Using Singular Value Decomposition To Recommend Movies

Kashyap Tanuku, Dev Patel, & Kartik Patel
Rutgers university
kt593@scarletmail.rutgers.edu
dbp128@scarletmail.rutgers.edu
khp76@scarletmail.rutgers.edu

ABSTRACT

This research uses Singular Value Decomposition (SVD) to predict movie ratings and recommend movies by discovering latent factors of user preferences and movie attributes. Data preprocessing and dimensionality reduction were applied to the MovieLens dataset, followed by a collaborative filtering approach that uses cosine similarity to compute user-item relationships. A k-nearest neighbors technique refines predictions and recommendations. The evaluation was done through metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE), which demonstrated optimal performance at $k = 121$ dimensions and 200 neighbors, achieving RMSE of 0.87 and MAE of 0.67. These findings highlight the potential of SVD-based methods for accurate and scalable recommendation systems.

KEYWORDS

Singular Value Decomposition (SVD); Collaborative Filtering; Cosine Similarity; Matrix Normalization; Matrix Factorization; Root Mean Squared Error (RMSE); Mean Absolute Error (MAE); Precision; Recall; F-measure; Normalized Discounted Cumulative Gain (NDCG); Idealized Discounted Cumulative Gain (IDCG); Discounted Cumulative Gain (DCG); Nearest Neighbors; Data Preprocessing; Concept Matrix

1 INTRODUCTION

Recommender systems are important for modern digital platforms like streaming and e-commerce services, fundamentally changing how users discover content and products. From Netflix's show/movie suggestions to Amazon's product recommendations, these systems play a crucial role in enhancing user experience, driving engagement, and maximizing service profit. This paper presents an intuitive approach to movie recommendations using Singular Value Decomposition and collaborative filtering methods.

Generating personalized movie recommendations relies on analyzing user-user and user-movie interactions. Most traditional methods struggle with the cold-start problem where there is not sufficient data in order to make recommendations. Our research addresses this challenges by implementing a system that uses Singular Value Decomposition (SVD) and cosine similarity.

Conference'17, Washington, DC, USA

DOI: 10.1145/nnnnnnn.nnnnnnn

We propose a framework that begins with the construction of a user-movie rating matrix, followed by its decomposition using SVD to extract latent features. The most important part of our approach is to utilize the V^T matrix, which is further processed to create a concept similarity matrix. This matrix serves as the foundation for identifying similar users and predicting movie ratings. Our system uses the concept similarity matrix to find similar users and generate recommendations based on rating patterns of similar users.

To validate our approach, we conducted extensive experiments using the MovieLens-Small dataset, a an extremely popular dataset in recommendation system research. To test the accuracy of our method, we calculated many different metrics to test both the accuracy of the predicted ratings and the relevance of recommendations. Our system showed strong performance in rating prediction:

- RMSE: 0.87
- MAE: 0.67

and recommendation quality:

- Precision: 0.92
- Recall: 0.6
- F-measure: 0.73
- NDCG: 0.96

These findings suggest that our approach effectively captures user preferences and generates relevant movie recommendations.

This research contributes to the field of recommender systems by presenting a practical implementation that balances computational efficiency with recommendation accuracy, making it suitable for real-world applications in content discovery platforms.

2 RELATED WORK

Recommendation systems have developed greatly, with collaborative filtering emerging as a dominant approach for predicting user preferences. User-user and item-item are two CF methods that use explicit ratings to recommend items, but they are not optimal with scalability and sparsity in big data [1]. So, Singular Value Decomposition (SVD) has become widely adopted, enabling the extraction of latent factors representing user preferences and item characteristics. This approach captures hidden patterns in the data, improving prediction accuracy and facilitating recommendations in sparse environments [2].

Recently, recommendation systems are improving with hybrid systems that combine CF with content-based methods, leveraging features like genre, tags, and textual metadata to improve recommendations and enhance personalization while lacking abundant historical data [3]. Deep learning methods, such as neural collaborative filtering and autoencoders, have also been proposed to model complex user-item interactions. However, these methods often require extensive computational resources and big data, making them less practical in limited resource scenarios [4]. Building on the strengths of SVD, our work integrates cosine similarity in the latent feature space and optimizes latent dimensions and neighborhood sizes, offering a scalable and efficient solution for accurate movie recommendations.

3 PROBLEM FORMALIZATION

Given

- Users Set: $U = \{u_1, u_2, \ldots, u_n\}$
- Movies Set: $M = \{m_1, m_2, \ldots, m_m\}$
- Rating matrix: $R \in \mathbb{R}^{n \times m} \mid r_{ij}$ = rating of user i for movie j
- $r_{ij} \in [0, 5] \mid 0$ represents no rating

Objective

(1) For user $u \in U$ and movie $m \in M$, predict the rating $\hat{r}_{um}$ and create a new predicted ratings matrix: $PR \in \mathbb{R}^{n \times m}$
(2) Create a recommendation list of k movies RL for user u

$$RL = \{rl_1, rl2, \ldots, rl_k\}$$

Such that

$$rl_1 = m = i \in max(\hat{r}_{ui}) | \hat{r}_{ui} \in PR_{um}$$

$$rl_2 = m = j \in max(\hat{r}_{uj}) | \hat{r}_{uj} \in PR_{um} \setminus \hat{r}_{ui}$$

$$\ldots$$

$$rl_k = m = k \in max(\hat{r}_{uk}) | \hat{r}_{uk} \in PR_{um} \setminus \{\hat{r}_{ui}, \hat{r}_{uj}, \ldots\}$$

Using Singular Value Decomposition (SVD)

$$R \approx USV^T$$

- $U \in \mathbb{R}^{n \times k}$ represents user latent factors
- $S \in \mathbb{R}^{k \times k}$ contains singular values
- $V^T \in \mathbb{R}^{k \times m}$ represents movie latent factors
- k is the number of latent factors ($k = 121$ in our implementation)

Similarity Matrix Computation

- After finding the SVD for the R, you find that V^T is the matrix of movie latent factors. Each row of V^T represents a concept or a genre.
- Multiplying the concept matrix with R results in the movie concept matrix $C_{m \times k} = R(V^T)^T$.
- Each row in C represents all the the users and each column represents the movies mapped to the different concepts.
- Each element in C shows how strongly a movie exhibits a latent concept on a scale from 0-1 where 0 means the movie is unrelated to the concept and 1 means the movie is fully related to the concept.
- Once you have C, you compute the cosine similarity which results in a matrix $CS \mid CS_{ij} = cosine\_similarity(p, q)$ where p is i^{th} row of CS and q is j^{th} row of CS.
- Cosine Similarity computes how similar two vectors are on a scale from 0 to 1 where 0 represents that the two vectors are orthogonal (vectors are not similar).

$$\text{Cosine Similarity}(i, j) = \frac{i \cdot j}{\|i\| \|j\|}$$

- CS is crucial to predict movie ratings because it is used to calculate similar neighbors to a specific user.

Rating Prediction

For a user u and a movie m, the predicted rating $\hat{r}_{um}$ is:

$$\hat{r}_{um} = \frac{\sum_{n \in N_k(m)} \text{sim}(m, n) \cdot r_{un}}{\sum_{n \in N_k(m)} |\text{sim}(m, n)|}$$

where $N_k(m)$ represents the k-nearest neighbors of movie m.

Evaluation Metrics

Root Mean Square Error (RMSE).

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum (r_{ij} - \hat{r}_{ij})^2}$$

Mean Absolute Error (MAE).

$$\text{MAE} = \frac{1}{n} \sum |r_{ij} - \hat{r}_{ij}|$$

Precision.

$$\text{Precision} = \frac{|\text{test items} \cap \text{recommended}|}{|\text{recommended}|}$$

Recall.

$$\text{Recall} = \frac{|recommended|}{|\text{test items}|}$$

Normalized Discounted Cumulative Gain (NDCG).

$$\text{NDCG} = \frac{\text{DCG}}{\text{IDCG}}$$

where:

$$\text{DCG} = \sum \frac{\text{rel}_i}{\log_2(i+1)}$$

$$\text{IDCG} = \sum \frac{2^{rel_i} -}{\log_2(i+1)}$$

4 THE PROPOSED MODEL

Our proposed model uses a collaborative filtering framework leveraging Singular Value Decomposition (SVD) to predict user ratings and recommend movies effectively. The use of latent factors which represent user preference help solve the challanges of sparsity and a large number of unrated data in the dataset.

The core idea of the proposed model involves decomposing the user-movie interaction matrix into three components: U, S, and V^T, where U captures the movie concepts, S represents the strength of concepts as singular values, and V^T contains movie-related latent features. In order to reduce computational complexity we truncate the SVD into a lower dimensionality k, where we isolate the most significant latent factors and thus reducing noise. Cosine similarity is then computed in this reduced latent space to evaluate user-item similarity, enhancing prediction accuracy.

For our predictions, we employ a k-nearest neighbors approach, where we identify the most relevant k neighbors for each user based on the computed similarity scores. Ratings are then predicted as a weighted sum of the neighbors' ratings. The optimal values f k-nearest neighbors and the k dimension were determined through multiple runs and testing of different values.

The model's performance and accuracy is then evaluated using Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE).

5 EXPERIMENTS

To test the effectiveness of our different models, we conducted multiple experiments to compare our user-user and user-item similarity approaches. The user-user method calculated the similarity between the users through their movie ratings. At the same time, the user-item approach focused more on the concepts using latent factors to predict a rating. Based on our testing, we found that the user-item algorithm was more effective in predicting user ratings as if it effectively captured the relationship between different movies.

We performed extensive experiments to further refine the model to determine the optimal value of k, the dimensionality reduction factor. To determine the best k value, we observed its impact on prediction accuracy as measured by Root Mean Squared Error (RMSE) and the Mean Absolute Error (MAE). We utilized the Rutgers supercomputer(Amarel) to reduce our computing time. After running our experiments, we found that small values of k led to underfitting and large values of k introduced noise. The optimal configuration was $k = 121$, which was the most effective value of k.

In addition to optimizing the value of k, we conducted a second experiment to determine the optimal number of neighbors (n). In our experiment, we varied n from 2 to 500 and analyzed its effect on the accuracy of the model. Once again running it on the supercomputer we found that prediction accuracy improved with an increasing number of neighbors, but beyond a certain point, the performance gains diminished. The optimal value was found to be $n = 200$, achieving the best trade-off between accuracy and computational cost. .

6 CONCLUSIONS AND FUTURE WORK

In this work, we proposed a collaborative filtering framework leveraging Singular Value Decomposition (SVD) to predict movie ratings and provide personalized recommendations. By focusing on the user-item similarity approach and optimizing key parameters such as the number of latent dimensions (k) and the number of neighbors (n), we achieved robust prediction accuracy. Using the Rutgers supercomputer enabled us to efficiently explore a wide range of configurations, resulting in an optimal setup of $k = 121$ and $n = 200$. For future work, we aim to extend the proposed framework by incorporating additional content-based features, such as movie genres, user demographics, and tags, to further refine recommendation accuracy. Moreover, exploring hybrid models that combine collaborative and content-based filtering approaches could unlock additional improvements in performance. Another important direction involves developing mechanisms to dynamically adjust the recommendation strategy based on user feedback, enabling the system to adapt over time. Lastly, we plan to use deep neural networks to capture more complex patterns and enhance scalability for big data.

The insights gained from this work demonstrate the potential of SVD-based approaches for building scalable, accurate, and fair recommendation systems.

7 OPTIONAL TASKS

An essential focus of our work was to ensure the development of a fair and unbiased recommender system. By using techniques that emphasize the latent relationships between users and movies rather than relying solely on explicit user feedback, our system mitigates biases introduced by popularity. This approach ensures fair recommendations across diverse user profiles, enhancing the inclusivity and fairness of the system.

ACKNOWLEDGEMENT

Yongfeng Zhang
Jure Leskovec
Anand Rajaraman

REFERENCES

[1] P. Resnick and H. R. Varian, "Recommender systems," *Communications of the ACM*, vol. 40, no. 3, pp. 56–58, 1997.
[2] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, no. 8, pp. 30–37, 2009.
[3] P. Lops, M. De Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends," in *Recommender systems handbook*. Springer, 2011, pp. 73–105.
[4] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *WWW*, 2017, pp. 173–182.