] 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

[21] Grand View Research. 2021. Video Streaming Market Size, Share Trends Analysis Report By Streaming Type, By Solution, By Platform, By Service, By Revenue Model, By Deployment Type, By User, By Region, And Segment Forecasts, 2021 - 2028. Retrieved Mar 30, 2022 from https://www.grandviewresearch.com/industry-analysis/video-streaming-market

[22] Abhijit Roy. 2020. Introduction To Recommender Systems- 1: Content-Based Filtering And Collaborative Filtering. Retrieved Sep 14, 2021 from https://towardsdatascience.com/introduction-to-recommender-systems-1-971bd274f421

[23] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2002. Incremental singular value decomposition algorithms for highly scalable recommender systems. In *Fifth international conference on computer and information science*, Vol. 1. Citeseer, 27–8.

[24] Rodrigo Schiavini. 2021. What is an E-commerce Recommendation Engine? Retrieved Sep 14, 2021 from https://www.smarthint.co/en/5-reasons-why-ecommerce-store-needs-recommendation-engine/

[25] scikitlearn. [n.d.]. MTuning the hyper-parameters of an estimator. Retrieved May 17, 2022 from https://scikit-learn.org/stable/modules/grid_search.html

[26] Guy Shani and Asela Gunawardana. 2011. Evaluating recommendation systems. In *Recommender systems handbook*. Springer, 257–297.

[27] Brent Smith and Greg Linden. 2017. Two decades of recommender systems at Amazon. com. *Ieee internet computing* 21, 3 (2017), 12–18.

[28] Enterprise Tech. 2021. Post-Pandemic Media Consumption: Online Streaming Accelerates A New Content Experience. Retrieved Mar 30, 2022 from https://www.forbes.com/sites/kyndryl/2022/02/11/why-the-mainframe-still-matters-3-tech-execs-explain-how-mainframes-work-in-concert-with-the-cloud/?sh=5d10979e6bbe

[29] Amy Watson. 2020. Streaming in the U.S. - Statistics Facts. Retrieved Mar 30, 2022 from https://www.statista.com/topics/1594/streaming/#dossierKeyfigures

[30] 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.