

# Performance comparison and analysis of SVD and ALS in recommendation system

**Tianyi Zhao**

Department of Computer Engineering and Science, Shanghai University, Shanghai, 200444, China.

zhao\_tianyi@shu.edu.cn

**Abstract.** This research predominantly focuses on the electronic segment of the Amazon dataset. In this setting, this study's primary objective is to use this particular dataset to carry out a detailed comparative analysis of two matrix factorization-based collaborative filtering techniques, namely Singular Value Decomposition (SVD) and Alternating Least Squares (ALS). The findings stemming from this investigation reveal a notable contrast in the performance of these algorithms. Specifically, the SVD algorithm demonstrates significantly higher overall accuracy when compared to ALS. This observation suggests that in scenarios characterized by denser and smaller datasets, the SVD algorithm outperforms ALS by a considerable margin. The implications of these results underscore the significance of algorithm selection in recommender systems, emphasizing that the performance of collaborative filtering methods can vary markedly depending on the dataset's characteristics. Additionally, this research highlights the potential limitations of ALS in scenarios similar to the one explored here, shedding light on the importance of tailoring algorithmic choices to the specific data environment. Overall, these findings contribute valuable insights to the field of recommendation systems and provide guidance for algorithm selection based on dataset properties.

**Keywords:** Recommendation Models, Collaborative Filtering, Singular Value Decomposition, Alternating Least Squares.

## 1. Introduction

In today's digital age, recommendation systems have emerged as powerful tools that subtly influence daily lives. They have been under development since the early 1990s [1]. These sophisticated algorithms are the linchpin of the digital world, seamlessly connecting us to content, products, and experiences tailored to the unique tastes and preferences. At the heart of recommendation systems lies a complex web of algorithms meticulously designed to predict user preferences and curate suggestions accordingly. These systems thrive on data – vast oceans of it – analyzing user behaviors, historical interactions, and item attributes to uncover intricate patterns and correlations [2]. Armed with these insights, recommendation systems craft personalized recommendations that engage and captivate users. The versatility of recommendation systems knows no bounds, permeating numerous facets of the daily lives: Recommendation algorithms have completely changed how consumers shop online in the world of e-commerce. Online retail giants like Amazon rely on these systems to navigate customers through

a seemingly endless array of products [3]. By meticulously analyzing purchase histories and customer profiles, these systems suggest items that align seamlessly with users' preferences and past buying habits. Platforms like Netflix and Spotify have made content discovery an art form in the world of streaming services. [4,5]. These systems meticulously examine users' past viewing or listening behaviors, ensuring that the next movie or music selection feels tailor-made for each individual. In the sphere of news and content, recommendation systems silently curate the daily news feed. They sift through a vast sea of articles and videos, delivering content that mirrors the interests and aligns with the reading habits.

One of the key technologies is the collaborative filtering (CF)-based recommendation system. [6,7]. CF presents a systematic approach which furnishes users with tailored recommendations. Within CF, Singular Value Decomposition (SVD) and Alternating Least Squares (ALS) emerge as significant algorithms. This paper mainly compares SVD and ALS two algorithms based on matrix decomposition, explains the principle of the two algorithms in detail and analyses the differences in accuracy caused by their subtle differences. Through this exploration, the paper aims to equip researchers, practitioners, and stakeholders with the knowledge needed to navigate the ever-evolving landscape of recommendation systems effectively.

## 2. Method

### 2.1. Dataset

The dataset under consideration in this research comes from a curation project led by Dr. Jianmo Ni at UCSD. This dataset consists of links, product metadata, and reviews. Since the algorithms involved in this paper are all based on matrix decomposition, the reviewer ID, product ID and rating are mainly used.

### 2.2. Dataset Preprocessing

The role of data preprocessing is to conduct a comprehensive exploration of data attributes, including data complexity, component elements, distribution characteristics, etc. This comprehensive analysis provides an accurate basis for the establishment of the recommendation system, and also provides an essential reference for the selection and discussion of the algorithm. Therefore, before starting the subsequent algorithm, the necessary data preprocessor is essential, and Table 1 and Table 2 demonstrates the data statistics and the statistics of dfl in the dataset respectively.

**Table 1.** Dataset Statistics.

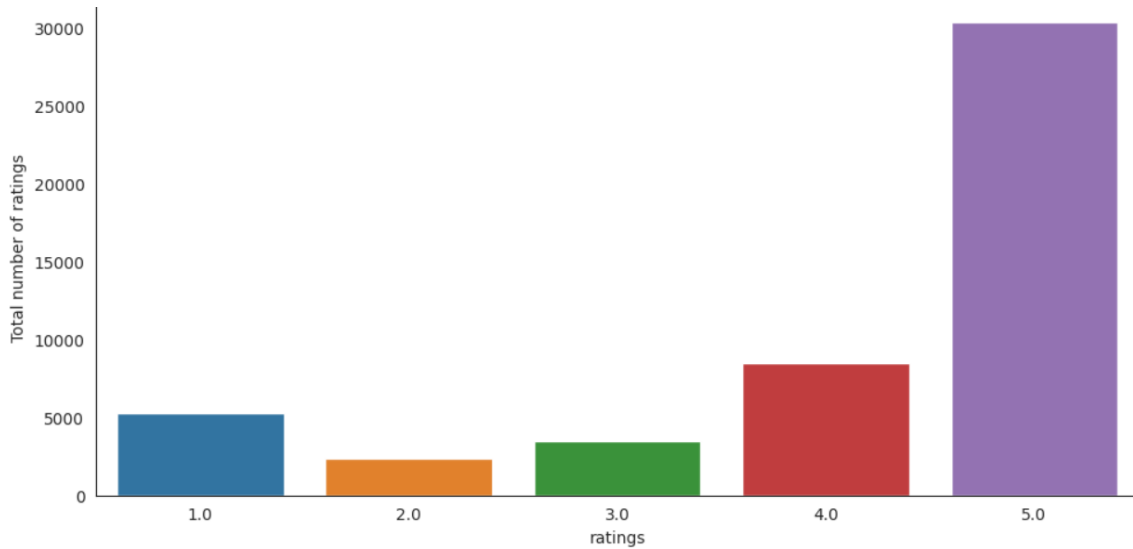
Metric	Electronics
Ratings Count	20994353
Users Count	9838676
Products Count	756489

Since the data is very big. Consider electronics\_dfl named data frame with first 50000 rows and all columns from 0 of dataset:

**Table 2.** Dataset Statistics of electronics\_dfl.

Metric	Electronics_dfl
Ratings Count	50000
Users Count	48682
Products Count	669
Ratings Mean	4.12
Ratings Std	1.34
Ratings Min	1
Ratings Max	5
Missing Count	0

In Electronic\_df1, there is no missing value. Therefore, the dataset does not need to do much additional preprocessing. These foundational metrics, coupled with descriptive statistics such as mean, standard, minimum, and maximum ratings, collectively afford a comprehensive contextualization of the dataset characteristics across various dimensions. It can be seen from the data that ratings mean as high as 4.07. The distribution diagram of ratings is displayed in Figure 1 to help with comprehension of the data set's intuitive distribution.



**Figure 1.** Histogram on Ratings for Electronics\_df1 (Figure credit: Original).

According to the histogram, it can be found that the evaluation of full marks accounts for more than half, the evaluation of four points is about one-fifth, and the evaluation of one point is slightly less than four points, but it also reaches one-tenth of the total number of ratings, indicating that users rarely give neutral ratings when evaluating, and usually three kinds of evaluations are very satisfied, basically satisfied and very dissatisfied.

### 2.3. Models

The performance of the underlying dataset is assessed using Singular Value Decomposition and Alternating Least Squares.

#### 2.3.1. Singular Value Decomposition (SVD)

SVD, a technique originating from the generalization of eigen decomposition principles for square normal matrices to matrices of arbitrary dimensions, made a significant impact when introduced by Simon Funk during the era of the Netflix Prize competition [8].

SVD achieves a decomposition that can be applied to rectangular matrices of real numbers by utilizing the eigen-decomposition of a positive semi-definite matrix. The core concept involves breaking down any matrix into three fundamental components: one diagonal matrix and two orthonormal matrices. When employed with a positive semi-definite matrix, SVD is essentially equivalent to eigen-decomposition. if  $A$  is a rectangular matrix, its SVD decomposes it as:

$$A = P\Delta Q^T \quad (1)$$

Here,  $P$  is the matrix  $AA^T$ 's (normalized) eigenvectors. The left singular vectors of  $A$  are referred to as the columns of  $P$ . The columns of  $Q$  are known as the right singular vectors of  $A$ , and  $Q$  is a reference to the (normalized) eigenvectors of the matrix  $A^T A$ .  $\Delta$  is the diagonal matrix of the singular values. In recommendation system,  $A$  is considered as the original user-item matrix,  $P$  is the user matrix,  $\Delta$  can be considered as the weight between users and items, and  $Q^T$  is the transpose of the item

matrix. The singular value matrix  $\Delta$  represents important features in the data. By retaining only the first few singular values in the  $\Delta$  matrix, it could receive a low-dimensional approximate matrix, reducing the dimensionality of the original data. This helps filter out noise and unimportant information while retaining the main features of the data. As a result, it can be observed that SVD is not suitable for excessively sparse data and encounters the cold start problem. When additional users or objects are added to the system, SVD necessitates a complete model recalculation, rendering it incapable of offering meaningful recommendations immediately.

In summary, SVD possesses certain advantages within recommendation systems, particularly in the realms of personalized recommendations and dimensionality reduction. However, it also exhibits constraints, notably when dealing with sparse data and real-time recommendations. In practical applications, it is often necessary to contemplate alternative recommendation algorithms and technologies to compensate for the limitations of SVD.

### 2.3.2. Alternating Least Squares (ALS)

ALS algorithm is fundamentally item-based collaborative, and in recent years, model-based recommendation algorithms like ALS have been successfully applied at Netflix, resulting in significant improvements in performance [9]. ALS utilizes machine learning algorithms to establish interaction models between users and items, enabling predictions of new items.

The core idea of ALS is to connect user interests with items through latent factors, based on user behavior to discover latent themes and categories. Then, it automatically clusters item and assigns them to different categories/themes (representing user interests). The  $m \times k$  user matrix  $U$  and the  $k \times n$  item matrix  $M$  are the two components that matrix factorization algorithms utilize to deconstruct the  $m \times n$  co-occurrence matrix  $R$ . There,  $k$  is the dimension of the latent vectors,  $n$  is the number of items, and  $m$  is the number of users. The size of  $k$  influences the strength of the latent vector's expressive capability, with larger  $k$  values conveying more information.

A  $k$ -dimensional latent vector represents each user and object. Therefore, to calculate a user's rating or other implicit behavior for a specific item, only the inner product of the corresponding vectors is needed to compute.

$$p_{ij} = U_i^T M_j = \langle u_i, m_j \rangle \quad (2)$$

Here,  $p$  reference to the corresponding vectors.  $U = [u_i]$  is the user feature matrix, where  $u_i$  denotes the  $i$ -th column of  $U$  and  $M = [m_j]$  is the item feature matrix, where  $m_j$  means the  $j$ -th column of  $M$ .

For users and items with interaction behavior, their rating bias is represented as  $r-p$ . When considering an individual sample:

$$L(r, u, m) = (r - p)^2 = (r - \langle u, m \rangle)^2 \quad (3)$$

Here,  $r$  represents the rating score of this individual sample. Therefore, the loss function on the entire sample is:

$$L(R, U, M) = \sum_{(i,j) \in I} L(r_{ij}, u_i, m_j) \quad (4)$$

where  $I$  is the index set of the known ratings. To minimize this loss function, the objective function is formulated as follows:

$$L_{\lambda}^{\text{reg}}(R, U, M) = \sum_{(i,j) \in I} [(r_{ij} - u_i m_j)^2 + \lambda(\|u_i\|^2 + \|m_j\|^2)] \quad (5)$$

In the formula, L2 regularization is employed to penalize the model's parameters to mitigate overfitting issues.

### 2.4. Evaluation Methods

RMSE: RMSE (Root Mean Square Error) is a commonly used metric for evaluating model performance, especially in regression problems [10]. It is used to quantify the size of the discrepancy or error between the values that a model predicts and the actual values that are observed.

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m (h(x_i) - y_i)^2} \quad (6)$$

Here, The i-th data point's actual observed value is denoted by  $h(x_i)$ , while the i-th data point's forecasted value is denoted by  $y_i$ .  $m$  refers to the whole amount of data points.

Fundamentally, RMSE measures the average size of prediction mistakes. As smaller prediction errors are represented by lower RMSE values, better predictive ability is indicated.

### 3. Result

SVD and ALS have been applied separately to the previous dataset, and the following will show the top four recommended items for the first three users by each algorithm, as demonstrated in Table 3 and 4 respectively.

**Table 3.** Recommended items for users from SVD.

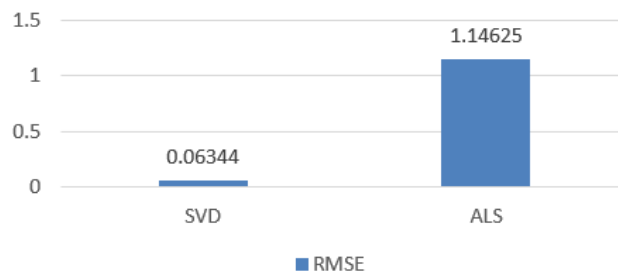
Items	Predictions user id = 1	Items	Predictions user id = 2	Items	Predictions user id = 3
B00000JD4V	2.176092	B00000JSGF	0.292601	B00001O2YP	2.386271
B00001QHP5	1.436154	B00001P4XA	0.227730	B00000J4O2	1.909017
B00000JMRV	1.296185	B00000JYLO	0.212570	B00000J3Q7	1.431763
B0000222MY	1.296185	B00001O2YP	0.207191	B00000J3II	0.721868

Each table provides personalized recommendations for an individual user, with the “user\_predictions” column indicating the predicted preferences or ratings for each recommended item. Higher values indicate stronger recommendations based on the algorithm's predictions. Some items (e.g., “B00001O2YP”) appear in the recommendations for multiple users, suggesting that these items may enjoy wider popularity or be better suited for a broader audience.

**Table 4.** Recommended items for users from ALS.

Items	Predictions user id = 1	Items	Predictions user id = 2	Items	Predictions user id = 3
Item 41	4.870513	Item 97	1.491549	Item 56	1.159948
Item 49	3.231210	Item 96	1.301014	Item 55	1.133313
Item 66	1.148225	Item 68	1.016828	Item 114	1.086499
Item 31	0.983692	Item 84	0.930076	Item 50	0.992916

Similar to SVD, these ALS recommendations are personalized for each user, offering items tailored to their potential preferences. Unlike SVD, there are no common items recommended across different users.



**Figure 2.** RMSE comparison between ALS, SVD (Figure credit: Original).

The results of this comparative evaluation reveal that the SVD model attains the lower RMSE values than the ALS model, which the RMSE of SVD equals to 0.06344 and the ALS equals to 1.14625, as displayed in Figure 2.

#### 4. Discussion

In this paper, RMSE was used to evaluate the two models respectively, and the results showed that SVD model performed significantly better than ALS, which may be caused by the following four reasons: a smaller subset is chosen, For small-scale datasets, SVD often finds better low-dimensional representations with lower computational costs because it directly decomposes the entire rating matrix. ALS may require more iterations to converge, especially on large-scale datasets. At the same time, this will also cause the data to be less sparse, ALS is typically better suited for handling sparse data because it can effectively deal with missing data points. ALS updates the latent feature vectors for users and items alternately, allowing modeling in the presence of a large number of missing values. In data processing, user data with more than five ratings is selected as a subset, which may lead to a certain linear relationship in the subset, that is, users give higher ratings to items with higher ratings. And SVD can better capture such relationships. ALS, on the other hand, minimizes the loss function through iterative steps and can also be used to discover linear relationships, but its advantage typically lies in adapting to non-linear relationships. This may also have created specific data noise, SVD relatively robust to noise or outliers in the data. Outliers do not usually have a significant impact on singular value decomposition. In contrast, ALS updates the model based on user-item interactions in each iteration and may be more sensitive to noise.

#### 5. Conclusion

In this comprehensive analysis and evaluation of recommender models, this work delves into the performance of two prominent matrix decomposition techniques, specifically SVD and ALS, within the context of the Amazon dataset. Through meticulous evaluation, this work aims to unravel the nuanced intricacies of these models, shedding light on their performance characteristics, predictive prowess, and a thorough exploration of the factors that might influence their precision and relevance.

Upon subjecting these models to rigorous scrutiny, the evaluation results unveil noteworthy disparities. SVD emerges as the frontrunner, demonstrating a notably superior performance on this dataset as evidenced by a remarkably low Root Mean Square Error (RMSE) of merely 0.06344, in stark contrast to ALS's RMSE of 1.14625. These findings underline the importance of selecting an appropriate recommendation algorithm that aligns with the dataset's underlying patterns and characteristics.

Furthermore, this analysis delves into the pervasive issue of data sparsity, which looms large as one of the principal challenges confronting recommendation systems. The paper not only highlights the existence of this challenge but also underscores the pressing need for ongoing research and continuous improvements to tackle it effectively. The complexities inherent in handling sparse data underscore the evolving nature of recommendation system research, opening doors to innovative solutions and strategies for enhancing their accuracy and relevance. As the field advances, addressing data sparsity remains a pivotal area for exploration and innovation.

#### References

- [1] Resnick, P., & Varian, H. R. (1997). Recommender systems. *Communications of the ACM*, 40(3), 56-58.
- [2] Beheshti, A., Yakhchi, S., Mousaeirad, S., Ghafari, S. M., Goluguri, S. R., & Edrisi, M. A. (2020). Towards cognitive recommender systems. *Algorithms*, 13(8), 176.
- [3] Linden, G., Smith, B., & York, J. (2003). Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1), 76-80.
- [4] Bennett, J., & Lanning, S. (2007). The netflix prize. In *Proceedings of KDD cup and workshop*, 2007, 35.

- [5] Madathil, M. (2017). Music recommendation system spotify-collaborative filtering. Reports in Computer Music. Aachen University, Germany 1-4.
- [6] Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 1-19.
- [7] Koren, Y., Rendle, S., & Bell, R. (2021). Advances in collaborative filtering. *Recommender systems handbook*, 91-142.
- [8] Kalman, D. (1996). A singularly valuable decomposition: the SVD of a matrix. *The college mathematics journal*, 27(1), 2-23.
- [9] Zhou, Y., Wilkinson, D., Schreiber, R., & Pan, R. (2008). Large-scale parallel collaborative filtering for the netflix prize. In *Algorithmic Aspects in Information and Management: 4th International Conference*, 4, 337-348.
- [10] Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1), 5-53.