A Comparison of Item-Based and User-Based Collaborative Filtering in Recommendation Systems.

Abstract This report discusses item-based collaborative filtering and user-based collaborative filtering in the context of recommendation systems. The report compares and contrasts the two approaches and discusses the advantages and limitations of using Singular Value Decomposition (SVD) in collaborative filtering. Collaborative filtering is used to predict user preferences based on past interactions with items or other users. Item-based collaborative filtering computes similarity between items and user ratings to predict a user's rating for an item, while user-based collaborative filtering aggregates similar users' ratings to predict a user's rating for an item. The main challenge in collaborative filtering is the sparsity of the user-item matrix and the cold-start problem, which can be addressed using SVD. This report explains the SimSVD algorithm, which uses SVD to compute the item-item similarity matrix.

1. Introduction

Collaborative filtering is a powerful and widely used technique in recommendation systems. It involves predicting user preferences based on their past interactions with items or other users. Item-based collaborative filtering and user-based collaborative filtering are two commonly used approaches within the collaborative filtering framework.

Item-based collaborative filtering relies on the similarities between items to make recommendations. It assumes that users who have shown a preference for a particular item are likely to have similar preferences for other similar items. In item-based collaborative filtering, a user's rating for an item is predicted by combining their ratings for similar items.Another approach is user-based collaborative filtering, which focuses on finding similar users and using their ratings to make recommendations. The underlying assumption is that users who have similar preferences in the past are likely to have similar preferences in the future. In user-based collaborative filtering, a user's rating for an item is predicted by aggregating the ratings of similar users who have rated the same item.

In this report, we will discuss item-based collaborative filtering with SVD and user-based collaborative filtering in detail. We will compare and contrast the two approaches, and provide insights into when each approach might be more suitable. We will also discuss the advantages and limitations of using SVD in collaborative filtering.

1.1. Background

Collaborative filtering is a popular technique in recommendation systems that predicts user preferences based on their past interactions with items or other users. Item-based and user-based collaborative filtering are two commonly used approaches in collaborative filtering. Item-based collaborative filtering computes the similarity between items based on user ratings and then uses this similarity to predict a user's rating for an item. User-based collaborative filtering finds users who have similar preferences to the target user and aggregates their ratings to make a prediction for a new item. Both approaches face challenges, such as the sparsity of the user-item matrix and the cold-start problem, but Singular Value Decomposition (SVD) has been used to address these issues by decomposing the user-item matrix into lower-dimensional representations. Hybrid approaches that combine item-based and user-based collaborative filtering have also been proposed.

2. Problem Definition

The problem addressed by item-based collaborative filtering (with SVD) and user-based collaborative filtering is the task of predicting user preferences for items in recommendation systems. The main challenge in this problem is the sparsity of the user-item matrix, which can lead to a lack of data for similar items or users. Another challenge is the cold-start problem, which arises when there is not enough data about a new user or item. Item-based collaborative filtering addresses these challenges by computing the similarity between items based on user ratings and then using this similarity to predict a user's rating for an item. SVD is used to address the sparsity problem by decomposing the user-item matrix into lower-dimensional representations. User-based collaborative filtering addresses these challenges by finding users who have similar preferences to the target user and aggregating their ratings to make a prediction for a new item.

3. Methodology and Algorithms

Following are the prominent algorithms that were implemented in this lab.

Collaborative Filtering

Collaborative filtering is a widely used technique in recommender systems, particularly in the domain of personalised recommendations. The basic idea behind collaborative filtering is to exploit the patterns of user behaviour and item properties to make predictions about new interactions. In other words, collaborative filtering aims to identify users who are similar to each other based on their interaction history, and then use this similarity to recommend items that similar users have liked. There are two main approaches to collaborative filtering: user-based and item-based.

Item Based Collaborative Filtering

In item-based collaborative filtering, we would start by constructing an item-by-user matrix where each row represents an item and each column represents a user. The cells of the matrix contain the rating that each user has given to each item. We can then compute the similarity between items using a similarity metric such as cosine similarity. Once we have computed the similarity between items, we can use this similarity to make recommendations by identifying items that are similar to the ones that the user has already rated highly.

Singular Value Decomposition (SVD) and Matrix Factorisation:

Collaborative filtering with Singular Value Decomposition (SVD) and regression is a popular method for recommendation systems. In this method, the goal is to find latent factors that represent the underlying preferences of users and items. The MovieLens dataset is a popular benchmark dataset for evaluating recommendation systems and will be used in this explanation.The next step is to perform matrix factorisation using SVD. Matrix factorisation is a way to represent a matrix in terms of its latent factors. The idea is to approximate the original matrix with two lower-rank matrices: U and V, where U represents the user preferences and V represents the item features.

$$r(u, i) = U[u, :] \cdot V[:, i]$$

Algorithm Steps:

The SimSVD algorithm performs the following steps to learn the item-item similarity matrix:

Process the dataset: The algorithm takes a dataset as input and reads the user-item matrix.

Initialise the similarity matrix: The algorithm initialises the item-item similarity matrix with zeros.

Compute the SVD of the user-item matrix: The algorithm computes the Singular Value Decomposition (SVD) of the user-item matrix to extract the latent factors.

Reduce the dimensionality of the matrix: Reduces the dimensionality of the matrix by selecting the top K latent factors.

Compute the reduced user-item matrix: The algorithm computes the reduced user-item matrix by multiplying the selected latent factors with the transpose of the user-item matrix.

Compute the item-item similarity matrix: Next, computes the item-item similarity matrix by taking the dot product of the reduced user-item matrix and its transpose.

Set the values of the similarity matrix: The algorithm sets the values of the item-item similarity matrix based on the values of the dot product.

Recommend items: The algorithm recommends items to users based on the item-item similarity matrix.

User Based Collaborative Filtering

User-based collaborative filtering is a popular approach to recommend items to users based on their past interactions with items and the behavior of similar users. It leverages the similarity between user profiles to identify users with similar preferences and recommends items that are highly rated by those similar users. User-based collaborative filtering is easy to understand and implement, and it works well when there is enough data on user interactions with items. However, it suffers from the cold-start problem, where new users or items have little to no interaction data, making it difficult to recommend items.

In the lab, It’s technical details are as follows

Mean Squared Difference Similarity Metric Algorithm:

This algorithm calculates the similarity between two user profiles based on the mean squared difference between their ratings.

$$sim(u_i, u_j) = \frac{1}{n} \sum_{k=1}^{n} (r_{i,k} - r_{j,k})^2$$

Threshold Neighbourhood Algorithm:

This algorithm forms a neighborhood of similar users for a given user by selecting only those users whose similarity scores are above a given threshold.

$$N(u_i) = u_j \in U \mid sim(u_i, u_j) \geq \theta$$

Deviation From User Mean Predictor Algorithm:

This algorithm predicts the rating of an item for a given user by computing the deviation of the user's rating from their mean rating, and then adding this deviation to the mean rating of all other users who have rated the item.

$$\hat{r}ui = \bar{r}u + \frac{\sum j \in N_i(u) sim(u,i)(ruj - \bar{r}j)}{\sum j \in N_i(u) \,|\, sim(u,j) \,|}$$

4. Evaluation

4.1. Research questions addressed

RQ1 How does the SimSVD algorithm perform in learning the similarity using the regression method that depends on the SVD of the interaction data matrix?

RQ2 What are the limitations of the algorithms implemented in the lab for generating item similarity matrices, and how can they be addressed to improve accuracy and performance?

RQ3 How do item-based and user-based collaborative filtering compare in terms of accuracy and efficiency in recommendation systems?

4.2. Datasets

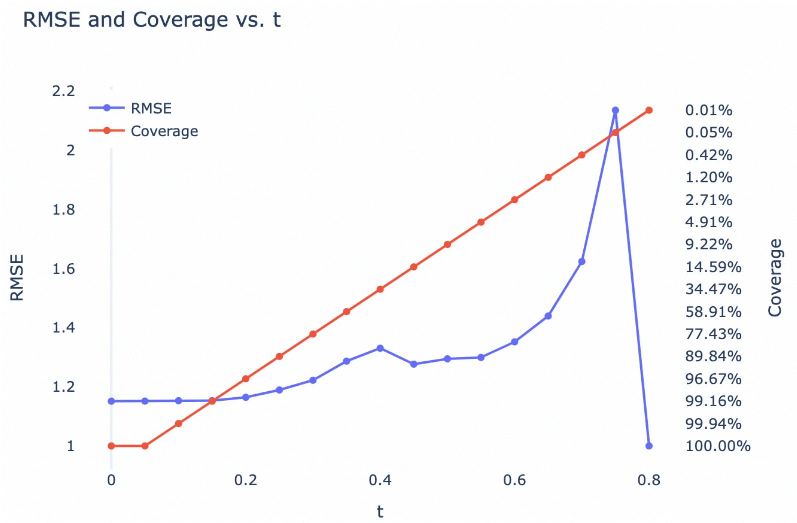
The MovieLens dataset ML20M is a widely recognised benchmark dataset that is extensively utilised for evaluating and analysing the performance of recommender systems. This dataset consists of 20 million ratings assigned to 27,000 movies by 138,000 users, which were collected by the MovieLens website. The dataset encompasses additional details regarding the movies, including genre, release year, and tags.

The diverse user and item space of the ML20M dataset, along with its substantial size, make it an optimal choice for building and assessing recommendation algorithms. This dataset allows for the assessment of both personalised and non-personalised recommendation techniques. The inclusion of supplementary files, such as movie metadata, user demographic data, and tag data, further enhances the potential for improving recommendation quality and executing sophisticated analyses.

4.3. Results and Discussion

Hyper Parameter Tuning in User based and Item based collaborative Filtering

We have tested with various threshold values to find the best neighbourhood threshold. Below graphs shows the result of UB collaborative filtering for different neighbourhood threshold values



We can see as the Neighbourhood threshold is increased, the coverage also increases. RMSE Error Plateau at first but it increases when we increase the Neighbourhood threshold, this is because It adds too much noise and thus performance of our algorithm is affected

Below Table shows the RMSE, Coverage and Threshold values used to tune network parameters.

t	RMSE	Coverage
0.000000	1.151493	100.00%
0.050000	1.151719	100.00%
0.100000	1.153079	99.94%
0.150000	1.153367	99.16%
0.200000	1.164595	96.67%

Different Metrics with their Loss and Coverage

Following Table shows the RMSE values for different metrics that we used to test our UBCF algorithm

Metric	RMSE	Coverage
Cosine	1.108856	99.93%
Pearson	3.709268	100.00%
Pearson Sig	3.709268	100.00%
MSD	1.218550	89.53%

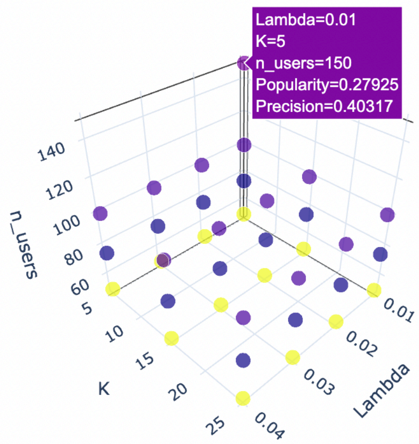
It shows that Cosine Similarity has Outperformed all other metrics and thus it is a good choice.

Hyper Parameter tuning and Results for IB Collaborative Filtering

We have tested with different values

- Number of users
- Latent space K
- Lambda
- Mean

Below 3D Scatter plot shows the precision, recall and popularity for different parameter values. It shows that As the number of users are increased precision is effected.



Also, The table Below show metric results for different parameter values

N	Lambda	K	n_users	Popularity	Precision	Recall
30	0.01	5	50	0.39	0.44	0.55
30	0.01	15	50	0.41	0.46	0.55
30	0.01	25	50	0.31	0.49	0.57
30	0.02	5	50	0.39	0.41	0.59
60	0.02	15	50	0.41	0.44	0.58

Item based and User based are very robust algorithms and the above results indicate that both have achieved excellent results on a very huge and sparse datasets.

Precision, Recall and Popularity were used to measure the performance of IB collaborative filtering. Whereas, RMSE and coverage are used for UB collaborative filtering. Comparing the results after hyperparameter tuning show that either one of them is robust enough to use for recommending content to users.

5. Conclusions

collaborative filtering is an important technique used in recommendation systems. Item-based and user-based collaborative filtering are two commonly used approaches within the collaborative filtering framework. Both approaches face challenges such as the sparsity of the user-item matrix and the cold-start problem, but Singular Value Decomposition (SVD) has been used to address these issues by decomposing the user-item matrix into lower-dimensional representations. Both approaches are have shown efficient and effective results for movielens dataset. And thus, These both can be used to recommend personalised content to users based on their history.

References

1. Harper, F. M., & Konstan, J. A. (2015). The MovieLens datasets: history and context. ACM Transactions on Interactive Intelligent Systems (TiiS), 5(4), 1-19.
2. Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8), 30-37.
3. Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web (pp. 285-295). ACM.