Enhancing Recommendor Sytems: A comparative analysis of matrix factorization techniques and collaborative filtering integration

Charanjit Singh*
CUID: C15246652
charans@g.clemson.edu

Parampreet Singh
CUID: C19377466
paramps@g.clemson.edu

Abstract

This paper presents the development of a sophisticated movie recommendation system employing diverse matrix factorization techniques. We explore and integrate methods such as Singular Value Decomposition (SVD), and Non Negative Matrix Factorization. By mathematically articulating each technique and demonstrating their effectiveness in capturing latent user-item interactions, we intend to evaluate their performance using RMSE and Mean absolute errors.

1 Introduction

The advent of digital streaming platforms has precipitated an unprecedented demand for sophisticated movie recommendation systems. These systems not only enhance user experience by personalizing content but also serve as pivotal tools for content providers to engage viewers effectively. In this research paper, we delve into the implementation of two robust matrix factorization techniques—Singular Value Decomposition (SVD) and Non-negative Matrix Factorization (NMF)—to construct a state-of-the-art movie recommendation system.

Matrix factorization algorithms have become the cornerstone of recommendation systems, owing to their efficiency in dealing with sparse datasets and their proficiency in uncovering latent factors that influence user preferences. SVD, in particular, has been lauded for its precision and versatility, allowing us to decompose the user-item interaction matrix into singular vectors that represent underlying features. This decomposition facilitates the prediction of a user's affinity for items they have not yet interacted with, by projecting both users and items into a shared latent space.

Complementing SVD, NMF offers a distinct approach by constraining the matrices to non-negative values, which aligns with the inherently positive nature of user ratings. This restriction not only ensures interpretability—where the factors can be viewed as the presence of features rather than the absence—but also aligns with the intuitive understandings of rating scales.

1.1 Collaborative Filtering

Collaborative Filtering (CF) is a vital technique in recommender systems, predicting user preferences based on others'. It comes in two forms: User-Based CF matches similar users for recommendations, while Item-Based CF suggests items similar to ones the user likes. CF can be Neighborhood-Based (using similarity metrics) or Model-Based (constructing predictive models). This technique addresses the data abundance challenge in digital platforms, enhancing content filtering and recommendation.

*

1.2 Matrix Factorization

The landscape of recommendation systems has been significantly shaped by the advent of Matrix Factorization (MF) techniques. MF methods have gained prominence due to their ability to efficiently handle large datasets and provide more accurate predictions compared to traditional CF methods. These techniques work by decomposing the user-item interaction matrix into lower-dimensional representations, capturing the latent factors inherent in the data. Several MF approaches have been developed, each with unique characteristics and applications. Singular Value Decomposition (SVD) is a fundamental technique that decomposes a matrix into singular vectors and values, offering a powerful method for dimensionality reduction. SVD factorization for recommendation:

$$
\begin{aligned}
R &: \text{User-Item Rating Matrix} \\
U &: \text{User Feature Matrix} \\
\Sigma &: \text{Singular Value Matrix} \\
V^T &: \text{Item Feature Matrix (Transpose)}
\end{aligned}
$$

SVD factorization for recommendation:

$$R \approx U\Sigma V^T$$

Minimize reconstruction error:

$$\min ||R - U\Sigma V^T||^2$$

As per the review-Based Collaborative Filtering, Using text-based embeddings for recommendations:

$$
\begin{aligned}
U_{\text{embedding}} &: \text{User Embeddings} \\
V_{\text{embedding}} &: \text{Item Embeddings}
\end{aligned}
$$

Review-Based Collaborative Filtering typically involves the use of user and item embeddings to calculate similarity scores based on user reviews. These similarity scores can then be used to make recommendations.

Calculating Cosine similarity: Cosine similarity is a metric used to measure the similarity between two non-zero vectors in an n-dimensional space. It's commonly used in various fields, including information retrieval, natural language processing, recommendation systems, and machine learning, to compare the similarity between two items or documents. Cosine similarity quantifies the cosine of the angle between the vectors, which reflects how similar or dissimilar the vectors are.

$$\text{Similarity score} = \text{Cosine Similarity}(U_{\text{embedding}}, V_{\text{embedding}}) \tag{1}$$

1.3 Challenges and Implementation

While matrix factorization methods have significantly improved recommendation quality, they come with their own set of challenges. One primary challenge is the cold start problem, where new users or items with limited interaction data are difficult to recommend. Additionally, scalability can be a concern as the number of users and items grows. However, advancements in these methods, such as the integration of biases and temporal dynamics, have made it possible to create more nuanced and accurate recommendation systems. Moreover, the adaptability of these methods to incorporate various forms of feedback and their ability to model temporal aspects have made them a powerful tool in the evolving landscape of recommender systems. In conclusion, the field of recommender systems is a dynamic and ever-evolving area of research, with collaborative filtering and matrix factorization techniques at its forefront. Understanding and leveraging these methods' strengths while addressing their inherent challenges is crucial for developing efficient, scalable, and accurate recommendation systems.

2 Methodology

2.1 Overview

The Netflix Prize dataset, utilized in the famous competition with a grand prize of 1 Million USD, serves as a cornerstone in this project. This extensive dataset is encapsulated in a file named 'training_set.tar', comprising 17,770 individual files. Each file represents a unique movie, beginning with the movie ID and a colon. What follows is a collection of customer ratings, structured in a format that includes the CustomerID, the given rating, and the date of the rating. These movie IDs are sequential, ranging from 1 to 17,770. The customer IDs, however, are more varied, ranging from 1 to 2,649,429, with some missing numbers, representing a total of 480,189 distinct users. The ratings provided are on a simple five-star scale.

Additionally, the dataset includes a file named "movie_titles.txt", which contains details about each movie. The format of this file lists the MovieID, the year of release, and the movie title. It's important to note that these MovieIDs are unique to this dataset and do not correspond to Netflix's internal movie IDs or those from IMDB. The 'Year of Release' might refer to the DVD release rather than the theatrical release and ranges from 1890 to 2005. Lastly, the title provided is the version used by Netflix, which may vary from titles used on other platforms. For this project, only 25% of the data was used.

2.2 Matrix Factorization using SVD

For a user-item ratings matrix R of size $m \times n$, where m is the number of users and n is the number of items, we aim to find matrices P (of size $m \times k$) and Q (of size $n \times k$), where k is the number of latent factors, such that $R \approx PQ^T$.

Applying SVD to R decomposes it into $U\Sigma V^T$, where U and V are orthogonal matrices, and Σ is a diagonal matrix of singular values. By retaining only the top k singular values and vectors, we approximate R as $R_k = U_k\Sigma_k V_k^T$. The latent factor matrices are then obtained as:

$$P = U_k\Sigma_k$$
$$Q = \Sigma_k V_k^T$$

To mitigate overfitting, a regularization term is added to the optimization problem, yielding the objective function:

$$\min_{P,Q} \sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}(r_{ij} - p_i^T q_j)^2 + \lambda(\|P\|_F^2 + \|Q\|_F^2)$$

Here, I_{ij} is an indicator function, λ is the regularization parameter, and $\|.\|_F$ is the Frobenius norm.

The gradients of the loss function with respect to p_i and q_j are:

$$\frac{\partial}{\partial p_i} = \sum_{j=1}^{n} 2I_{ij}(p_i^T q_j - r_{ij})q_j + 2\lambda p_i$$

$$\frac{\partial}{\partial q_j} = \sum_{i=1}^{m} 2I_{ij}(p_i^T q_j - r_{ij})p_i + 2\lambda q_j$$

These gradients are used to iteratively update P and Q:

$$p_i \leftarrow p_i - \alpha\frac{\partial}{\partial p_i}$$
$$q_j \leftarrow q_j - \alpha\frac{\partial}{\partial q_j}$$

The specific update rules become:

$$p_i \leftarrow p_i - \alpha(\sum_{j=1}^{n} I_{ij}(p_i^T q_j - r_{ij})q_j + \lambda p_i)$$

$$q_j \leftarrow q_j - \alpha(\sum_{i=1}^{m} I_{ij}(p_i^T q_j - r_{ij})p_i + \lambda q_j)$$

The process is repeated until the objective function's decrease is below a predefined threshold, indicating convergence. The SVD-based approximation of the ratings matrix, augmented with regularization, yields the latent factors P and Q. Optimization via first-order differential ensures the model's generalization by preventing overfitting and underfitting, thus revealing the underlying structure in the data for more accurate predictions.

2.3 Non Negative Matrix Factorization

In our methodology, we adopt the Non-negative Matrix Factorization (NMF) algorithm for collaborative filtering, ensuring all user and item factors remain positive throughout the computation. Following the strategy recommended in [7], which is in line with [8] in its non-regularized form, we apply a regularized stochastic gradient descent (SGD) optimization. This approach is selected for its direct applicability to dense matrices as well as its step size adjustment that preserves the non-negativity of factors. User and item factors are initialized between defined lower and upper bounds, a crucial step since the algorithm's performance is sensitive to these initial values. To address potential overfitting, we also have the option to implement a biased version of the algorithm by enabling the biased parameter, which incorporates baseline predictors into the factorization process. However, this biased approach requires careful consideration of the number of factors and the regularization parameters, λ_u and λ_i, to maintain a balance between accuracy and model complexity.

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u$$

$$p_{uf} \leftarrow p_{uf} \cdot \frac{\sum_{i \in I_u} q_{if} \cdot r_{ui}}{\sum_{i \in I_u} q_{if} \cdot \hat{r}_{ui} + \lambda_u |I_u| p_{uf}}$$

$$q_{if} \leftarrow q_{if} \cdot \frac{\sum_{u \in U_i} p_{uf} \cdot r_{ui}}{\sum_{u \in U_i} p_{uf} \cdot \hat{r}_{ui} + \lambda_i |U_i| q_{if}}$$

3 Evaluation Metrics

In movie recommender systems, evaluation metrics are crucial for assessing the performance of recommendation algorithms. These metrics help us understand how well the system predicts user preferences and how effective it is at suggesting relevant movies. Below we define some of the commonly used metrics:

MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It's the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight.

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |p_i - a_i|$$

Where p_i is the predicted value, a_i is the actual value, and N is the total number of predictions.

RMSE is a quadratic scoring rule that also measures the average magnitude of the error. It's the square root of the average of squared differences between prediction and actual observation.

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (p_i - a_i)^2}$$

4 Results

4.1 Overview

The algorithms using SVD and NMF were trained using 25% data from the Netflix Prize Dataset. The required RMSE and MAE metrics are calculated using the formulas given in section 3. The following table shows the Values achieved from implementing the two algorithms. As observed from the table below, SVD resulted in lower values of RMSE and MAE.

Algorithm	Evaluation Metric	Result
SVD	RMSE	0.85
	MAE	0.66
NMF	RMSE	0.91
	MAE	0.71

Table 1: Evaluation Results of Different Algorithms

4.2 Sampling recommendations for a random user

The algorithms were used to sample movie recommendations for a random user (83510). The movies liked by the user are shown below in figure 1.

Movie_Id	
57	Richard III
175	Reservoir Dogs
311	Ed Wood
329	Dogma
331	Chasing Amy
395	Captain Blood
788	Clerks
798	Jaws
907	Animal Crackers
985	The Mummy

Figure 1: Movies liked by a sample user 83510

The accompanying figures 2 and 3 provide a visual representation of the movie recommendations generated by our implementation of the Singular Value Decomposition (SVD) and Non-negative Matrix Factorization (NMF) algorithms. These snapshots capture a subset of the tailored suggestions derived from our models, highlighting the diversity and relevance of the movies presented to users. Notably, the SVD algorithm leverages latent factorization to predict user ratings, while NMF maintains non-negativity to offer interpretability. The depicted recommendations underscore the unique recommendations each method yields, illustrating the potential of matrix factorization techniques in personalizing user experiences in the digital streaming arena.

Recommendations using SVD:

	Year	Name	Estimate_Score
3455	2004.0	Lost: Season 1	4.602650
4426	2001.0	The West Wing: Season 3	4.514904
2113	2002.0	Firefly	4.464223
2101	1994.0	The Simpsons: Season 6	4.405251
3443	2004.0	Family Guy: Freakin' Sweet Collection	4.377196
1475	2004.0	Six Feet Under: Season 4	4.350491
3961	2003.0	Finding Nemo (Widescreen)	4.317341
2056	2001.0	Buffy the Vampire Slayer: Season 6	4.303222
2451	2001.0	Lord of the Rings: The Fellowship of the Ring	4.298282
4237	2000.0	Inu-Yasha	4.273250

Figure 2: Recommendations obtained by using SVD algorithm

Recommendations using NMF:

	Year	Name	Estimate_Score
17769	2003.0	Alien Hunter	3.59057
2	1997.0	Character	3.59057
7	2004.0	What the #$*! Do We Know!?	3.59057
15	1996.0	Screamers	3.59057
16	2005.0	7 Seconds	3.59057
17	1994.0	Immortal Beloved	3.59057
25	2004.0	Never Die Alone	3.59057
27	2002.0	Lilo and Stitch	3.59057
29	2003.0	Something's Gotta Give	3.59057
31	2004.0	ABC Primetime: Mel Gibson's The Passion of the...	3.59057

Figure 3: Recommendations obtained by using NMF algorithm

Future Scope

Cold Start Problem

- Develop hybrid models that incorporate content-based filtering techniques alongside collaborative filtering to provide initial recommendations for new users or items without historical interaction data.
- Utilize transfer learning methods to leverage patterns learned from existing users and items to make more accurate predictions for newcomers.

References

[1] Rui Duan, Cuiqing Jiang, Hemant K. Jain, Combining review-based collaborative filtering and matrix factorization: A solution to rating's sparsity problem, Decision Support Systems,

[2] Roy, D., Dutta, M. A systematic review and research perspective on recommender systems. J Big Data 9, 59 (2022). https://doi.org/10.1186/s40537-022-00592-5

[3] Dheeraj Bokde, Sheetal Girase, Debajyoti Mukhopadhyay, Matrix Factorization Model in Collaborative Filtering Algorithms: A Survey, Procedia Computer Science, Volume 49, 2015, Pages 136-146, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2015.04.237

[4] arXiv:1210.5631 [stat.ML] (or arXiv:1210.5631v2 [stat.ML] for this version) https://doi.org/10.48550/arXiv.1210.5631

[5] Folasade Olubusola Isinkaye (21 Nov 2021): Matrix Factorization in Recommender Systems: Algorithms, Applications, and Peculiar Challenges, IETE Journal of Research, DOI: 10.1080/03772063.2021.1997357

[6] https://datajobs.com/data-science-repo/Recommender-Systems-[Netflix].pdf

[7] Xin Luo, Mengchu Zhou, Yunni Xia, and Qinsheng Zhu. An efficient non-negative matrix factorization-based approach to collaborative filtering for recommender systems. 2014

[8] Sheng Zhang, Weihong Wang, James Ford, and Fillia Makedon. Learning from incomplete ratings using non-negative matrix factorization. 1996