RESEARCH ARTICLE

SAMRIDDHI Volume 15, Issue 4, 2023 Print ISSN: 2229-7111 Online ISSN: 2454-5767

Comparative Study of Matrix Factorization Techniques for Personalized Recommender Systems

Abhilasha Sankari*, Shraddha Masih, Maya Ingle

School of Computer Science and Information Technology DAVV, Indore

Abstract

Matrix factorization techniques are generally used in recommender systems to determine latent features of users and items, and to generate personalized recommendations based on these latent features. The main idea behind matrix factorization is to decompose the user-item rating matrix into two lower dimensional matrices - one representing users and the other representing items. The user and item latent factor vectors are then multiplied to generate the predicted ratings. In this paper, we present a comparative analysis of three matrix factorization techniques, UV decomposition, singular value decomposition and CUR algorithm for recommender system. We perform an experimental evaluation to compare the three techniques on two real datasets.

Keywords: Recommender Systems, Matrix Factorization, Personalized Recommendation

SAMRIDDHI : A Journal of Physical Sciences, Engineering and Technology (2023); DOI: 10.18090/samriddhi.v15i04.04

Introduction

Recommender systems are based on two main strategies, one is collaborative filtering and another one is content based filtering. The collaborative filtering approach relies on users past rating whereas the content filtering approach creates a user profile or product to characterize its nature. There are two primary types of collaborative filtering latent factor models and neighborhood methods. Latent factor models try to explain the ratings by characterizing both user and items. Whereas the neighborhood methods are centered on computing the relationships between user or items. Many successful realizations of latent factor models are based on matrix factorization techniques. Matrix factorization techniques characterizes both users and items by vectors of factors inferred from the rating patterns.[1]

Matrix factorization also deals with several challenges in recommender systems, such as the sparsity of the user-item matrix, the cold start problem for new users and items, and the scalability of the algorithms. Matrix factorization techniques overcome these challenges by leveraging the implicit feedback from the users, incorporating information about the items or users, and scaling to large datasets using distributed computing frameworks.

There are several matrix factorization methods used in recommender systems, such as Singular Value Decomposition (SVD), Non-negative Matrix Factorization (NMF), UV Decomposition. and Probabilistic Matrix Factorization (PMF). These methods differ in their assumptions about the distribution of the data, the objective function, and the regularization techniques used to avoid overfitting. Matrix factorization is a crucial component of modern recommender systems. It enables accurate and personalized recommendations to users in various domains such as e-commerce, social networks, and online advertising.

Corresponding Author: Abhilasha Sankari, School of Computer Science and Information Technology DAVV, Indore, e-mail: abhilasha.sankari@gmail.com

How to cite this article: Sankari A., Masih, S., Ingle M. (2023). Comparative Study of Matrix Factorization Techniques for Personalized Recommender Systems. *SAMRIDDHI : A Journal ofPhysicalSciences,EngineeringandTechnology,*15(4),406-409.

Source of support: Nil

Conflict of interest: None

Methodology

In this section we present three popular techniques namely singular value decomposition, UV decomposition and CUR algorithm for matrix factorization that are extensively used in recommender systems. In[2] SVD is utilized to perform two main tasks. Firstly, SVD is used to identify hidden correlations between users and item to predict the likelihood of a user to purchase a specific product. Secondly, SVD is used to reduce the dimensions of the user-item space. It generates a condensed representation, which is then utilized to compute neighborhoods. Based on this, the system can generate a list of top recommended products for a given user.

The paper[3] proposes a new framework for top-N recommendation based on non-negative matrix tri-factorization (NMTF). The proposed framework consists of a data imputation module and a top-N recommendation module. The modified NMTF model includes a context factorization component that captures the contextual information in the recommendation process

The approach proposed in[4] extends the traditional matrix factorization model by incorporating item metadata from a related domain, such as item descriptions, tags, or item reviews. The item metadata is used to learn a joint low-dimensional representation of both the item content and the user-item interactions. The proposed approach also employs a cross-domain regularization term to encourage the similarity between the item embeddings in the two domains. The experimental results on real-world dataset demonstrate that the proposed approach outperforms other state-of-the-art methods. This approach addresses the user cold start problem, especially when the amount of user interaction data is limited. The proposed approach also shows good scalability and can handle large-scale datasets.

Recently, deep learning-based matrix factorization techniques such as autoencoders and neural collaborative filtering have also been proposed for recommender systems. These techniques use neural networks to learn the latent features of users and items from the user-item rating matrix. Overall, matrix factorization techniques are widely used and effective in generating personalized recommendations in recommender systems.

SVD (Singular Value Decomposition)

Singular Value Decomposition (SVD)[5] is a matrix factorization technique that decomposes a matrix into three matrices: a left singular matrix, a diagonal singular value matrix, and a right singular matrix. In other words, given an m x n matrix A, SVD factorizes it into three matrices such that

$$A_{[mXn]} = U_{[mXr]}\Sigma_{[rXr]}(V_{[rXn]})^T$$

where A is input data matrix which is huge but sparse, U is left singular vector which is big, Σ is singular values which is sparse but small, r is the rank of matrix A and V is right singular vector which is dense.

In context of recommender systems, the singular value decomposition (SVD) is used to predict missing values in a user-item matrix. The user-item matrix represents the ratings that users have given to items, where each row corresponds to a user and each column corresponds to an item. SVD factorizes the user-item matrix into three matrices: a user-feature matrix, a feature-feature matrix, and an item-feature matrix. The user-feature matrix represents the latent features that characterize the preferences of each user, while the item-feature matrix represents the latent features that characterize the properties of each item. The feature-feature matrix represents the correlations between

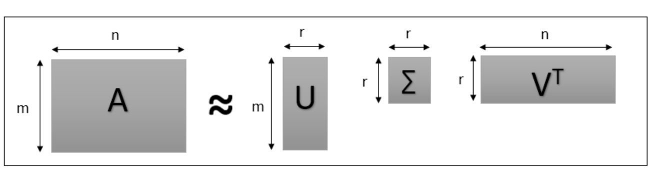

Figure 1: Illustrating the SVD theorem

the latent features. Once the user-item matrix is factorized, the missing values can be predicted by computing the dot product of the corresponding rows of the user-feature and item-feature matrices. This predicts the user's rating for the item based on their preferences for the latent features and the item's properties for the same features.

SVD is a popular matrix factorization technique for recommender systems because it can handle sparse and incomplete data, which is often the case in real-world scenarios. It can also be used to identify hidden patterns and relationships in the user-item matrix, which can help improve the quality of recommendations.

Some of the drawback of SVD is that it has optimal low-rank approximation in terms of Frobenius norm. Computational time of SVD is high and the result is dense vector that take more space.

UV Decomposition Algorithm

This is a matrix factorization technique that decomposes a matrix into two lower rank matrices. For example, if we have a matrix A of size m x n, we can decompose it into two matrices U and V such that

$$A = U \times V$$

where U is of size m x k and V is of size k x n, and k is a smaller rank than both m and n. The goal of UV decomposition is to find the values of U and V that minimize the difference between A and the product of U and V.

CUR Algorithm

This is a matrix factorization technique that selects a subset of columns and rows from the original matrix to create a smaller matrix. The goal of the CUR algorithm is to create a smaller matrix that preserves the important features of the original matrix.

CUR algorithm is a low-rank matrix approximation technique that uses column and row subset selection to reconstruct an approximation of the original matrix. The algorithm selects a subset of columns and rows based on their importance, computed using the leverage scores of the matrix. In the CUR algorithm, given an input matrix A, the goal is to approximate it using a product of three matrices, C, U, and R, such that

$$A \approx C\ U\ R$$

The matrix C is constructed by selecting a random subset of columns from A, while R is constructed by selecting a random subset of rows from A. The matrix U is then computed as the product of the pseudoinverse of C, denoted as C^+, the input matrix A, and the pseudoinverse of R, denoted as R^+, that is,

$$U = C^+ A\ R^+$$

The matrix C contains the selected columns from A, while R contains the selected rows from A. The matrix U is the low-rank approximation of A obtained from the selected columns and rows of A.

Therefore, the equation A = CUR represents the matrix decomposition where the original matrix A is approximated

Table 1: Time Complexity for all three algorithms

Technique	*Time Complexity*
CUR decomposition	O(nm)
UV decomposition	O(knm)
Singular Value Decomposition	O(kn+m)

as the product of matrices C, U, and R. The matrix C contains selected columns of A, the matrix R contains selected rows of A, and U is the low-rank approximation of A obtained from C, U, and R.

The CUR algorithm has been shown to be computationally efficient and effective for handling sparse matrices. The original paper on the CUR algorithm was published in.[6] Since then, several variations of the algorithm have been proposed, such as the robust CUR algorithm and the adaptive CUR algorithm.

The CUR algorithm has been applied in various fields, including recommender systems, image and signal processing, and machine learning. In recommender systems, the algorithm has been used to construct low-rank approximations of user-item rating matrices, leading to improved prediction. The CUR algorithm is a valuable technique for low-rank matrix approximation, particularly for sparse matrices. Its effectiveness and computational efficiency make it a valuable alternative to other matrix decomposition techniques.

Experimentation and Results

In this section we performed an experimentation analysis of UV decomposition, SVD, and CUR techniques for recommender systems based on their RMSE and MAE performance on the MovieLens and Epinions dataset.

Data Sources

We use two benchmark datasets, MovieLens and Epinions, to evaluate the effectiveness of the algorithms. MovieLens data sets were collected by the GroupLens Research Project at the University of Minnesota. The data set consists of 100,000 ratings (1-5) from 943 users on 1682 movies. Each user has rated at least 20 movies.[7] The Epinions dataset we used in our experiments is taken from the Epinions.com web site. Epinions is a consumers opinion site where users can review items and assign them numeric ratings in the range 1 to 5.

Performance Evaluation Metrics

To evaluate the performance of the three matrix factorization techniques, we used two datasets. The dataset was split into training and test sets. The training set is used to learn the user and item latent factors using various optimization techniques whereas test set is used to evaluate the performance of the learned model by comparing the predicted ratings to the actual ratings. Common metrics used to evaluate matrix factorization techniques include root-mean-square error (RMSE), mean absolute error (MAE), precision, recall, and F1 score. RMSE and MAE measure the accuracy of predicted ratings, while precision, recall, and F1 score measure the effectiveness of the recommender system in terms of how well it can identify relevant items for users.

Let n be the number of actual ratings in an item set, then MAE is defined as the average absolute difference between the n pairs. Assume that p1, p2, p3, ..., p_n is the prediction of users' ratings, and the corresponding real ratings data set of users is q1, q2, q3, ..., q_n. Then MAE is defined as:

$$MAE = \frac{\sum_{i=1}^{n} |p_i - q_i|}{n}$$

Smaller value of MAE indicates better accuracy.[8] To improve the accuracy of the techniques, the following hyperparameter is tuned:

- n_factors is the number of latent factors in the model. Increasing this value can improve accuracy, but can also increase the risk of overfitting.
- n_epochs is the number of iterations of the optimization algorithm. Increasing this value can improve accuracy, but can also increase the risk of overfitting.
- lr_all is the learning rate for all parameters. Increasing this value can help the algorithm converge faster, but can also make it more likely to get stuck in local minima.
- reg_all is the regularization parameter for all parameters. Increasing this value can help prevent overfitting, but can also reduce the accuracy of the model.

Table 1 summarizes the time complexity of the three techniques. Overall, the time complexity of these techniques can vary widely depending on the specific implementation, the size of the input matrix, and the rank or the number of factors used for decomposition. In practice, the CUR algorithm can be much faster than SVD for large sparse matrices, especially when the rank is small or the number of non-zero entries is small compared to the size of the matrix. However, SVD is often more accurate and can handle dense matrices and larger ranks, at the expense of higher computational complexity.

Table 2: MEA value for all three algorithms on MovieLens and Epinions datasets

Technique	*MovieLens Dataset*	*Epinions Dataset*
CUR decomposition	0.74	0.79
UV decomposition	0.65	0.47
Singular Value Decomposition	0.69	0.72

Table 3: RMSE value for all three algorithms on MovieLens and Epinions datasets

Technique	*MovieLens Dataset*	*Epinions Dataset*
CUR decomposition	0.98	1.00
UV decomposition	0.93	0.96
Singular Value Decomposition	0.95	0.98

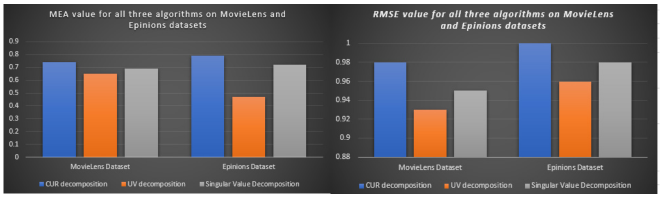

Figure 2:

Conclusion

In conclusion, we have presented a comparative analysis of three matrix decomposition techniques: UV decomposition, SVD, and CUR algorithm for recommender systems based on their MAE and RMSE values using MovieLens and Epinions dataset. We found that UV decomposition has lower RMSE AND MAE values than SVD and CUR algorithm on both datasets. Here, the smaller value indicates better performance. Table 2 report the RMSE values of comparison of three algorithms on Epinions dataset. Table 3 report the RMSE values of comparison of three algorithms on MovieLens dataset.

The results showed that UV decomposition outperformed SVD and CUR in terms of RMSE and MAE performance. UV Decomposition had the lowest RMSE value of 0.93, indicating that it provided the most accurate recommendations on the testing set. SVD decomposition had a slightly higher RMSE value of 0.95, indicating that it provided less accurate recommendations than UV Decomposition. CUR had the highest RMSE value of 1.00, indicating that it provided the least accurate recommendations among the three algorithms.

These results suggest that UV decomposition is a more effective algorithm than SVD and CUR for collaborative filtering-based recommender systems.

Overall, this comparative analysis provides insights into the performance of UV decomposition, SVD, and CUR algorithms for recommender systems based on their RMSE and MAE performance on the MovieLens and Epinions dataset.

It is important to note that the performance of these algorithms can vary depending on the dataset and the specific problem being addressed. Therefore, it is recommended to perform a comparative analysis of multiple algorithms on different datasets to determine the most suitable algorithm for a given problem. In addition, it is important to consider other factors such as computational efficiency, scalability, and interpretability when selecting an algorithm for a recommender system. SVD can be computationally expensive for large datasets, and CUR may not be as scalable as UV decomposition and SVD. UV decomposition and SVD are also more interpretable since they directly factorize the user-item rating matrix into user and item latent factor matrices. Therefore, the selection of an algorithm should be based on a comprehensive evaluation of multiple factors.

In future work we will compare the performance of matrix factorization techniques, such as alternating least squares (ALS), stochastic gradient descent (SGD), and non-negative matrix factorization (NMF), to identify the most effective technique for the given dataset and recommendation task.

References

[1] Y. Koren, R. Bell and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer,* vol. 8, no. 42, pp. 30-37, 2009.

[2] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, "Application of dimensionality reduction in recommender system-a case study," Minnesota Univ Minneapolis Dept of Computer Science, 2010.

[3] J. Tian and Y. Qu, "A Novel Framework for Top-N Recommendation Based on Non-negative Matrix Tri-Factorization," in *Proceeding of the 24th International Conference on Industrial Engineering and Engineering Management 2018*, 2019.

[4] I. Fernández-Tobías1, I. Cantador1, P. Tomeo2, V. W. Anelli2 and T. D. Noia2, "Addressing the user cold start with cross-domain collaborative filtering: exploiting item metadata in matrix factorization," *User Modeling and User-Adapted Interaction,* pp. 443-486, 2019.

[5] G. W. Furnas, S. Deerwester, S. T. Dumais, T. K. Landauer, R. A. Xarshman, L. A. Streeter and K. E. Lochbaum, "Information retrieval using a singular value decomposition model of latent semantic structure," *In ACM SIGIR Forum,* vol. 51, no. 2, pp. 90-105, 1998.

[6] P. Drineas, M. W. Mahoney and S. Muthukrishnan, "Relative-Error CUR Matrix Decompositions," *SIAM Journal on Matrix Analysis and Applications,* vol. 30, no. 2, pp. 844-881, 2008.

[7] Harper, F. Maxwell and J. A. Konstan, "The movielens datasets: History and context," *Acm transactions on interactive intelligent systems (tiis),* pp. 1-19, 2015.

[8] S. Gon, "A collaborative filtering recommendation algorithm based on user clustering and item clustering," *Journal of Software,* vol. 5, no. 7, 2010.

[9] S. Zhang, L. Liu, Z. Chen and H. Zhong, "Probabilistic Matrix Factorization with Personalized Differential Privacy," *Knowledge-Based Systems,* vol. 183, p. 104864, 2019.

[10] P. Symeonidis and A. Zioupos, "Matrix and tensor factorization techniques for recommender systems," *New York:Springer International Publishin,* vol. 1, 2016.

[11] Paterek and Arkadiusz, "Improving regularized singular value decomposition for collaborative filtering," in *Proceedings of KDD cup and workshop*, 2007.

[12] A. Rajaraman and J. D. Ullman, Mining of massive data sets, Cambridge university press, 2020.