Lightweight Similarity-Based Approach for Reducing User Interaction Data with Matrix Factorization in a Recommendation System

1st Yejung Lee
School of Computer Science and Engineering
Soongsil University
Seoul, South Korea
yejung@soongsil.ac.kr

2nd Yohan Park
School of Computer Science and Engineering
Soongsil University
Seoul, South Korea
imjin3027@soongsil.ac.kr

3rd Hyston Kayange
School of Computer Science and Engineering
Soongsil University
Seoul, South Korea
hyston@soongsil.ac.kr

4th Jihwan Um
School of Computer Science and Engineering
Soongsil University
Seoul, South Korea
djawlghks17@soongsil.ac.kr

5th Jongsun Choi
School of Computer Science and Engineering
Soongsil University
Seoul, South Korea
jongsun.choi@ssu.ac.kr

6th Jaeyoung Choi
School of Computer Science and Engineering
Soongsil University
Seoul, South Korea
choi@ ssu.ac.kr

***Abstract*— The accumulation of large-scale data in recommendation systems can significantly increase resource consumption during model training. In particular, when redundant data exists, the model may overfit as it evaluates test data based on already seen, similar data during the training process. To address the issues of resource consumption and overfitting, this study proposes a lightweight similarity-based algorithm to reduce interaction data. The proposed algorithm decomposes interaction data using matrix factorization to derive a user matrix and then calculates the similarity between users using cosine similarity. The data is then reduced by removing the data of users with fewer items among pairs of users whose similarity exceeds a certain threshold. Experiments were conducted to analyze the impact of the proposed algorithm on recommendation performance and training time after data reduction. Furthermore, the algorithm's effectiveness was evaluated through performance comparisons with existing data reduction methods. The primary contribution of this study is the introduction of a lightweight similarity-based approach that focuses on eliminating redundant user data in recommendation systems, thereby preventing overfitting and minimizing resource consumption.**

Keywords*— *lightweight similarity, data reduction, matrix factorization, overfitting prevention, resource saving, recommendation system

I. Introduction

A recommendation system is a technology used to recommend personalized items to users by utilizing data filtering and artificial intelligence techniques. It is widely used in various fields such as online shopping, music streaming, movie recommendations, news article recommendations, and social media feeds. These systems enhance user experience by analyzing users' past behaviors and preferences to suggest the most suitable items for them [1].

Collaborative filtering is one of the most commonly used approaches in recommendation systems. This approach generates recommended items by utilizing interaction data between users and items, based on the behaviors of users with similar preferences [2]. This interaction data is created through actions such as purchases, ratings, and clicks by users on items and such of kind of data serves as the foundation for developing collaborative filtering based recommendation systems. Collaborative filtering is generally divided into memory-based and model-based approaches. This paper focuses on model-based collaborative filtering.

As user-item interaction data accumulates over time, the scale of the data increases, leading to a rapid rise in the resources required for model training. Particularly, when redundant data is present, the model may overfit as it evaluates test data based on already seen, similar data during training [3]. Consequently, to effectively operate recommendation systems with large datasets, it is essential to employ data reduction techniques. In this paper, lightweight data reduction refers to the process of removing unnecessary information while retaining only the most essential data to minimize training time and resource consumption. We propose a method that considers similar users as redundant and unnecessary information, removing such data while preserving the critical information. The goal of data reduction in this study is to reduce training time and resource consumption.

In this paper, we propose a lightweight method for reducing interaction data by using Matrix Factorization (MF) to extract a user matrix, followed by generating a cosine similarity matrix based on this user matrix. If the similarity between users in this matrix exceeds a certain threshold, the user with fewer items among the highly similar users is removed. This approach aims to reduce the size of the dataset while minimizing any degradation in recommendation performance.

The experiments results analyze the impact of the proposed algorithm on recommendation performance and training time, and we evaluate the effectiveness of the algorithm by comparing its performance with existing data reduction methods. This paper seeks to explore the potential of a new approach to data reduction.

II. Related Work

Recommendation systems play a crucial role in delivering personalized content to users across various domains. Approaches for content recommendation include collaborative filtering, content-based filtering, and hybrid methods. Recently, there has been active research on collaborative filtering methods integrated with deep learning technologies.

For example, one research have proposed enhancing the performance of collaborative filtering by applying deep learning techniques and one research utilizing recurrent neural networks to handle time series data in collaborative filtering [4] [5]. Additionally, one research has shown that employing graph neural networks to model complex interactions among high-dimensional data can significantly improve recommendation system performance [6].

These studies have gained attention for their potential to address the issues of data sparsity and scalability, which are commonly cited as limitations of traditional collaborative filtering. However, as the volume of data increases, deep learning-based training processes that utilize large datasets may encounter challenges such as overfitting and increased operational costs [7]. To mitigate these issues, the introduction of various data reduction strategies is necessary. It has been demonstrated that it is possible to maintain reasonable recommendation accuracy while using minimal data [8].

In this study, we propose a method to reduce data by extracting latent features of users through Matrix Factorization from the user's interaction data, and then reducing the data based on the similarity of these latent features. Similarly, In our previous research work, we proposed a method to remove duplicate users to prevent overfitting. But it had limitations in effectively identifying similar users [9]. This study was conducted as one approach to overcome such a limitation.

While this study emphasizes the reduction of redundant data, other research has explored diverse strategies for data reduction. For instance, LightFR employs binary coding and privacy-preserving matrix factorization techniques to minimize communication and computation costs in large-scale federated recommendation systems [10]. Moreover, some studies have focused on reducing data by leveraging user preferences or time sequences, successfully preserving or even enhancing prediction accuracy for certain user groups [11]. Another notable approach involves the use of time windows to discard outdated data, retaining crucial features and thereby reducing the volume of data and processing time without significantly affecting accuracy levels [12].

This study presents a differentiated approach to existing data reduction methods and overfitting solutions. Previous studies [10][11][12] have suggested reducing data based on the most recently consumed or most preferred items by users. While this approach can retain user behavior patterns, it may leave redundant data, which might not be sufficient to prevent overfitting. In contrast, this study proposes a method to remove redundant users based on their latent similarity. This approach effectively reduces the overfitting problem that can arise when there are too many users within a user group. Furthermore, by utilizing Matrix Factorization and cosine similarity, this method achieves efficient data reduction in large-scale datasets with low computational cost.

III. Lightweight Data Processing Module

This study proposes a method to shorten training time and prevent overfitting by introducing a lightweight module in the data preprocessing stage to remove unnecessary information. The lightweight module detects similar linear relationships between users and reduces the data that exhibit similarity above a certain threshold.

Figure 1 shows a detailed implementation process of the recommendation system using the proposed data reduction module. This lightweight module consists of three stages. The first stage is Matrix Factorization, where the data, including users, items, and user ratings, are decomposed into k latent features, that creates a user matrix. The second stage is the generation of a similarity matrix, where the similarity of the latent feature values for each user are analyzed using cosine similarity to create a similarity matrix. The third stage is the reduction of interaction data, where, if the similarity between users in the similarity matrix exceeds a certain threshold, the data of the user with fewer items among the similar users is subsequently removed. After this process, the reduced interaction data undergoes basic data preprocessing, followed by training the model.

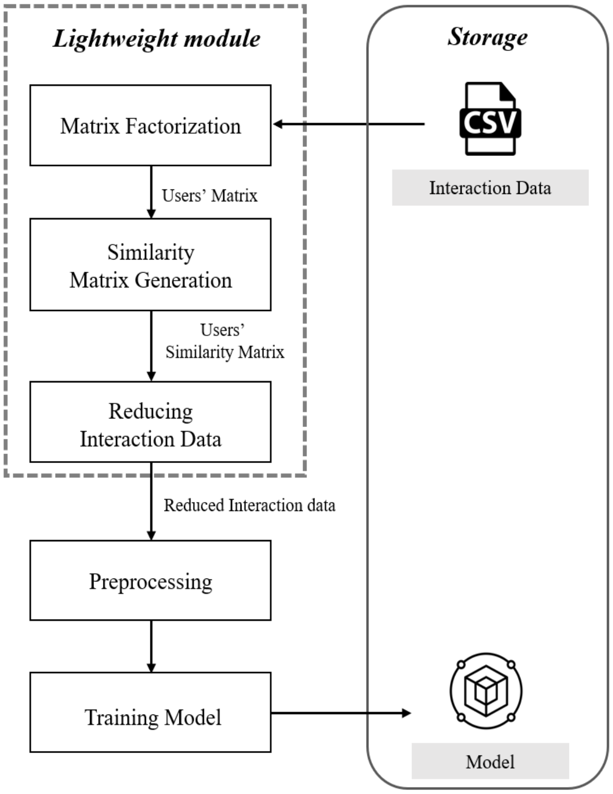

Fig. 1. Recommendation system training structure with applied lightweight module

3.1 Matrix Factorization Algorithm

Matrix factorization is used to reduce the dimensionality of data and to obtain a user matrix that extracts the latent features of the data. Matrix factorization plays a role in memory-based collaborative filtering methods, where the computation of similarity and preference becomes expensive when users have a large number of items. Additionally, it helps in finding similar user cases by allowing the calculation of users' latent preferences in subsequent similarity computations.

The MF algorithm decomposes interaction data into k latent features. The user-item rating matrix contains rating information for m users and n items. The objective of the algorithm is to decompose the interaction data into two lower-dimensional matrices: P and Q. P represents the latent feature vectors for the m users, and Q represents the latent feature vectors for the n items. This decomposition is detailed in Eq. (1), where P and Q correspond to the latent feature matrices for users and items, respectively.

$$\mathbf{R} \approx \mathbf{P}\mathbf{Q}^T \qquad (1)$$

In this study, we use Stochastic Gradient Descent (SGD) to optimize the latent feature vectors for users and items, as described by Eq.(2)

$$\min_{\mathbf{P},\mathbf{Q}} \sum_{(i,j)\in\mathcal{K}} \left(R_{ij} - \mathbf{P}_i \cdot \mathbf{Q}_j^T\right)^2 + \lambda\left(\|\mathbf{P}_i\|^2 + \|\mathbf{Q}_j\|^2\right) \qquad (2)$$

In this context, κ denotes the set of user-item pairs for which ratings are available, while λ serves as a regularization parameter to prevent overfitting. The SVD algorithm proceeds as follows: First, the latent feature matrices P and Q are initialized randomly. Next, the prediction error for each rating Rij is computed, and the parameters Pi and Qi are updated accordingly. This process is repeated until the parameters converge. The detailed algorithmic steps are showed in Table Ⅰ.

TABLE I. MF ALGORITHM

```
Algorithm 1 : Matrix Factorization
Input : R (User-item interaction data)
Output : U (Users' Matrix)
Set : latent factors k

1: Convert to Sparse Matrix R
2: Decompose the sparse matrix R :
      P for users and Q for items each linked to latent factors
3: while not converged do :
      for each non-zero rating Rij in M:
          Predict Rij using Pi and Qi
          Calculate error eij = Rij – Rij
          Update Ui and Vj using stochastic gradient descent
4: Extract the final User Latent Factor Matrix U
```

3.2 Similarity Matrix Generation Algorithm

Based on the user matrix generated in Section 3.1, a user similarity matrix is created. The primary purpose of generating this similarity matrix, as shown in Table Ⅱ, is to identify duplicate users based on their latent preferences. This similarity matrix quantitatively expresses the preference similarities between users, helping to select which user's data has to be removed during the data reduction stage.

The user matrix represents each user's preferences across various latent features in the form of vectors. To calculate the similarity between these user vectors, we employ Cosine Similarity. Cosine Similarity measures the similarity by calculating the angle between two vectors, with values closer to 1 indicating that the two users have similar preferences.

The process of generating the similarity matrix is as follows: First, the cosine similarity between the vectors in the user matrix is calculated to generate the similarity matrix. Then, the resulting similarity matrix is saved as a CSV file for use in the reduction algorithm.

TABLE II. USER SIMLIARITY MATRIX GENERATION ALGORITHM

```
Algorithm 2 : User Similarity Matrix Generation
Input : U (Users' Matrix)
Output : S (User Similarity Matrix)

1: Similarity Calculation
Treat each row of the user matrix U as a vector and compute the cosine similarity between these vectors.
2: Similarity Matrix Generation
Generate the similarity matrix S based on the computed cosine similarities
3: Similarity Matrix Storage: Save the resulting similarity matrix as a CSV file.
```

3.3 Reducing Interaction Data Algorithm

The Reducing Interaction Data stage is a process designed to efficiently reduce data based on the similarity matrix. The goal of this algorithm, as shown in Table Ⅲ, is to maximize system efficiency by removing unnecessary data while maintaining the performance of the recommendation system.

Specifically, the lightweight algorithm identifies pairs of users in the similarity matrix whose similarity exceeds a certain threshold. If two users have similar preferences, the algorithm identifies the user with fewer items. The data associated with that user is then removed, reducing the overall size of the dataset. This approach helps to reduce training time and resource consumption while also preventing overfitting.

The primary objective of this algorithm is to reduce the amount of data while preserving essential information, thereby maintaining the performance of the recommendation system.

TABLE III. REDUCING INTERACTION DATA ALGORITHM

```
Algorithm 3 : Reducing Interaction Data
Input : S (User Similarity Matrix)\
        R (Interaction Data)
Output : R'(Reduced Interaction Data)
Set : similarity threshold k

1: Identify similar user pairs:
    For each user pair (i, j) in S where i is not equal to j :
        If Sij > threshold, add (i, j) to the list of similar pairs
2: Remove users with fewer ratings:
    for each pair (user1, user2) in similar pairs:
        Compare the number of ratings for user1 and user2.
        Identify and mark the user with fewer ratings for removal.
        Ensure no user is marked for removal more than once.
3: Reduce ratings data:
        Remove the identified users from the original ratings data R to obtain the reduced data R'

4: Save the reduced dataset:
        Save the filtered ratings data R' to a new CSV file.
```

IV. Expreiment

In this study, experiments were conducted to validate the effectiveness of the proposed algorithm. The first experiment evaluated how much data could be reduced from a large dataset and analyzed the impact of this reduction on recommendation performance. The second experiment assessed how the reduced data affected the model's accuracy and training time. Finally, the proposed method was compared with existing techniques to demonstrate its ability to maintain both performance and efficiency during large-scale data reduction. The experiments were conducted using the MovieLens-1M dataset, with the proposed lightweight module applied before data preprocessing. The recommendation model, specifically the NCF (Neural Collaborative Filtering) model, was then trained, and its performance was evaluated using RMSE and NDCG metrics.

4.1. Data Reduction

The data used in this study is from the ratings.dat file of the MovieLens-1M dataset [13]. The dataset consists of 1,000,209 entries, including UserID, MovieID, Rating, and TimeStamp. UserIDs are categorized into 6,040 groups, and MovieIDs are categorized into 3,952 groups. Ratings range from 1 to 5, representing the users' ratings of the movies. The reduction rates achieved using the proposed reduction module are shown in Table Ⅳ, with an average time of 41 seconds required to reduce the data.

TABLE IV. Data Reduction and Performance Metrics

MovieLens-1M Data				
Similarity Threshold	***Reduction Rate(%)***	***RMSE***	***NDCG***	***Train time***
Original	0%	0.21	0.47	4,833
95%	38.2%	0.211	0.449	3,706
93%	43.4%	0.217	0.442	2,412
90%	61.3%	0.214	0.437	2,314
87%	78.5%	0.216	0.43	1,443

4.2. Validation of Model Accuracy and Training Time

The performance metrics used to evaluate the model are RMSE (Root Mean Squared Error) and NDCG (Normalized Discounted Cumulative Gain) based on the K value in the Top-K recommendations. RMSE measures the difference between the predicted and actual values, with a value closer to 0 indicating higher prediction accuracy. NDCG is a normalized measure of the cumulative relevance of the recommended items in order, with values closer to 1 indicating better model performance. The corresponding performance metrics are depicted in Fig. 2 and Fig. 3.

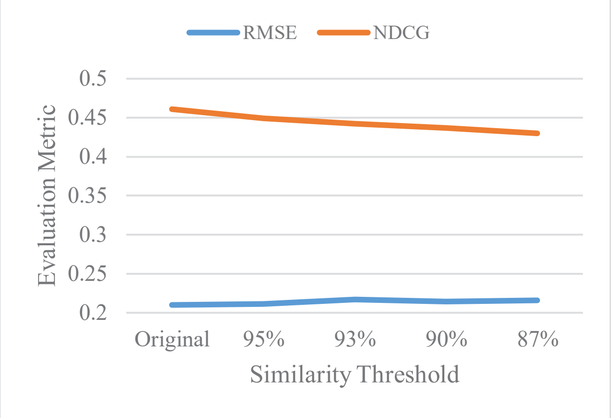

Fig. 2. RMSE & NDCG performance metrics graph by reduction rate

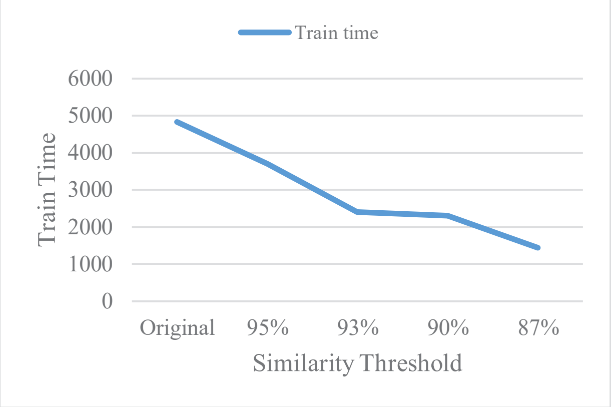

Fig. 3. Lightweight reduction impact on training time

The experimental results showed that, compared to the original data, the RMSE difference in the reduced data was within 0.06, indicating minimal performance degradation. For NDCG, although the performance slightly decreased as the data reduction rate increased, the difference remained within 4%, which is relatively minor. On the other hand, the training time was reduced by up to 70%, demonstrating significant efficiency in resource consumption. However, as the data reduction rate increased, there was a tendency for performance metrics to decline, highlighting the importance of selecting an appropriate similarity threshold. By doing so, it is possible to achieve effective data reduction while maintaining optimal performance.

4.3. Comparison of Different Data Techniques

The results of the comparison of different data reduction techniques are shown in Fig. 4. To validate the effectiveness of the data reduction method proposed in this paper, it was compared with the "Most Recent" and "Least Favorite" strategies suggested in previous studies [11]. The experiments were conducted using the MovieLens-1M dataset, with the data reduction rate set uniformly at approximately 61% across all methods. This rate is higher than the 20-50% reduction rate used in Yazdi's experiments.

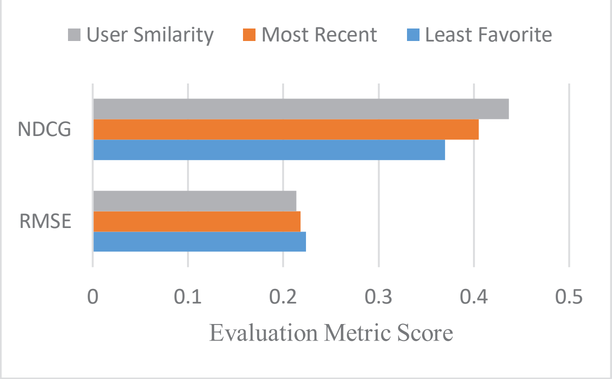

Fig 4. Recommendation performance comparison graph

The comparison results, shown in Table V, demonstrated that the proposed method achieved 36% better performance in NDCG and slightly better results in RMSE. This indicates that the lightweight approach in this study is particularly effective in large-scale data reduction scenarios.

TABLE V. COMPARISON OF RECOMMENDATION PERFORMANCE

MovieLens-1M Data			
Reduction Module	***Reduction Rate(%)***	***RMSE***	***NDCG***
User Similarity	61.3%	0.214	0.437
Most Recent	61.6%	0.218	0.405
Least Favorite	61.6%	0.224	0.37

V. CONCLUSION

In this study, we proposed a lightweight algorithm to address the issues of overfitting and resource consumption in recommendation systems caused by large-scale data. This algorithm implements data reduction by removing redundant users using matrix factorization and cosine similarity. The experimental results showed that the algorithm could reduce data by up to 78.5% with a performance degradation of only up to 4%. Additionally, when compared to other reduction strategies, our approach demonstrated superior performance metrics even at a 60% data reduction rate. These findings indicate that it is possible to reduce training time and resource consumption while minimizing the impact on recommendation performance. The primary contribution of this study is the introduction of a new approach that enhances model efficiency and mitigates overfitting in recommendation systems by removing redundant similar users.

ACKNOWLEDGMENT

This work was supported by an Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No.2022-000218).

REFERENCES

[1] C. C. Aggarwal, Recommender systems, vol. 1. Cham: Springer International Publishing, 2016.

[2] A. Shetty, A. Shetye, P. Shukla, A. Singh, and S. Vhatkar, "A collaborative filtering-based recommender systems approach for multifarious applications," Journal of Electrical Systems, vol. 20, no. 4s, pp. 478-485, 2024.

[3] A. J. Bansal, H. H. Malik, and S. K. Dash, “Redundancies in data and their effect on the evaluation of recommendation systems: A case study on the Amazon reviews datasets,” in Proc. 2017 SIAM Int. Conf. Data Mining, Houston, TX, 2017, pp. 567-575.

[4] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S. Chua, "Neural collaborative filtering," in Proceedings of the 26th International Conference on World Wide Web, April 2017, pp. 173-182.

[5] A. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Recurrent collaborative filtering," arXiv:1608.07400, 2016. [Online]. Available: https://arxiv.org/abs/1608.07400.

[6] X. Wang, X. He, Y. Cao, M. Liu, and T. S. Chua, “Neural graph collaborative filtering” in Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, Paris, France, 2019, pp. 165-174.

[7] D. Basaran, E. Ntoutsi, A. Zimek, “Redundancies in data and their effect on the evaluation of recommendation systems: A case study on the amazon reviews datasets,” Proceedings of the 2017 SIAM international conference on data mining, 2017, pp. 390-398

[8] A. J. Biega, P. Potash, H. Daumé, F. Diaz, and M. Finck, "Operationalizing the legal principle of data minimization for personalization," in Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, July 2020, pp. 399-408.

[9] Y. J. Lee, Y. H. Park, J. H. Um, Y. W. Jo, and J. Y. Choi, "User Similarity Analysis based Interaction Data Compression Method for Improving the Quality of Learning Data in a Recommendation System," Proceedings of the 2024 Spring Conference of the Korean Society of Internet Information (KSII), unpublished, April 2024.

[10] H. Zhang, F. Luo, J. Wu, X. He, “LightFR: Lightweight federated recommendation with privacy-preserving matrix factorization," in ACM Transactions on Information Systems, vol. 41, March 2023.

[11] X. Niu, R. Rahman, X. Wu, Z. Fu, D. Xu, and R. Qiu, "Leveraging uncertainty quantification for reducing data for recommender systems," in 2023 IEEE International Conference on Big Data (BigData), December 2023, pp. 352-359, IEEE.

[12] H. Ahmadian Yazdi, S. J. Seyyed Mahdavi Chabok, and M. KheirAbadi, "Effective data reduction for time-aware recommender systems," Control and Optimization in Applied Mathematics, vol. 8, no. 1, pp. 33-53, 2023.

[13] MovieLens Dataset. "MovieLens 1M dataset." [Online]. Available: https://grouplens.org/datasets/movielens/1m/. Accessed: August 21, 2024.