Tuning Baseline Singular Value Decomposition and comparing benchmark algorithms

Arambhumi Yaswanth Reddy
Affiliated to FSU, Florida State University, ar22br@fsu.edu

Lakshmi Harika Gamidi
Affiliated to FSU, Florida State University, lg22k@fsu.edu

1 Abstract

Recommender systems have brought about a transformative paradigm shift in the e-commerce landscape, notably witnessed on prominent platforms such as Amazon and Netflix. These systems have emerged as indispensable tools in elevating customer experience and driving sales. The trajectory of their evolution has been characterized by continuous innovation. Initially, the emphasis was placed on rudimentary aspects like item popularity and user-based demographics. However, the quest for more nuanced and accurate recommendations propelled the development of collaborative filtering methodologies.

In this comprehensive paper, we embark on an exploration of the diverse techniques employed in Matrix Factorization. Furthermore, we delve into the intricacies of Dimensionality Reduction through Singular Value Decomposition (SVD). Our exploration extends to a comparative analysis of benchmark algorithms, seeking to identify and optimize models for delivering superior recommendations. The ultimate goal is to compile a curated list of top-N recommendations tailored to individual users, thereby enhancing the overall recommendation system's efficacy and user satisfaction. As we navigate through these cutting-edge methodologies and benchmarking endeavors, we aim to provide a deeper understanding of the recommender systems landscape.

Keywords: Recommender systems, Collaborative Filtering, Matrix Factorisation, SVD, SVDpp, KNN, Dimensionality Reduction, Surprise

2 Introduction

Recommender Systems have been pivotal in transforming the landscape of online shopping. These systems have progressively shifted from rudimentary suggestion mechanisms to sophisticated, user-centric models, primarily through the development and integration of collaborative filtering techniques. This paper delves into the nuances of user-based collaborative filtering, focusing on advanced matrix factorization techniques and dimensionality reduction strategies, which are instrumental in addressing the inherent challenges such as data sparsity. Matrix factorization, particularly through the evolution of Singular Value Decomposition (SVD) algorithms, has emerged as a cornerstone in enhancing the efficiency and accuracy of recommender systems. The adaptation of SVD in collaborative filtering has been groundbreaking in mitigating the sparsity of user-item interaction matrices, a common challenge in recommendation scenarios with vast but sparsely populated datasets. By decomposing these matrices into lower-dimensional representations, we can extract latent factors that encapsulate user preferences and item characteristics, leading to more precise and relevant recommendations. In this paper, we will introduce and evaluate various techniques for user-based collaborative filtering. Our focus will be on fine-tuning hyperparameters and comparing benchmark algorithms such as SVD, SVD++, k-Nearest Neighbors (KNN), KNN

with K-means, Slope One, KNNBasic, Regularized SVD (RSVD), and baseline algorithms[3]. Each of these algorithms brings unique strengths and considerations to the table, and our comparative analysis aims to focus on these aspects.

3 Related Work

SVD++ Recommendation Algorithm Based on Backtracking[10]
This study explores the application of Collaborative Filtering (CF) in personalized recommendation systems, specifically focusing on the Singular Value Decomposition (SVD)++ algorithm. While SVD++ enhances prediction accuracy by incorporating implicit feedback, its computational efficiency remains a challenge. To address this limitation, a novel approach is proposed to accelerate the SVD++ algorithm's computation, aiming to improve recommendation accuracy. The proposed method employs backtracking line search within the SVD++ algorithm, optimizing the recommendation process and seeking the optimal solution through local gradient backtracking. Comparative evaluations with traditional CF algorithms on FilmTrust, MovieLens 1 M, and 10 M public datasets demonstrate the effectiveness of the proposed method, as evidenced by superior results in root mean square error, absolute mean error, and recall rate simulations.

Movie Recommendation and Rating Prediction using K-Nearest Neighbors

This study presents the development of a Movie Recommendation System employing the K-Nearest Neighbors (kNN) algorithm. The research systematically addresses critical stages, including data exploration, cleaning, and the integration of diverse features such as genres, cast, directors, and keywords, utilizing the TMDb dataset. The theoretical underpinnings of collaborative filtering, content-based filtering, and hybrid recommendation systems are elucidated. Additionally, the study implements a kNN-based score predictor, offering practical insights into system performance. In essence, this research provides a comprehensive examination of the intricacies involved in constructing a movie recommendation system using kNN, catering to both theoretical understanding and practical application.

Improving regularized singular value decomposition for collaborative filtering.

This paper presents an array of collaborative filtering algorithms, each demonstrating efficacy on the dataset from the Netflix Prize. Employing a strategy that combines the outcomes of multiple methodologies through linear regression, the researchers achieved an improvement of 7.04 percent in RMSE (Root Mean Square Error) values compared to Netflix's Cinematch, as seen in the evaluation set of the Netflix Prize competition. The suite of algorithms employed in this study encompasses those recommended by participants of the Netflix Prize, which include regularized singular value decomposition (SVD) for handling data with missing values, K-means clustering, and a combination of SVD and k-nearest neighbors (KNN) in a post-processing capacity. Building upon this foundation, they have then proposed an expansion of the predictive toolkit with several innovative approaches. These include the integration of biases into the regularized SVD framework, the application of kernel ridge regression as a post-processing step following SVD, the adoption of distinct linear models for each individual movie, and the exploration of methods akin to regularized SVD but characterized by a reduced parameter set. This expanded array of methodologies aims to enhance the predictive accuracy and efficiency of the model, adapting to the intricate dynamics of user preferences and movie characteristics.

Incremental singular value decomposition algorithms for highly scalable recommender systems.[6]
In this paper, the authors experiment with an incremental approach to building SVD-based recommendation models, which shows potential for high scalability and strong predictive accuracy. Specifically, they introduce an algorithm that expands on a pre-computed small SVD model to develop larger SVD models using cost-efficient techniques. The experimental results indicate that this algorithm operates at twice the speed of traditional methods

while delivering similar levels of predictive accuracy. We have incorporated the same by a k-folding approach to diversify the samples and optimise tha accuracy.

4 User-based collaborative filtering

Collaborative filtering stands as a cornerstone in modern recommender systems, particularly in platforms where product reviews and customer interactions form a vast web of data. Unlike content-based filtering, which relies solely on the attributes of items themselves, collaborative filtering delves into the preferences and behaviors of users. This method further classifies itself into user-based and item-based approaches. In user-based collaborative filtering, the system identifies users with similar preferences or behaviors and recommends products liked by these similar users. [2] Conversely, item-based filtering focuses on the relationships between items themselves, suggesting products similar to those a user has previously shown interest in. Our paper delves deeper into the user-based collaborative filtering techniques, and finding correlations between users' preferences for future recommendations of items to a user.

Central to this methodology is the use of a pivot matrix, often referred to as a utility matrix, where rows represent users and columns represent items. Each entry in this matrix signifies a user's interaction with an item, which could be explicit (like ratings) or implicit (like views or past purchases).[2] This matrix forms the foundation for discerning correlations between user preferences. In user-based filtering, the pivot matrix helps identify users with overlapping interactions, suggesting a similarity in preferences.

However, a significant challenge in this matrix arises from its inherent sparsity; not every user interacts with every item, leading to a matrix filled mostly with unpopulated cells. Addressing this sparsity, and extracting meaningful patterns from it, is crucial for the efficiency of the collaborative filtering process. Advanced techniques such as matrix factorization are often employed to mitigate this issue, enabling the system to uncover latent factors that represent underlying user preferences and item characteristics. These factors help understand user-user interactions and similarities, helping in the generation of more personalised recommendations.

5 Matrix-Factorisation

In the realm of recommender systems, the user-item interaction matrix is inherently large and sparse due to the extensive range of items and the interactions of numerous users. Each element of this matrix typically signifies a user's interaction with an item, such as a rating or a preference indicator, but the sparsity of interaction results in a matrix replete with unpopulated cells. Here's where matrix factorisation comes into picture, addressing the curse of dimensionality by decomposing the user-item matrix into more manageable matrices that reveal underlying latent factors and patterns.

The essence of matrix factorization in recommender systems lies in its approach to break down the original user-item matrix (R) into two lower-dimensional matrices: the user-factor matrix (P) and the item-factor matrix (Q). The relationship is mathematically denoted as

$R = P \times Q^T$

where Q^T is the transpose of Q[4]. This decomposition illuminates latent factors, which are essentially abstract features capturing the characteristics of items and preferences of users. For example, in a movie recommendation system, these latent factors for movies might encompass genres, while for users, they could reflect a penchant for specific genres.

User	Horror	Sci-Fi	Humanity	Drama	...
Alice	0.3	2.1	6.1	4.7	...

User	Horror	Sci-Fi	Humanity	Drama	...
God father	2.3	0.4	4.9	5.7	...

Figure 1: Latent factors for User Alice and Item Godfather

Although matrix factorization techniques have proven effective in recommender systems, they come with their own set of challenges. Firstly, the

determination of the number of latent factors is not straightforward and often requires extensive experimentation to balance the complexity and generalization of the model. If the number of latent factors is too large, the model might fit too closely with the training data and may not be able to generalise well for new samples, resulting in overfitting, and might underfit if the number of latent factors is small.
Some of the dimensionality reduction techniques and their variation algorithms aim to address some of the challenges, namely Singular Value Decomposition(SVD), Regularised Singular Value Decomposition(RSVD) which extends SVD by introducing regularisation or bias terms to prevent overfitting.[5] KNN for collaborative filtering works by finding the k closest neighbors to a user or item and making predictions based on their preferences. The method relies on similarity metrics like cosine similarity or Pearson correlation. Similarly, KNN with K-means is a variation that combines KNN with K-means clustering to improve scalability. Items or users are first clustered, and then KNN is applied within these clusters. SVDpp further extends SVD by incorporating implicit feedback, making it effective in handling sparse data. The model integrates the user's implicit interactions into the factorization process.The complexity of SVDpp is higher, but it offers improved accuracy, especially in scenarios with rich implicit feedback data, like that of large e-commerce giants like Amazon. In our execution, we will perform a comparison of the algorithms and dive deeper into Singular Value Decomposition and optimisation of recommendations.

6 Singular Value Decomposition

SVD is a matrix factorization technique commonly used for producing low-rank approximations. Given a matrix A_{mn} with rank(A) = r, the Singular Value Decomposition of A is defined as the following: $A = USV^T$,

where U_{mm}, V_{nn} and S_{mn}. The matrices U, V are orthogonal, with their columns being the eigenvectors of A A^T and A^T A, respectively. The middle matrix S is a diagonal matrix with r nonzero elements, which are the singular values of A. Therefore, the effective dimensions of these three matrices U, S and V are m × r, r × r and n × r, respectively.The initial diagonal r elements $\alpha 1, \alpha 2, ..., \alpha r$ of S have the property that $\alpha 1 \geq \alpha 2 \geq ... \geq \alpha r > 0$. An important property of SVD, which is particularly useful in recommender system, is that it can provide the optimal approximation to the original matrix A using three smaller matrices multiplication. By keeping the first k largest singular values in S and the remaining smaller ones set to zero, we denote this reduced matrix by S_k . Then by deleting the corresponding columns of U and V , which are the last r ⇒ k columns of U and V , we denote these two reduced matrices by U_k and V_k, respectively. The truncated SVD is represented as $A_k = U_k S_k V_k^T$.

Singular Value Decomposition (SVD) serves as a cornerstone in the field of recommender systems. In collaborative filtering, SVD enables the decomposition of the user-item interaction matrix, providing a mechanism to reveal latent patterns and preferences. This factorization allows for the creation of efficient, low-dimensional representations of users and items, capturing the underlying structure of the data. One notable advantage lies in its ability to handle missing values gracefully, crucial in scenarios where not all users have rated all items. The truncated SVD, denoted as $A_k = U_k S_k V_k^T$, plays a pivotal role in generating personalized recommendations. By retaining the top-k singular values and their associated vectors, it allows for a compact representation of the original matrix, effectively capturing the most significant interactions. This reduction in dimensionality not only facilitates storage efficiency but also contributes to enhanced computational efficiency in recommendation tasks. The collaborative filtering approach utilizing SVD has proven to be effective in revealing latent user preferences and item characteristics, forming the basis for many successful recommendation algorithms in real-world applications.

7 Dataset and EDA

This project uses the Amazon Product Review dataset curated by the University of San Diego(UCSD). This dataset includes reviews (rat-

ings, text, helpfulness votes), product metadata (descriptions, category information, price, brand, and image features), and links (also viewed/also bought graphs)[8]. We have used the Movie and TV ratings dataset, where the CSV file had the item ID, user ID, rating and the JSON file had the movie names, description, title, summary, reviews and other metadata related to the movies. We have combined the two dataframes to obtain a simplified version with ratings and movie titles as shown in the figure, where,
'reviewerID' denotes the ID of the reviewer, e.g. A2SUAM1J3GNN3B, 'asin' denotes the ID of the product, e.g. 0000013714, 'rating' being an overall rating of the product and 'title' denotes the Title of the Movie or TV show being rated.

	asin	reviewerID	rating	title
0	0001527665	A3478QRKQDOPQ2	5.0	Peace Child VHS
1	0001527665	A2VHSG6TZHU1OB	5.0	Peace Child VHS
2	0001527665	A23EJWOW1TLENE	5.0	Peace Child VHS
3	0001527665	A1KM9FNEJ8Q171	5.0	Peace Child VHS
4	0001527665	A38LY2SSHVHRYB	4.0	Peace Child VHS
5	0001527665	AHTYUW2H1276L	5.0	Peace Child VHS

Figure 2: Dataset snapshot

Upon loading the ratings for movies as dataframes, we then explored the dataset by exploring the distribution of ratings, top-rated movies and users with most ratings and so on, to get a better view of the dataset before applying the collaborative filtering techniques. This also enables us to select better samples while testing the algorithm, or one-off testing on individual users or items.

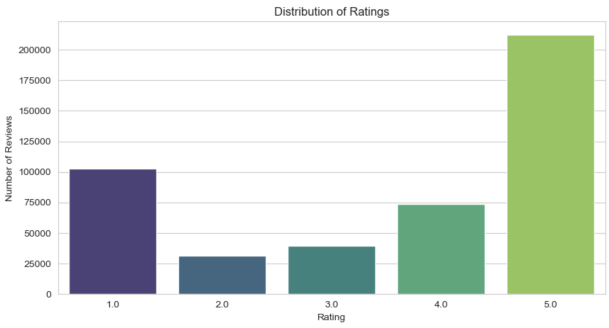

Figure 3: Distribution of ratings

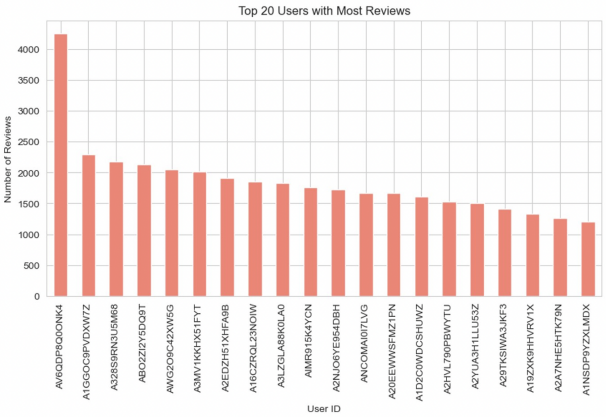

Figure 4: Top 20 Users

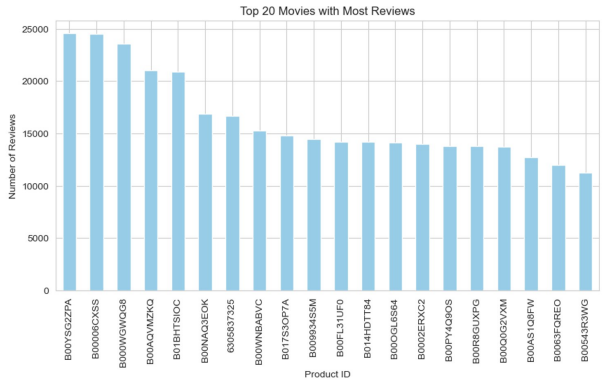

Figure 5: Top 20 Movies

8 Model Selection

In order to improve the performance of our system, we performed a comprehensive comparison and evaluation of various advanced algorithms and variations. These included KNNBaseline, SVDpp, KNNwithMeans, and KNNBasic, each offering unique strengths and capabilities in handling user-item interaction data. To rigorously assess the efficacy of these algorithms, we employed a 3-fold cross-validation technique. This method involves dividing the dataset into three parts, using two parts for training and the remaining one for testing, and rotating this process three times. Such a cross-validation approach ensures that each subset of the data is used for both training and testing, providing a comprehensive evaluation of the model's performance.
After calculating the RMSE for each algorithm, we arranged them in ascending order of their RMSE

values. This ordering provided a clear and objective means to compare the algorithms' performance. The algorithm with the lowest RMSE was identified as the most accurate, thereby enabling us to select the optimal algorithm for our recommender system. Since SVD offered the lowest RMSE, just above the baseline, we chose SVD algorithm to proceed further with the training.

Algorithm	test_rmse	fit_time	test_time
BaselineOnly	1.022461	0.091306	0.057847
SVD	1.023334	0.487221	0.067836
SVDpp	1.025483	33.117089	0.881494
KNNBaseline	1.031946	0.090931	0.076367
KNNWithMeans	1.084568	0.014641	0.082035
SlopeOne	1.087040	8.726848	0.363178
KNNBasic	1.090521	0.007004	0.077622
CoClustering	1.091252	2.511543	0.054965
NMF	1.100628	1.620182	0.066588
NormalPredictor	1.373919	0.034633	0.087059

Figure 6: Comparing test RMSE of SVD with some variations

Since SVD offered the lowest RMSE, just above the baseline, we chose SVD algorithm to proceed further with the training. In our efforts to optimize the performance of the Singular Value Decomposition (SVD) algorithm in our recommender system, we further tuned the hyperparameters to slightly reduce the RMSE for test predictions.

The number of latent factors in an SVD model is crucial for capturing the complexities of user-item interactions. We experimented with varying the number of latent factors to find the optimal balance between capturing sufficient information and avoiding overfitting.We fine-tuned the regularization parameters to prevent overfitting, especially important given the sparsity of the user-item interaction data. We have also adjusted the learning rate to optimize the convergence speed of the algorithm during the training phase.

After trial-and-error and taking the above factors into consideration, we have arrived at the following hyperparemeters, that reduced the RMSE value from 1.0042 to 0.99, and Mean Absolute Error(MAE) from 0.933 to 0.707:

Number of latent factors = 100
Number of epochs = 20
Learning rate = 0.007
random state = 10
regularisation parameter = 0.04

9 Model Evaluation

In our model evaluation process, we rigorously tested predictions on a strategically chosen subset of 10,000 rows. Adopting a 5-fold cross-validation strategy, we aimed to thoroughly assess the generalization performance of our model. The dataset was partitioned into five subsets, and the model underwent training on four folds while being evaluated on the remaining one in each iteration. This iterative process was repeated five times, ensuring a comprehensive exploration of various training and testing combinations. Our approach of employing cross-validation provides a robust evaluation metric that accounts for the potential variability in the data. By systematically rotating through different subsets, we obtain a more accurate estimate of the model's predictive capabilities on unseen data. This meticulous evaluation methodology is pivotal in gauging the model's effectiveness, identifying areas for improvement, and fostering confidence in its real-world applicability.

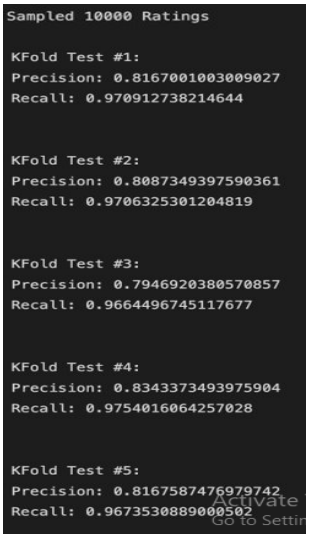

Figure 7: Sampled Set Evaluation: (10,000 Rows)

Intriguingly, our evaluation on the sampled subset of the dataset revealed a notable increase in recall, prompting a closer examination of potential contributing factors. One plausible explanation is that the sampled subset encapsulates a more diverse array of user preferences or represents specific user segments where our model excels. This diversity allows the model to better capture a wider range of user behaviors and preferences, leading to heightened recall rates. Additionally, the smaller size of the sample introduces a degree of variability, which, while potentially affecting precision, appears to positively impact recall. This finding underscores the importance of assessing model performance on various subsets to gain a comprehensive understanding of its strengths and areas for refinement.

```
Full Dataset

KFold Test #1:
Precision: 0.9223250937668417
Recall: 0.8641948160156651


KFold Test #2:
Precision: 0.9223464618392903
Recall: 0.8641980112520732


KFold Test #3:
Precision: 0.9224052025135365
Recall: 0.8638809873359804


KFold Test #4:
Precision: 0.9223479080416341
Recall: 0.8638508874119581


KFold Test #5:
Precision: 0.9225106550554031
Recall: 0.864692299410979
```

Figure 8: Full Dataset Evaluation

This elevated precision can potentially be attributed to the model's proficiency in generating more accurate predictions when provided with a larger training dataset. The expanded dataset size offers the model a more comprehensive learning experience, facilitating the discernment of intricate and nuanced patterns within the user-item interactions. Additionally, our findings suggest a significant alignment between the recommendations generated on the full dataset and user preferences, resulting in a reduction of false positives. This alignment underscores the model's effectiveness in delivering recommendations that resonate well with users, contributing to a higher precision rate. The elucidation of these dynamics provides valuable insights for model enhancement and emphasizes the importance of leveraging comprehensive datasets for training robust and accurate recommendation models.

10 Results

In presenting our conclusive findings, we harnessed the refined Singular Value Decomposition (SVD) algorithm to forecast user ratings across the entire dataset. Our methodology consisted of creating of a comparative bar graph, visually comparing the actual ratings against the predicted or estimated ratings. This graphical representation served as a pivotal tool for assessing the efficacy of our model.

Upon meticulous evaluation, our model demonstrated commendable performance on average, evidenced by the close alignment of estimated ratings within the decimal range of (0, 5). The nuanced depiction of actual versus predicted ratings in the bar graph highlighted the proficiency of our tuned SVD algorithm in accurately capturing user preferences and item interactions. This substantiates the reliability and effectiveness of our model in delivering precise recommendations.

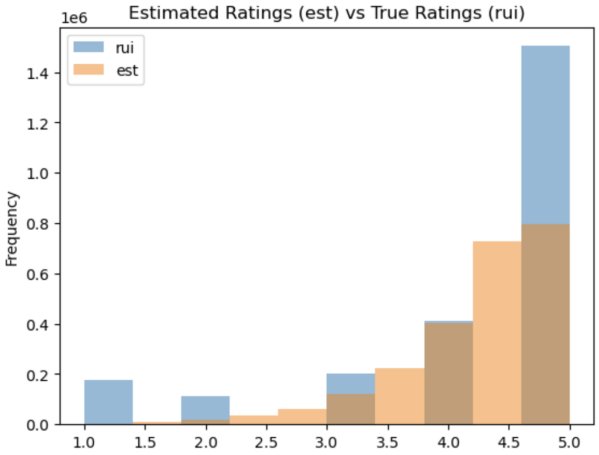

Figure 9: Estimated Ratings vs. True Ratings

We then chose the top-10 rated estimations for a test user in order to produce the final set of recommended movies/TV shows.

From the results, it can be seen that the show

'Touched by an angel: Season 8' received the highest estimated rating of 4.8, making it to the top of the list, followed by the movies Gone, Gran Torino, Peace Child VHS with estimated ratings of 3.66, 3.63, 3.60 respectively.

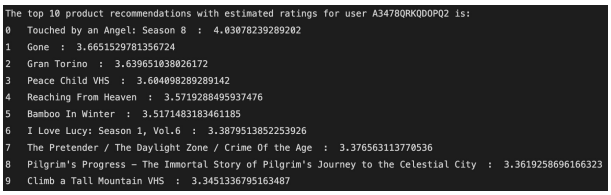

Figure 10: Top-10 recommendations for a test user

11 Challenges

While Singular Value Decomposition (SVD) and user-based collaborative filtering (CF) algorithms have been pivotal in advancing e-commerce recommender systems like Amazon, they encounter specific challenges that can impact their efficiency.

Computational Intensity: SVD is inherently computationally intensive, especially when dealing with large matrices typical in e-commerce platforms. Processing and decomposing these matrices to extract latent features require significant computational resources, which can be challenging in real-time recommendation scenarios.

Rating Score Range Ignorance: Standard SVD does not specifically consider the range of rating scores in its decomposition process. This can result in a reconstructed matrix that does not adhere to the original rating scale, necessitating additional post-processing steps to adjust the predicted ratings.

The Cold-Start Problem: A significant challenge for user-based CF is the cold-start problem, where making accurate recommendations for new users or items with little to no interaction history is problematic. In e-commerce, this is a frequent scenario as new products are continuously added, and new users join the platform.

Scalability: Both Singular Value Decomposition (SVD) and user-based Collaborative Filtering (CF) algorithms face scalability challenges when dealing with a rapidly growing user base or an expanding catalog of movies in recommendation systems. As the number of users and movies increases, these algorithms may encounter difficulties efficiently processing the expanding dataset, potentially leading to slower recommendation generation and response times.

Handling Unrated Movies: A specific challenge in movie recommendation systems is dealing with unrated movies. Users often do not rate all movies they interact with, resulting in missing data points. Treating these missing ratings as zero ratings can introduce bias and affect the accuracy of the recommendation models. Addressing this challenge is crucial to ensure that the recommendation system can effectively handle scenarios where users have not explicitly provided ratings for certain movies.

12 Conclusion

In the course of our extensive research focused on enhancing the top-N recommendation system for Amazon, we dedicated our efforts to significantly elevate the accuracy and reliability of predictive outcomes. Our approach revolved around a meticulous comparison of diverse matrix factorization algorithms, with a particular emphasis on evaluating their test Root Mean Square Error (RMSE) as a primary performance metric. Through this thorough comparative analysis, we successfully choose Singular Value Decomposition (SVD) as the optimal algorithm for our recommender system.

Our methodological framework encompassed the training of the algorithm on a representative sample dataset, followed by the implementation of a k-folding approach on the entire dataset. This dual strategy was employed to ensure not only the accuracy of our model but also its robustness and generalizability across diverse user profiles and item categories.

The rigorous nature of our process improved

the overall precision and recall scores. Furthermore, our approach facilitated the successful generation of a curated list comprising the top-10 recommended movies tailored to the preferences of a given user. This comprehensive research endeavor has not only propelled the effectiveness of Amazon's recommendation system but also contributes valuable insights to the broader landscape of recommender system optimization.

13 Future Scope

Moving forward, there are multiple ways we can further improve our recommendation system.

Addressing the Computational Intensity of SVD: While SVD offers an optimal balance currently, its computational intensity can be a limitation, especially as the dataset grows. Future research could explore more efficient algorithms like truncated SVD, where only a subset of singular values and vectors are used, can help reduce computational demands while still capturing the most significant latent features. By strategically selecting a limited number of singular values, truncated SVD provides a way to significantly lower computational requirements. This approach is particularly beneficial in scenarios where the full spectrum of latent factors is not necessary for achieving high-quality recommendations.

Another potential solution to the computational challenges posed by SVD is the adoption of parallel computing technologies. Frameworks like Apache Spark and Hadoop offer robust platforms for handling big data tasks through distributed computing. By leveraging these technologies, the task of matrix factorization can be distributed across multiple nodes, drastically reducing processing time and enhancing scalability. Apache Spark, for instance, is renowned for its in-memory computation capabilities, which can accelerate the processing speed of large-scale matrix operations. Hadoop, with its Hadoop Distributed File System (HDFS) and MapReduce programming model, presents another viable option for processing large datasets in a distributed and scalable manner.

Tackling the Cold Start Problem: To mitigate the cold-start problem in recommender systems, we might need to use hybrid recommender systems to combat the lack of interaction data. Hybrid systems bring together the strengths of both collaborative filtering and content-based approaches. While collaborative filtering excels in capturing user preferences based on interactions with items, content-based methods leverage item features (like descriptions, categories, tags) to make recommendations. By integrating these two approaches, hybrid systems can provide reliable recommendations even when interaction data is sparse. Actively incorporating user feedback mechanisms can help in quickly adapting recommendations to new user preferences, further mitigating the cold start problem.

Rating Score Range Adjustment: The issue of not inherently considering the rating score range in predictions is a notable limitation in many recommender system models, including those based on algorithms like Singular Value Decomposition (SVD) and its variants. In typical scenarios, these models generate predictions based on user-item interactions, but they may not align these predictions with the actual rating scale used in the system. This misalignment can lead to predictions that are outside the expected range of ratings, affecting the accuracy and usability of the recommendations.

Users typically expect ratings to fall within a specific range (e.g., 1 to 5 stars). Predictions that fall outside this range can be confusing and diminish the user's trust in the system. To mitigate this problem, a preprocessing step before feeding the data into the model can standardize ratings. This process involves scaling the ratings so that they have a consistent format and range, making the data more uniform. Techniques like min-max normalization can be used to scale the ratings to a fixed range, such as 0 to 1. This scaling helps the model in processing and generating predictions that are within a sensible range.

similarly, After the model generates predictions, a post-processing step can rescale these predictions back to the original rating scale. For example, if predictions are scaled between 0 and 1, they can be transformed back to a 1 to 5 scale.

References

[1] Jason Brownie(2020) Singular Value Decomposition for Dimensionality Reduction in Python.

[2] Shivam Balda(2023) Introduction to Collaborative Filtering.

[3] Paterek, Arkadiusz. (2007). Improving regularized singular value decomposition for collaborative filtering. Proceedings of KDD Cup and Workshop.

[4] Y. Koren, R. Bell and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," in Computer, vol. 42, no. 8, pp. 30-37, Aug. 2009, doi: 10.1109/MC.2009.263.

[5] R. Mehta and K. Rana, "A review on matrix factorization techniques in recommender systems," 2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA), Mumbai, India, 2017, pp. 269-274, doi: 10.1109/CSCITA.2017.8066567.

[6] Sarwar, Badrul Karypis, George Konstan, Joseph Riedl, John. (2002). Incremental singular value decomposition algorithms for highly scalable recommender systems. Fifth International Conference on Computer and Information Science.

[7] Sarwar, Badrul; Karypis, George; Konstan, Joseph; Riedl, John T. (2000) Application of Dimensionality Reduction in Recommender System - A Case Study.

[8] Justifying recommendations using distantly-labeled reviews and fined-grained aspects Jianmo Ni, Jiacheng Li, Julian McAuley Empirical Methods in Natural Language Processing (EMNLP), 2019.

[9] Blog post - Movie Recommendation and Rating Prediction using K-Nearest Neighbors.link

[10] SVD++ Recommendation Algorithm Based on Backtracking.link