STAT542 FINAL PROJECT REPORT
UNSUPERVISED LEARNING: MATRIX COMPLETION

Alice Mei
University of Illinois at Urbana-Champaign
Illinois, 61801

Tomy Fan
University of Illinois at Urbana-Champaign
Illinois, 61801

Jui-Yu Lin
University of Illinois at Urbana-Champaign
Illinois, 61801

Yu-Ching Liao
University of Illinois at Urbana-Champaign
Illinois, 61801

Dimitrios Mandravelis
University of Illinois at Urbana-Champaign
Illinois, 61801

Nattanai Na Songkhla
University of Illinois at Urbana-Champaign
Illinois, 61801

Supanut Wanchai
University of Illinois at Urbana-Champaign
Illinois, 61801

Bo-Yang Wang
University of Illinois at Urbana-Champaign
Illinois, 61801

May 12, 2023

ABSTRACT

The aim of this project is to explore unsupervised learning techniques for matrix completion. We compared several models, including KNN, user-based collaborative filtering, item-based collaborative filtering, SVD, and KNN+SVD, to impute missing values in a rating matrix. We also collected a new dataset that has similar characteristics to the provided dataset to enhance the validity of the results. Our results show that the IBCF using KNN model achieved the lowest RMSE of 0.6337, outperforming the other models. We also discussed some difficulties encountered in the project, such as user rating bias and dataset limitations. Finally, we have provided a Github implementation of our recommendation system to facilitate future work in this area. The Github link to our code can be found at `https://rb.gy/la6kj`.

***K*eywords** Unsupervised Learning · Matrix Completion · Collaborative Filtering · SVD · Recommender Systems

1 Problem Formulation

The Netflix Prize was a well-known competition that challenged teams to improve the accuracy of Netflix's movie recommendation algorithm. The competition demonstrated the effectiveness of collaborative filtering, a technique that predicts user preferences by comparing them with similar users. In this project, we aimed to apply the same collaborative filtering approach to the restaurant industry in the Urbana-Champaign area. By collecting a dataset of ratings (1 to 5) from survey participants, we aimed to develop a recommendation system for restaurant-goers in the area. The project's goal is to utilize the collaborative filtering algorithm to discover new restaurants that users may enjoy based on their past ratings and the ratings of similar users. However, one of the main issues we encountered was the sparsity of the data and the prevalence of neutral ratings (3 out of 5). To address this, we collected our own dataset, which maximized its similarity with the provided dataset, and applied various similarity metrics and matrix completion techniques, including KNN, SVD, and their combination to improve the accuracy of the recommendation system. The project's ultimate objective is to provide a valuable tool for both residents and visitors to the Urbana-Champaign area, aiding them in discovering new and exciting dining options.

2 Dataset

Our unsupervised learning project was based on a dataset collected from roughly 40 students at UIUC, Feedback.csv. However, the dataset provided to us was not suitable for evaluating our algorithm due to heavy missing data. We considered the Netflix challenge dataset, which is a good choice for projects related to user preferences, but found that most of the ratings in Feedback.csv were 3, making it unsuitable for our case.

To address the issue of heavily missing data, we collected a new dataset by developing an identical survey with the same 15 restaurants and rating scale (1 to 5). The new dataset has dimensions of 50 customers x 15 restaurants and a sparsity of 50% in the training set. We also shuffled randomly to the column and rows to make it more similar to the Feedback.csv dataset. Our motivation for collecting this dataset was to ensure that it reflected the original dataset while addressing the issue of missing data. The new dataset was also collected from 50 students at UIUC and should provide a more accurate representation of user preferences in the Urbana-Champaign area. New survey respondents are asked to fill in 3 for restaurants they haven't been to, in order to mimic the behaviors of the original respondents.

3 Methods

We collected a dataset from 50 students of UIUC with the same 15 restaurants and rating scale (1-5) as the original dataset. The training set had a 50% sparsity, which we used to evaluate different methods. We implemented K-Nearest Neighbor (KNN), Singular Value Decomposition (SVD), and KNN-SVD hybrid methods for matrix completion.

3.1 K-Nearest Neighbor (KNN)

KNN is a machine learning algorithm used for classification or regression problems. It works by computing the similarity between data points and selecting the k-nearest neighbors to make predictions. In this project, we used KNN for collaborative filtering, a recommendation system that predicts user preferences by comparing them with similar users. The first step in KNN is to compute the similarity between data points. We used three similarity measurement criteria: *Pearson correlation*, *Cosine similarity*, and *Euclidean distance*, with formulas as shown below. Pearson correlation measures the linear correlation between two variables, while cosine similarity measures the cosine of the angle between two vectors. Euclidean distance measures the distance between two points in space. The second step in KNN is to define the k-nearest neighbors. We selected the k-nearest neighbors based on their similarity to the target user/item, where k is a parameter that we optimized through cross-validation. Once we have the k-nearest neighbors, we can predict the missing ratings by taking the average rating of users in the k nearest neighborhood.

$$m_{Pearson}(X,Y) = \frac{1}{n-1}\sum_{l=1}^{n}\left(\frac{x_l-\bar{x}}{s_x}\right)\left(\frac{y_l-\bar{y}}{s_y}\right), \text{ Similairity } s = \frac{m_{Pearson}+1}{2} \tag{1}$$

$$m_{Cosine}(X,Y) = \frac{X\,Y}{||X||\,||Y||}, \text{ Similairity } s = \frac{m_{Cosine}+1}{2} \tag{2}$$

$$d(X,Y) = \sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}, \text{ Similairity } s = \frac{1}{d+1} \tag{3}$$

User-Based Collaborative Filtering (UBCF) using KNN

User-based collaborative filtering is a type of collaborative filtering where we find a neighborhood of similar users and use their ratings to make predictions. The methodology for user-based collaborative filtering involves finding a neighborhood of similar users, selecting the k-nearest neighbors, and using their ratings to predict the missing ratings. UBCF with KNN can work well when there is a large number of users and items, but it can suffer from the sparsity problem when there are a lot of missing ratings. To find the neighborhood of similar users, we computed the similarity between users using the three similarity measures mentioned above. We then selected the k-nearest neighbors based on their similarity to the target user, where k is a parameter that we optimized through cross-validation. Finally, we predicted the missing ratings by taking the average rating of users in the k nearest neighborhood.

Item-Based Collaborative Filtering (IBCF) using KNN
Item-based collaborative filtering is a type of collaborative filtering where we find a neighborhood of similar items and use their ratings to make predictions. IBCF generally handles the sparsity problem better than UBCF and it has been widely adopted across all the Web giants. The methodology for item-based collaborative filtering involves finding a neighborhood of similar items, selecting the k-nearest neighbors, and using their ratings to predict the missing ratings. To find the neighborhood of similar items, we computed the similarity between items using the three similarity measures mentioned above. We then selected the k-nearest neighbors based on their similarity to the target item, where k is a parameter that we optimized through cross-validation. Finally, we predicted the missing ratings by taking the weighted-average rating of items in the k nearest neighborhood, skipping missing ratings of active user, where the weights are the similarities between items.

3.2 Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) is a matrix factorization technique that decomposes a matrix into three matrices: U, Σ, and V, where U and V are orthogonal matrices and Σ is a diagonal matrix with singular values on the diagonal in decreasing order. SVD is commonly used for dimensionality reduction, data compression and also in recommendation systems. In fact, the winning algorithm of the Netflix Prize is a SVD-blend algorithm. In this project, we used SVD to fill in the missing values in our dataset. We used iterative SVD, a technique that involves initializing the missing values in the matrix, truncating the matrix and iteratively applying SVD to the matrix until convergence. We experimented with different initialization strategies, including initializing the missing values with 3, row mean, and column mean.

$$A = U\Sigma V^T \tag{4}$$

3.3 KNN+SVD

From the concept of SVD, we know that the initialization of SVD plays a crucial role to the result. Therefore, we decided to use KNN as our initialization. KNN+SVD is a hybrid approach that combines KNN and SVD. In this approach, we first use KNN to predict the missing values in the matrix and then apply SVD to refine the predictions. We experimented with two versions of KNN+SVD, one using user-based collaborative filtering and the other using item-based collaborative filtering. We optimized the parameters for both KNN and SVD through cross-validation.

4 Results

We compare and evaluate the performance of different collaborative filtering techniques, including k-Nearest Neighbors (KNN), Singular Value Decomposition (SVD), and their combination, KNN+SVD, in predicting user ratings for restaurants. Two collaborative filtering methods were used, including user-based collaborative filtering (UBCF) and item-based collaborative filtering (IBCF). The performance of each technique was measured based on the root mean square error (RMSE) of the predicted ratings compared to the actual ratings. The results of this study suggest that the IBCF using KNN can provide the best performance in predicting user ratings for restaurants. However, the selection of the optimal technique highly depends on the type of data and the application domain.

4.1 K-Nearest Neighbor (KNN)

For KNN, three similarity measures were used to calculate the similarity of the data points, including Pearson correlation, cosine similarity, and Euclidean distance. The optimal number of neighbors (k) for each similarity measure was determined based on the lowest RMSE value.

UBCF using KNN
For UBCF, the performance of the technique was evaluated by determining the optimal number of neighbors (k) for each similarity measure, which resulted in the lowest RMSE value. The results showed that the Euclidean distance had the lowest RMSE value (0.6415) at k=22, followed by Pearson correlation with RMSE value of 0.6640 at k=11, and cosine similarity with RMSE value of 0.6442 at k=26.

IBCF using KNN
For IBCF, the performance of the technique was evaluated by determining the optimal number of neighbors (k) for each similarity measure, which resulted in the lowest RMSE value. The results showed that the Euclidean distance had the lowest RMSE value (0.6337) at k=11, followed by cosine similarity with RMSE value of 0.6338 at k=8, and Pearson correlation with RMSE value of 0.6829 at k=7.

Criteria	Optimal K	RMSE
Pearson	11	0.6640
Cosine	26	0.6442
Euclidean	22	0.6415

Table 1: Results of UBCF using KNN

Criteria	Optimal K	RMSE
Pearson	7	0.6829
Cosine	8	0.6338
Euclidean	11	0.6337

Table 2: Results of IBCF using KNN

4.2 Singular Value Decomposition (SVD)

For SVD, three different approaches were used to initialize NaN values in the matrix A. The results showed that initializing NaN values with column mean had the lowest RMSE value (0.6417), followed by initializing NaN values with row mean with RMSE value of 0.6706 and initializing NaN values with 3 with RMSE value of 0.7236.

Substitute NaN with	RMSE
3	0.7236
Row mean	0.6706
Column mean	0.6417

Table 3: Results of SVD

4.3 KNN+SVD

For KNN+SVD, the UBCF and IBCF methods were used to predict the user ratings, and the results showed that IBCF with KNN(k=11)+SVD had the lowest RMSE value (0.6412) compared to other techniques, including UBCF using KNN+SVD (RMSE=0.6516), UBCF using KNN (RMSE=0.6415), and SVD (RMSE=0.6417).

KNN	RMSE
UBCF (22NN Euclidean Dist) + SVD	0.6516
IBCF (11NN Euclidean Dist) + SVD	0.6412

Table 4: Results of KNN+SVD

5 Discussions

The purpose of this project was to investigate the effectiveness of unsupervised learning techniques for matrix completion, specifically focusing on KNN, SVD, and their combinations. The dataset used in this project was Feedback.csv, which contained ratings of restaurants by students. However, the dataset had limitations, such as the majority of ratings being 3, making it unreliable for analysis. To address this issue, we collected our own dataset from UIUC students that included the same restaurants and rating scale. Two methods of collaborative filtering were used for this project: user-based and item-based. User-based CF was used to find a neighborhood of similar users and predict missing ratings by taking the average rating of users in the k nearest neighborhood. Item-based CF was used to find a neighborhood of similar items and predict missing ratings by taking the weighted-average rating of items in the k nearest neighborhood, where the weights were based on similarity. Other than two collaborative filtering mentioned above, we also explored non-negative matrix factorization (NMF) to decompose the matrix into two smaller matrices. This is a popular dimension reduction method widely used in data imputation and recommender systems. However, we found that collaborative filtering and SVD are more suitable for our project, so we decided to investigate more on those two methods in the end.

One of the challenges we encountered was the user rating bias in the dataset. Some users tended to give higher ratings while others tended to give lower ratings, which could affect the accuracy of the CF methods. To address this issue, we centered the rows of user-item rating by doing normalization. We also implemented SVD, which decomposed the matrix into three matrices, U, Σ, and V, and iteratively updated the NaN values until convergence. The results showed that initializing NaN values with column mean achieved the lowest RMSE of 0.6417. We also combined KNN and SVD, where the results from KNN were used as the initial matrix for SVD. The best results were obtained using item-based CF with 11NN Euclidean distance, achieving an RMSE of 0.6412.

In conclusion, this project showed that unsupervised learning techniques such as KNN, SVD, and their combinations could effectively complete a matrix with missing values. However, the choice of algorithm and similarity measurement could significantly affect the accuracy of the results. Furthermore, the quality and bias of the dataset could also impact the performance of the methods. Overall, this project contributes to the growing field of unsupervised learning and matrix completion.

6 Conclusions

In conclusion, we explored several unsupervised learning methods for matrix completion using a dataset of restaurant ratings. We compared the results of three similarity measurements for both user-based and item-based collaborative filtering, as well as the iterative SVD approach and a combination of KNN and SVD. Our results showed that the best method was the item-based collaborative filtering with KNN, which achieved an RMSE of 0.6337. However, the user-based collaborative filtering with KNN and the SVD approach were also competitive, with RMSEs of 0.6415 and 0.6417, respectively. We also identified two difficulties during the process: the dataset had a bias towards ratings of 3, and there was user rating bias that needed to be addressed through normalization. Overall, these methods and insights can be applied to other similar problems and datasets, and further research can be done to improve the performance and accuracy of matrix completion techniques.

7 Contributions

Names	Contributions
Yu-Ching Liao	Discuss in the meeting, coding (overall code setup, explore and implement all methods we have, result visualization, and parameter tuning for fetching best K), review the report, and research on IBCF KNN Model.
Tomy Fan	Discuss in the meeting, coding on SVD part and parameter tuning, research on SVD model, and review the report.
Jui-Yu Lin	Discuss in the meeting, research on NMF algorithm with Alice, prepare slides for presentation, give presentation in class, and review the report.
Alice Mei	Discuss in the meeting, research on NMF algorithm with Jui-Yu, slides proofread, and write the report.
Bo-Yang Wang	Discuss in the meeting, research on KNN methods, revise and review the report.
Dimitrios Mandravelis	Research on SVD model, review the report.
Nattanai Na Songkhla	Discuss in the meeting, research on KNN, coding on UBCF part and parameter tuning, and review the report.
Supanut Wanchai	Discuss in the meeting, research on KNN, coding on UBCF part and parameter tuning, prepare slides for presentation, give presentation in class, and review the report.

8 References

1. Hahsler, M., Grün, B., & Hornik, K. (n.d.). Building and Evaluating Recommender Systems with R using Package recommenderlab. Retrieved from `https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf`
2. Wikipedia contributors. (n.d.). K-nearest neighbors algorithm. In Wikipedia. Retrieved May 9, 2023, from `https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm`
3. Wikipedia contributors. (n.d.). Singular value decomposition. In Wikipedia. Retrieved May 1, 2023, from `https://en.wikipedia.org/wiki/Singular_value_decomposition`
4. GeeksforGeeks. (n.d.). Singular Value Decomposition (SVD) - GeeksforGeeks. Retrieved May 1, 2023, from `https://www.geeksforgeeks.org/singular-value-decomposition-svd/`
5. Wikipedia contributors. (n.d.). Non-negative matrix factorization. In Wikipedia. Retrieved May 1, 2023, from `https://en.wikipedia.org/wiki/Non-negative_matrix_factorization`