GROUP RECOMMENDATION SYSTEM USING MATRIX FACTORIZATION

Submitted in partial fulfillment of the requirements of the degree of

Master of Computer Applications

BY

Rajesh

Roll No. 177924

Under the supervision of

Dr. Venkateswara Rao Kagita

Department of Computer Science and Engineering

National Institute of Technology, Warangal

2019-2020

Approval Sheet

This Project Work titled **Group Recommendation System using Matrix Factorization** by **Rajesh** is approved for the degree of Master of Computer Applications.

Examiners

____

____

Supervisor (s)

____

Dr. Venkateswara Rao Kagita
Asst. Professor, CSE
NIT Warangal

Chairman

____

Dr. P. Radha Krishna
Head of Department, CSE
NIT Warangal

Date: ________________
Place: ________________

Declaration

I declare that this written submission represents my ideas in my own words and where others ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

____

Signature

Rajesh

Roll No.: 177924

Date: ___________

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL-506004

CERTIFICATE

This is to certify that the project entitled "**Group Recommendation System using Matrix Factorisation**" is a bonafide work carried out by Rajesh in partial fulfillment of the requirements for the award of the degree of Master of Computer Applications and submitted to the Department of Computer Science and Engineering, National Institute of Technology, Warangal.

Dr. Venkateswara Rao Kagita
Project Guide
Asst. Professor, Department of CSE
NIT Warangal-506004

Dr. P. Radha Krishna
Head of the Department
Professor, Department of CSE
NIT Warangal-506004

Acknowledgements

I consider it as a great privilege to express my deep gratitude to many respected personalities who guided, inspired, and helped me in the successful completion of my project.

I would like to express my thanks with deepest gratitude to my project guide **Dr. Venkateswara Rao Kagita,** Department of Computer Science and Engineering, National Institute of Technology Warangal, for his invaluable guidance and constant encouragement. He has been a constant source of inspiration and helped me at every stage.

I am also grateful to **Dr. P. Radha Krishna**, Head of the Department, Computer Science and Engineering, National Institute of Technology, Warangal for his moral support to carry out this project.

I am very thankful to the Project Evaluation Committee, for their strenuous efforts to evaluate our projects.

I wish to thank all the staff members in the department for their kind cooperation and support given throughout our project work. We are also thankful to all our friends who have given valuable suggestions and helped in all stages of the development of my project.

RAJESH (177924)

ABSTRACT

Group recommender systems are becoming very popular on the social web owing to their ability to provide a set of recommendations to a group of users. Literature has several group recommender systems that extend personal recommender systems to group recommendation. However, most of these approaches make recommendations to the individual group without considering other groups. In this work, we propose to learn recommendations for all the groups collaboratively from other groups and users. More precisely, We explore the group recommendations using matrix factorization (MF) based collaborative filtering (CF) in this work.

We propose original approaches to decompose the rating matrix into four matrices representing users, items, user groups, and item groups in latent factor space. We use these matrices to recommend items to a group of users. We analyze the proposed methods in three different scenarios: when the group size is small, medium, and large. We also compare the precision of the proposed methods with existing group recommendation systems using KNN based collaborative filtering and matrix factorization based collaborative filtering method. We analyze group movie ratings on MovieLens datasets. Our study demonstrates the performance of group recommender systems depending on the size of the group.

Contents

CHAPTER 1

INTRODUCTION

A recommendation system is a model used for information filtering where it tries to predict a user's preferences and provide suggestions based on these preferences. These systems have become increasingly popular nowadays and are widely used today in areas such as movies, music, books, videos, clothing, restaurants, food, places, and other utilities. These systems collect information about a user's preferences and behavior and then use it to improve their suggestions in the future.

Many companies are making use of recommendation systems to increase user interaction and enrich a user's shopping experience. However, in the real world, there are many activities participated by groups that consist of multiple users, for instance, recommending a movie to a group of friends. The users in the group may have different choices. Group recommendation aims to provide customized recommendations for groups consisting of two or more users.

1.1 MOTIVATION

Recommendation systems are becoming tools of choice to select the online information relevant to a given user. Recommendation systems can be broadly classified as Content-Based (CB), Collaborative Filtering (CF), and Hybrid Recommendation System. CF is the most popular approach to build a recommendation system and has been successfully employed in many applications. Collaborative filtering is a much-explored technique in the field of data mining and information retrieval. It analyzes user behavior to establish connections between users and items and make recommendations based on other users' opinions. CF techniques assume that customers who had similar past preferences will have similar choices in the future. Many e-commerce companies have already incorporated the recommendation system with their Services. Examples for such recommendation systems include product and book recommendation by amazon, movie recommendations by Netflix, and product advertisements shown by Google based on the search history.

Recommender systems aim to provide information items that are of potential interest to a user. There are specific scenarios in which recommending a set of items to a group of several users is more appropriate than providing several sets of recommendations to each user of the group e.g., recommending a vacation destination to a family or recommending a song to play in a coffee shop. Group recommender systems aim to provide a set of recommendations that satisfy all users' preferences in a group. Matrix factorization is the most popular and successful approach in personalized recommender systems. The key idea of matrix factorization based techniques is to learn low dimensional user latent factors (U) and item latent factors (V) to simultaneously approximate the observed entries under some loss measure and predict the unobserved entries. We accomplish prediction by determining U×V.

1.2 OBJECTIVE

In this work, we extend the concept matrix factorization to the group recommendation scenario. We propose a matrix factorization Model to decompose the rating matrix into four matrices that represent users, items, user groups, and item groups in latent factor space. We then use these matrices to recommend items or groups of items as packages to a user or a group of users.

CHAPTER 2

FOUNDATIONAL CONCEPTS AND RELATED WORK

In this chapter, we outline the concepts related to recommender systems and point out some major works which have come out in this area over the years. We also distinguish the proposed approach with the existing techniques in the literature review section.

2.1 FOUNDATIONAL CONCEPTS

Recommender systems are the programs that attempt to recommend the most suitable items (products or services) to particular users by predicting a user's interest in an item based on related information about the items, the users, and the interactions between items and users. Recommender systems aim to reduce information overload by retrieving the most relevant information and services from a massive amount of data, thereby providing personalized services.

Recommendation systems are broadly classified into the following three categories based on the kind of data exploitable.

- **Content-based filtering:** Content-based recommendation techniques recommend articles or commodities similar to items previously preferred by a specific user.
- **Collaborative filtering:** Collaborative filtering makes recommendations by learning from user-item historical interactions, either explicit (e.g., user's previous ratings) or implicit feedback (e.g., browsing history).
- **Hybrid recommender systems:** The hybrid model integrates two or more types of recommendation strategies.

Collaborative Filtering (CF) based techniques are the most popular and successful in e-commerce applications. In this thesis, we focus on CF-based approaches. Figure 1 represents the different types of techniques used for the recommendation system.

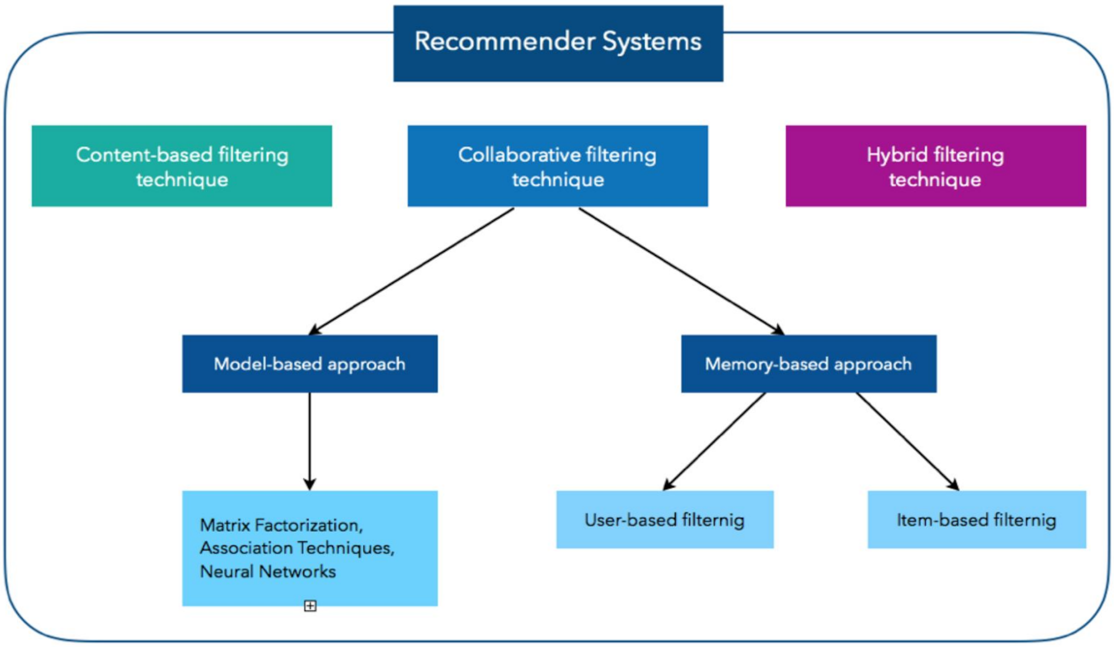

Figure 1: Recommendation Systems Techniques

2.1.1 COLLABORATIVE FILTERING

The exponential growth of data in E-Commerce demands more efficient and scalable algorithms and implementations. Collaborative filtering usually performs better than content-based techniques in recommendation accuracy, for large and complex data. Figure 2 shows how recommendations are generated for collaborative and content-based filtering.

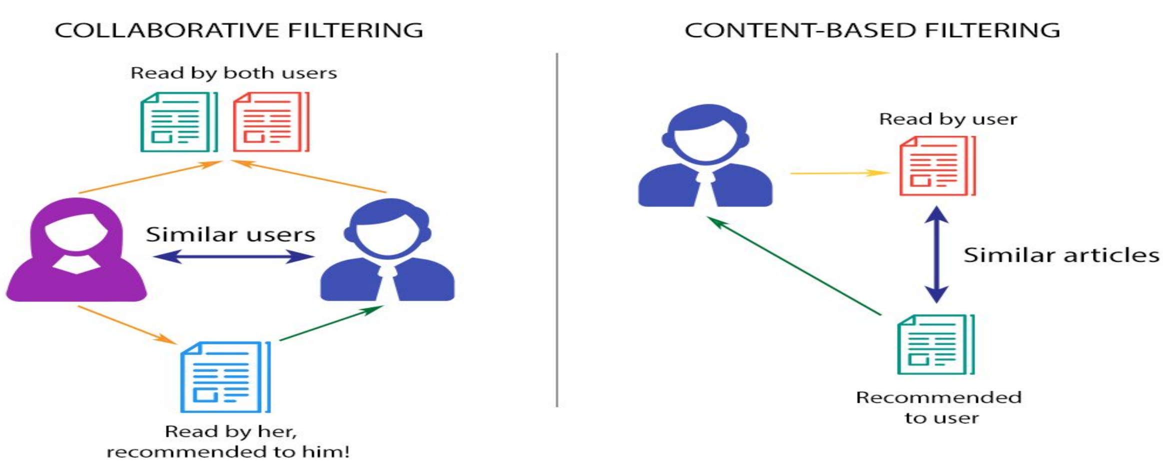

Figure 2: Collaborative Filtering vs Content-Based Filtering

The main advantage of collaborative approaches is that they require no information about users or items. Moreover, the more users interact with objects, the more new recommendations become accurate: for a fixed set of users and items, new interactions recorded over time bring new information and make the system more and more effective. Most of the CF-based algorithms exploit users' rating information. They work on the principle that the user who has the same preferences in the past will have similar future choices. They examine similarities in users' rating behavior for making recommendations. The idea is to find similar users to the target user and then recommend the items that are popular among them.

Types of Collaborative Filtering Techniques

Figure 3 shows the types of CF with their advantages and disadvantages. The CF techniques are broadly divided into two types :

1. Memory-based approach:

Memory-based collaborative filtering approaches are classified into user-item filtering and item-item filtering approaches.

- **User-item filtering:** User-item filtering finds users that are similar to the target user based on the similarity of ratings and recommends items those similar users liked.

- **Item-item filtering:** Item-item filtering finds items similar to a candidate item based on user rating information and uses the target user rating for those items to determine the relevance of a candidate item.

2. Model-Based Approach :

CF models are developed using machine learning algorithms to predict users' ratings of unrated items in this approach. There are many model-based CF algorithms, such as Clustering-based algorithms, Matrix Factorization based algorithms, and Deep Learning.

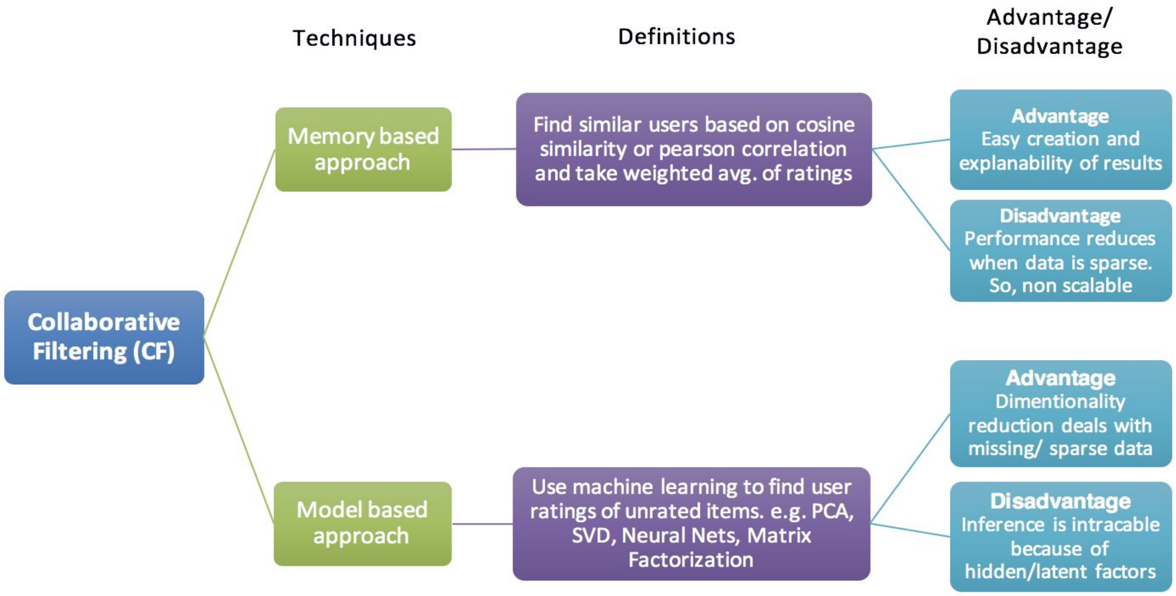

Figure 3: Types of Collaborative Filtering

2.1.2 MATRIX FACTORIZATION

Matrix factorization is the collaborative based filtering method that decomposes the original sparse matrix to low-dimensional matrices with latent factors/features. The intuition behind using matrix factorization is that some latent features determine how a user rates an item. For example, two users would give high ratings to a particular movie if they both like the movie's actors/actresses or if the film is an action movie, which is a genre preferred by both users. Hence, if we can discover these latent features, we could predict a rating for an individual user and item.

Matrix factorization came into limelight after the Netflix competition (2006) when Netflix announced prize money of $1 million to those who will improve its root mean square performance by 10%. Netflix provided a training data set of 100,480,507 ratings that 480,189 users gave to 17,770 movies. The matrix factorization approach was the most accurate approach to reduce the problem from high sparsity levels in recommendation system databases. Matrix factorization is beneficial for processing large recommendation system databases. The advantage of this over the standard nearest neighborhood is that even though two users haven't rated any same movies, it's still possible to find the similarity between them if they share similar underlying tastes, again latent features. Figure 4 represents the basic matrix factorization process where user-item rating matrix R is decomposed into two low-dimensional matrices P (user latent factors) and Q (item latent factors).

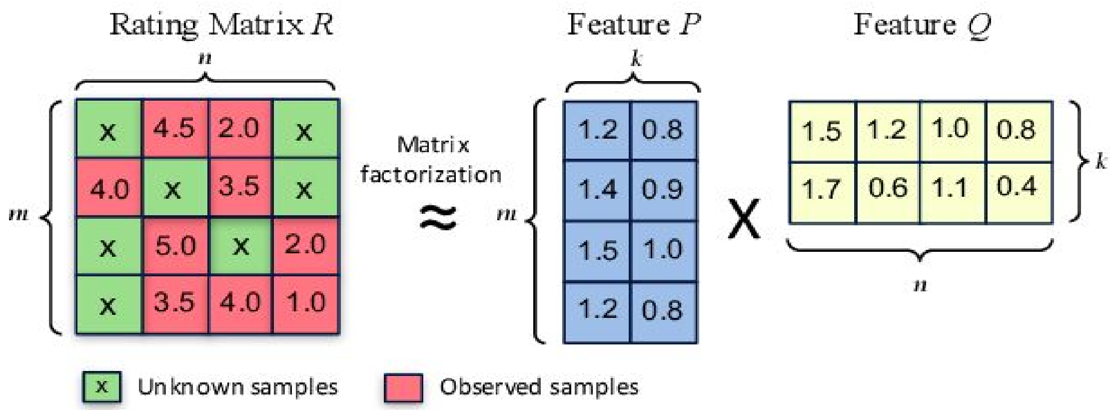

Figure 4: Example of Matrix Factorization Process

Let's quickly take a look at the matrix factorization formulation. Let *M* be the number of users, N be the number of items, and R be the user-item rating matrix of size M×N. Our task is to find two matrices $P^{M\times K}$ and $Q^{N\times K}$ such that their product approximates R.

$$R \approx P \times Q^T = \hat{R}$$

Rows of P represent users' latent factors, and rows of Q represent items' latent factors. To get the prediction for entry (i,j) in the rating matrix, we compute the dot product of the i^{th} row in matrix P and the j^{th} row in matrix Q as shown below.

$$\hat{r_{ij}} = p_i^T q_j = \sum_{k=1}^{K} p_{ik} q_{kj}$$

Hence, the objective is to obtain P and Q so that their product closely approximates R. That is the difference between the actual and the predicted ratings should be minimum for all the observed ratings. We achieve this by using a *gradient descent algorithm*. We initialize P and Q with some random values. We then determine how different their product is from R, and we minimize this difference iteratively.

The difference here is usually called the error between the estimated rating and the real rating and can be calculated using the following equation :

$$e_{ij}^2 = (r_{ij} - \hat{r}_{ij})^2 = (r_{ij} - \sum_{k=1}^{K} p_{ik} q_{kj})^2$$

To minimize the error, we have to know in which direction we have to modify the values of p_{ik} and q_{ki}. In other words, we need to know the gradient at the current values, and therefore we differentiate the above equation with respect to these two variables separately.

$$p'_{ik} = p_{ik} + \alpha \frac{\partial}{\partial p_{ik}} e_{ij}^2 = p_{ik} + 2\alpha e_{ij} q_{kj}$$

$$q'_{kj} = q_{kj} + \alpha \frac{\partial}{\partial q_{kj}} e_{ij}^2 = q_{kj} + 2\alpha e_{ij} p_{ik}$$

Here, α is a constant whose value determines the rate of approaching the minimum.

Regularization

The above algorithm is a very basic matrix factorization algorithm. A common extension to this basic algorithm is to introduce regularization to avoid overfitting. This is done by adding a parameter β and modify the squared error as follows:

$$e_{ij}^2 = (r_{ij} - \sum_{k=1}^{K} p_{ik} q_{kj})^2 + \frac{\beta}{2} \sum_{k=1}^{K} (\| P \|^2 \| + Q \|^2)$$

In other words, the new parameter β is used to control the magnitudes of the user-feature and item-feature vectors such that P and Q would give a good approximation of R without having to contain large numbers. The new update rules for this squared error can be obtained by a procedure similar to the one described above. The new update rules are as follows.

$$p'_{ik} = p_{ik} + \alpha \frac{\partial}{\partial p_{ik}} e_{ij}^2 = p_{ik} + \alpha(2 e_{ij} q_{kj} - \beta p_{ik})$$

$$q'_{kj} = q_{kj} + \alpha \frac{\partial}{\partial q_{kj}} e_{ij}^2 = q_{kj} + \alpha(2 e_{ij} p_{ik} - \beta q_{kj})$$

2.2 RELATED WORK

The increasing relevance of groups of users on the social web has led to a significant expansion of the group recommendation system (GRS). Most recent RS surveys have included important sections to explain the state-of-the-art GRS: [8] classifies GRS based on the type of items that are recommended: text-based items (books, documents, and webpages), multimedia items (music and movies), or tourism items (attractions, accommodation, and restaurants).

MF based CF is the most popular algorithm to compute single user recommendations; however, in spite of its popularity and the increasing relevance of GRS, based on our knowledge, there are no methods that attempt to combine GRS and MF to compute the user and item groups latent factor. The most similar approach is by Ortega et al.[1]. The key factor of the proposed method is to compute the group's factors representing the group-item interactions in the latent factor space. They defined three approaches to compute these factors.

After Factorization (AF): It computes the group's factors by merging the factors of the users that belong to the group. This approach can be seen as the baseline of the GRS using MF based CF.

Before Factorization (BF): It models the group of users by building a virtual user representing the item preferences of the users of the group. To compute the group's factors, it uses the folding-in technique on the virtual user to add it to the factorized model.

Weighted BF (WBF): This is an extension of BF. It adds weight to each item that the virtual user has 'rated'. These weights will be computed based on the number of ratings that each item has received from the group's users and the consensus of these ratings. The items with the highest weights will contribute more when we compute the group latent factors than those with a low weight.

Another similar approach is by Christensen et al.[2] using the group recommendation system with matrix factorization which proposes to modify the MF model to include a wide variety of sociological factors such as cohesion, social similarity, and social centrality.

In comparison with our proposed MF based CF method, none of the existing MF based CF methods are able to directly compute the latent factors for a group of users. In the method proposed by Ortega et al.[1], they use the user's latent factor to compute the latent factor of the group by combining the latent factors of the user's in the group. In their other approach to compute group latent factors, they add weight to each item based on the number of ratings the item has received. The Item with the highest weight will contribute more to the group latent factor. But in our proposed method numbers of ratings of items don't affect the latent factor of the group.

In another approach by Christensen et al. [2], they include a wide variety of sociological factors such as cohesion, social similarity, and social centrality to compute the group latent factor. Our method doesn't include any sociological factors to compute the group latent factor, we used the user-item interaction to compute the latent factor for the group.

GRS has been used in different areas: tourism, entertainment, web, among others. The most popular method to compute recommendations for a group of users is KNN based CF by Baltrunas et al.[3] uses lists of individual recommendations to aggregate recommendations for group individuals into one recommendation list for the group of users. Berkovsky et al.[4] propose aggregating the users' predictions to build the group's predictions of unknown items. Bobadilla et al.[5] defines the set of neighbors of the group as the intersection of the sets of neighbors of each user of the group. Ortega et al.[7] presents a similarity metric to compute the KNN set of the group of users. Ghazarianand et al.[6] improves GRS by resolving the data sparsity problem of KNN based CF using a support vector machine learning model that computes similarities between items.

CHAPTER 3

MATRIX FACTORIZATION BASED GROUP RECOMMENDER SYSTEM

In this chapter, we present a matrix factorization based approach for group recommendation. Certain scenarios recommend a set of items or a set of item groups to a group of users, and we call this a group recommendation problem. Group recommender systems aim to provide a set of recommendations that satisfies the preferences of all users in a group.

MF models' main idea is to factorize the original rating matrix into two matrices that represent user-item interactions. The critical concept of the proposed method is to factorize the original matrix into four matrices representing users (U), items (V), user groups (U^G), and item groups (V^G) latent factors. These matrices are later used to recommend items or groups of items as a package to users or groups of users. Figure 5 represents the matrix factorization process for a group recommendation system. The rating matrix is decomposed into four low-dimensional matrices representing users, items, user groups, and item groups latent factors.

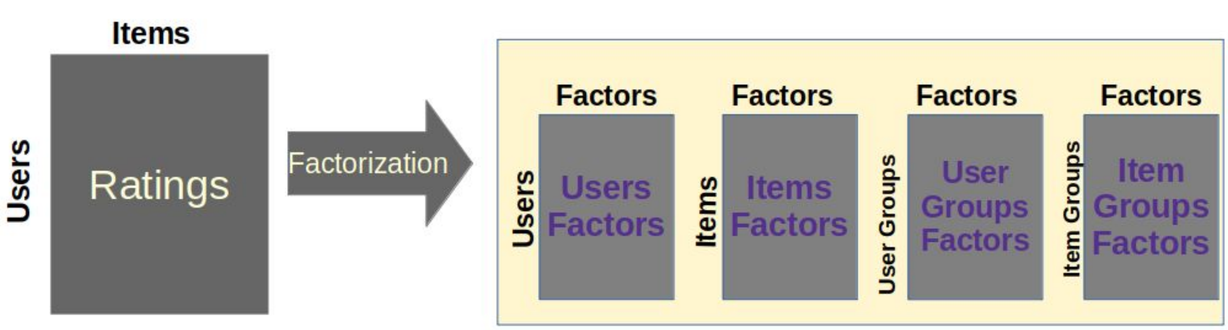

Figure 5: Matrix Factorization process for Group Recommendation System

The advantage of it over the standard nearest neighborhood is that even though two users haven't rated any same movies, it's still possible to find the similarity between them if they share similar underlying tastes, again latent features.

3.1 PROPOSED APPROACH

Let Y be the user-item rating matrix of size M×N, where M is the number of users and N is the number of items. The rating matrix is decomposed into four low-dimensional matrices representing users (U), items (V), user groups (U^G), and item groups (V^G) latent factors/features. We have a k_1 number of user groups and k_2 number of item groups. Also, we can assume that we'd like to discover K latent factors.

In this way, each row of U would represent the strength of the associations between a user and the features. And each row of V would represent the strength of the associations between an item and the features. Similarly, Each row of U^G represents the strength of the associations between the user group and the features and each row of V^G represents the strength of the associations between the item group and the features.

The model used in this paper factorizes the rating matrix into the following elements:

$U_i = (U_{i,1} .....U_{i,k})$ represents the factor vector of the user i .

$V_j = (V_{j,1} .....V_{j,k})$ represents the factor vector of the item j .

$U^G_u = (U^G_{u,1} .....U^G_{u,k})$ represents the factor vector of the user group u.

$V^G_g = (V^G_{g,1} .....V^G_{g,k})$ represents the factor vector of the item group g.

Our task, then, is to find four matrices U=M×K and V=N×K, $U^G = k_1 \times K$ and $V^G = k_2 \times K$ such that the product of U and V approximates Y, U^G approximates the combined latent factor of user groups and V^G approximates the combined latent factor of item groups. Now, we have to find a way to obtain U, V, U^G, and V^G.

To approach this problem, we have to first initialize the matrices with random values and calculate the error between the estimated rating and the real rating as well as user groups and user's latent factors, item groups and items latent factors and then try to minimize this error iteratively.

To learn the factor vectors U_i, $V_{j,}$ $U^G_{u,}$ and V^G_g the system minimizes the following expression for a set of known ratings:

$$\min_{U,V,U^G,V^G} J(U,V,U^G,V^G) = \sum_{(i,j)\in\Omega} (Y_{i,j} - U_i V_j^T)^2 + \lambda_1 \sum_{i=1}^{M} \| U_{I_i}^G - U_i \|^2 + \lambda_2 \sum_{j=1}^{N} \| V_{J_j}^G - V_j \|^2 + \lambda_3 \| U \|^2 + \lambda_4 \| V \|^2$$

Where $Y_{M\times N}$ is a user-item rating matrix wherein M is the number of users and N is the number of items. Ω is the set of observed entries, I_i represents the group of user i, and J_j represents the group of an item j. λ_1, λ_2, $\lambda_{3,}$ and λ_4 are parameters that control the training process. The matrices are initialized randomly. To minimize the error, we have to know in which direction we have to modify the values of U_{ik}, V_{kj}, $U^G_{uk,}$ and V^G_{gk}. In other words, we need to know the gradient at the current values. Therefore, we differentiate the above equation with respect to U_{ik}, V_{kj}, $U^G_{uk,}$ and V^G_{gk} separately. After obtaining the gradient, we update U_{ik}, V_{kj}, $U^G_{uk,}$ and V^G_{gk} as follows.

$$U_{ik} = U_{ik} + \alpha((Y_{ij} - U_{ik}V_{kj})V_{kj} + \lambda_1(U^G_{I_ik} - U_{ik}) - \lambda_3 U_{ik}),$$

$$V_{kj} = V_{kj} + \alpha((Y_{ij} - U_{ik}V_{kj})U_{ik} + \lambda_2(V^G_{kJ_j} - V_{kj}) - \lambda_4 V_{kj}),$$

$$U^G_{uk} = U^G_{uk} - \alpha\lambda_1 \sum_{u=I_i\ \forall\ i\in(0,M)} (U^G_{uk} - U_{ik}),$$

$$V^G_{gk} = V^G_{gk} - \alpha\lambda_2 \sum_{g=J_j\ \forall\ j\in(0,N)} (V^G_{gk} - V_{jk}).$$

Where α is a parameter that controls the speed of the learning process and parameters λ_1, λ_2, $\lambda_{3,}$ and λ_4 are used to control the magnitude of user group feature, item group feature, user-features and item-features respectively.

We then use these matrices to recommend items or groups of items as packages to a user or a group of users.

3.2 RECOMMENDATIONS

After learning the users, items, user groups, and item groups latent factors we then need to generate recommendations for the users and the user groups.

1. **Items Recommendation**
   Product of U and V approximate the ratings of users will give to items and the product of U^G and V approximate the ratings of user groups will give to items.
2. **Item Groups Recommendation**
   Similarly, the product of U and V^G approximate the ratings of users will give to item groups and the product of U^G and V^G approximate the ratings of user groups will give to item groups.

We use these predicted values to compute top-K recommendations for a user or a group.

CHAPTER 4

EVALUATION

4.1 EXPERIMENTAL SETUP

We have performed experiments with benchmark datasets (Movie-Lens)[10]. Table 1 summarizes the details of these datasets. We fine-tuned All the parameters according to the dataset by validating the results with different settings. Table 2 shows the parameters used in the experiment. MovieLens datasets do not contain information about the user and item groups, so we generated random groups of users and items of different sizes $\in [2, 12]$. We selected 30% of ratings as test ratings, and the remaining are used during the training phase. In each experiment, we generated the set of recommendations for each group of users, and we computed their precision and recall. We averaged the evaluation metrics values based on the group size. We set three different types of groups: small groups (2 to 4), mid-size groups (5 to 8), and large groups (9 to 12). We aim to analyze the impact of the group size on the quality of recommendations. To avoid fluctuations generated by the random selection of the groups and the random initialization of the MF model's latent factors, we repeated each experiment multiple times and averaged the results.

	MovieLens (1M)	MovieLens (100k)
Number of Users	6,040	943
Number of Items	3,706	1,682
Number of Ratings	1,000,209	100,000
Rating scale	1-5	1-5

Table 1: Information about the datasets used in the experiments

Database	MovieLens (1M)	MovieLens(100k)
Test Data	30%	30%
Group Size	2-12	2-12
No. of Factors	30	20
Learning Rate	0.00008	0.0004
λ1,λ2,λ3,λ4	10	10
Max. Iteration	500	500
Relevant / Recommendation Threshold	3	3
Number of Recommendations	20	20

Table 2: Parameters used in the experiment

4.2 EVALUATION MATRICES

After almost two decades of research on CF algorithms, various researchers came up with many evaluation metrics. This section presents the various evaluation metrics used to evaluate the prediction accuracy, effective implementation of the MF models with CF algorithms in the recommendation system.

4.2.1 Mean Absolute Error (MAE)

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. It is computed using the formula :

$$MAE = \frac{1}{n}\sum_{j=1}^{n} | y_j - \hat{y}_j |$$

Where n is the number of users, y_j is the actual observed ratings of user j and $\hat{y}_j$ is the prediction of user j.

4.2.2 Root Mean Squared Error (RMSE)

The RMSE is related to the previous metric i.e, MAE. The reason for using this metric is that these errors can have the greatest impact on the user's decision. This can be computed using the formula :

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

4.2.3 Precision and Recall

Precision is defined as the ratio of relevant items that have been recommended to recommended items. Precision can be calculated using the formula :

$$Precision = \frac{(Relevant\ Items\ \cap\ Recommended\ Items)}{Recommended\ Items}$$

Recall is defined as the proportion of relevant items that have been recommended to the total number of relevant items. Recall is calculated by the formula:

$$Recall = \frac{(Relevant\ Items\ \cap\ Recommended\ Items)}{Relevant\ Items}$$

4.3 EXPERIMENTAL RESULTS

Group Size	2-4		5-8		9-12	
	Precision	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**
MovieLens (100k)	0.76	0.71	0.75	0.72	0.75	0.72
MovieLens (1M)	0.69	0.72	0.69	0.71	0.69	0.71

Table 3: Precision and Recall comparison for Recommendation of Items to user groups on different group size

Group Size	2-4			5-8			9-12		
	KNN	MF (existing)	MF (proposed)	KNN	MF (existing)	MF (proposed)	KNN	MF (existing)	MF (proposed)
MovieLens (1M)	0.61	0.84	0.69	0.54	0.77	0.69	0.49	0.73	0.69

Table 4: Precision comparison between the proposed method (MF), existing MF based CF method and best memory-based CF method (KNN) for user group recommendations

Table 3 compares the precision and recall for recommendations of Items to user groups for different group sizes. In **Table 4** we have compared the precision of the method proposed in this paper with the MF based CF group recommendation method proposed by Fernando Ortega [1] and the best KNN based CF recommendation method proposed by [9].

We compare the precision for the top 20 recommendations with existing recommendation models. In the comparison between the traditional KNN based collaborative filtering and the proposed MF based collaborative filtering for groups of users, there is a significant increase in the quality of recommendations. We got average precision 0.69 for different sizes of groups. In comparison with the existing MF based CF method for groups of users, we got less precision value which can be later improved in the future.

In the existing recommendation methods for groups of users, the precision value decreases with the increasing group size but in our proposed method there is very minimal effect of group size on quality of recommendation.

Group Size	2-4		5-8		9-12	
	Precision	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**
MovieLens (100k)	0.74	0.47	0.75	0.36	0.75	0.31
MovieLens (1M)	0.65	0.64	0.64	0.56	0.64	0.54

Table 5: Precision and Recall comparison for Recommendation of Item groups to users on different group size

Group Size	2-4		5-8		9-12	
	Precision	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**
MovieLens (100k)	0.72	0.49	0.70	0.35	0.71	0.30
MovieLens (1M)	0.63	0.68	0.62	0.65	0.61	0.64

Table 6: Precision and Recall comparison for Recommendation of Item groups to user groups on different group size

Table 5 compares the precision and recall for recommendations of Item groups to the users for different group sizes. **Table 6** compares the precision and recall for the recommendation of Item groups to user groups for different group sizes.

CONCLUSION AND FUTURE WORK

In this paper, we proposed a matrix factorization based CF approach for group recommendations. We tested our approach for different group sizes and compared it with the traditional KNN based CF. In the comparison between the traditional KNN based CF and the proposed MF based CF for groups of users, we can observe a significant increase in the quality of recommendations. Also, In KNN based CF and other group recommendation approaches group size affects the quality of recommendation (e.g precision decreases with increasing group size). But in our MF based CF approach, there is very minimal effect of group size on quality of recommendation. There are no methods that attempt to provide the item group recommendations to the user and user group. We are able to provide the item group recommendations to users and user groups with good precision.

As future work, we propose to implement different state-of-art methods with our proposed method to improve the quality of recommendation for a group of users.

REFERENCES

1. Fernando Ortega, Antonio Hernando, Jesus Bobadilla, Jeon Hyung Kang, Recommending items to a group of users using Matrix Factorization based Collaborative Filtering, in Informatics and Computer Science Intelligent Systems Applications, 2016 Elsevier Inc.
2. I. Christensen, S. Schiaffino, Matrix factorization in social group recommender systems, in Proceedings of the 12th Mexican International Conference on Artificial Intelligence (MICAI), IEEE, 2013, pp. 10–16.
3. L. Baltrunas, T. Makcinskas, F. Ricci, Group recommendations with rank aggregation and collaborative filtering, in Proceedings of the Fourth ACM Conference on Recommender Systems, ACM, 2010, pp. 119–126.
4. S. Berkovsky, J. Freyne, Group-based recipe recommendations: analysis of data aggregation strategies, in Proceedings of the Fourth ACM Conference on Recommender Systems, ACM, 2010, pp. 111–118.
5. J. Bobadilla, F. Ortega, A. Hernando, J. Bernal, Generalization of recommender systems: collaborative filtering extended to groups of users and restricted to groups of items, Expert Syst. Appl. 39 (1) (2012) 172–186.
6. S. Ghazarian, M. Nematbakhsh, Enhancing memory-based collaborative filtering for group recommender systems, Expert Syst. Appl. 42 (7) (2014) 3801–3812.
7. F. Ortega, J. Bobadilla, A. Hernando, A. Gutiérrez, Incorporating group recommendations to recommender systems: Alternatives and performance, Inf. Process. Manag. 49 (4) (2013) 895–901.
8. J. Lu, D. Wu, M. Mao, W. Wang, G. Zhang, Recommender system application developments: A survey, Decis. Support Syst. 74 (2015) 12–32.
9. F. Ortega, J. Bobadilla, A. Hernando, A. Gutiérrez, Incorporating group recommendations to recommender systems: Alternatives and performance, Inf. Process. Manag. 49 (4) (2013) 895–901.
10. F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015), 19 pages. DOI=http://dx.doi.org/10.1145/2827872
11. Dheeraj Bokde, Sheetal Girase, Debajyoti Mukhopadhyay. Matrix factorization Model in Collaborative Filtering Algorithms: A Survey. Published by Elsevier B. V. 2015