The CSI Journal on
Computer Science and Engineering
Vol. 18, No. 2, 2021
Pages 32-40
Regular Paper

A Multi-View Group Recommender System based on Trust and Ratings

Maryam Sadeghi, Seyyed Amir Asghari, Mir Mohsen Pedram

Department of Electrical and Computer Engineering, Faculty of Engineering, Kharazmi University
Tehran, Iran

Abstract

Sometimes, depending on the type of system, it is not possible to offer a list of items for each user individually and the number of items in the recommended lists should be limited, therefore group recommender systems will be used. So far, various group recommender systems have been presented. Most of them have been investigated from the standpoint of user preferences (rating matrix). In this paper, the proposed Multi-View Group Recommender System (MVGRS), investigates from two points of view, i.e., user preferences and social connections (trust). This system, with multi-view, first offers the individual recommendations, then generates the group recommendation by aggregating individual recommendations. The system has been tested with the different numbers of rating clusters and trust clusters. Finally, the error of the MVGRS is compared with the error of the similar single-view group recommender system. The results show that the MVGRS outperforms the single-view group recommender system.

Keywords: Group Recommender system, Multi-view, Rating, Social Connections, Clustering.

1. Introduction

The root of research about recommender systems is traced back to the 1990s. Various types of recommender software for different domains have been produced in recent years. Today, the use of recommender systems has become a necessity and many internet sites use recommender systems to provide suitable services to their customers [1].

Recommended systems are divided into two general types based on the user's feedback [2]:

Collaborative Filtering: Collaborative filtering algorithms are based on the similarity of the users who have the same preferences as the target user. These types of algorithms have two main categories:

User-Based Collaborative Filtering (UBCF): The UBCF systems use neighbors of the target user to predict non-rated items. The neighbors of the target user are the users who have similar interests in items [3].

Item-Based Collaborative Filtering (IBCF): In this case, neighbors are defined as similar items to items that the target user rated before. The IBCF systems use neighbors of the target user to predict non-rated items [3].

Content-based Filtering: Content-based filtering systems concentrate on the attributes of items. The system recommends a list of items based on the records of the needs and tastes of the user (such as purchased items) that were previously saved in the user's profile. The assessment of the similarity between the items selected in the past by the user and the new items is based on the relationship between the attributes and characteristics of the two items which are compared [4].

Other approaches for designing recommender systems are Demographic-based, Knowledge-based and Utility-based, but these methods are not used as common as previous methods, and usually are combined with collaborative filtering and content-based filtering methods [5].

1.1. Basic concepts in recommender systems

In the following, the basic concepts in recommender systems are briefly introduced.

Items are objects that are offered to the user. If an item is effective in the recommending process, it will have a positive role; otherwise, it will have a neutral/negative role [6].

A target user is a person who the system generates an appropriate recommendation to him [7].

Rating Matrix is a matrix, in which each row of it is a token of a user, and each column represents an item. Each matrix entry represents the user's vote to the corresponding item [8, 9].

In most of the current research, the definition of trust relates to the various categories to which it relates, and there is no clear definition for it in many cases. But in this research, trust has been taken explicitly by the user. In this way, each user specifies his trust in other users in the existing dataset [10, 11]. Recommender systems have been generated for recommending to individual users in order to help them to find their relevant and appropriate information. Various recommender systems have been proposed during recent years. In some situations, recommending to individual users is tough or the system may face some limitations for the individual recommendations. For example, it is not possible to offer a list of recommendations to each user; to overwhelm these kinds of obstacles group recommender systems have been proposed.

For instance, a group recommender system may recommend television programs for a group of people to view, based on models of all group members' preferences. In some circumstances, the group recommender system should recommend in a multi-view. Consider the following example: a recommender system may offer some roommates for a group of students in a dormitory. It is so important for the students to live with a person who they trust. To solve this issue, a group recommender system should recommend in a multi-view of student preferences and student trusts simultaneously. Many group recommender systems have been proposed in recent years but most of them have been investigated from a single view of rating.

In this paper, a Multi-view group recommender system has been proposed which investigates from two standpoints of user preferences (ratings) and social connection (trust).

The rest of this paper is organized as follows: Section 2, discusses previous related works on group recommender systems. Section 3 introduces the overall MVGRS framework and describes it in detail. Section 4, applies the MVGRS for a dataset to evaluate and analyze its performance comparison with the competing method. Finally, section 5, provides the conclusion and some future research directions.

2. Related works

Different group recommender systems designed in the past few years. Group recommender systems are ones where more than one person is considered in recommending process [12- 14]. Group recommender systems are used when there are some limitations for recommending to individuals (for example, it is not possible to recommend a list of items to each user). Some of the group recommender systems are introduced, in the following. The system that has been proposed in [15], is a system for recommending a restaurant to a group of people who want to eat together, and users vote for their favorite items by filling out a form. Depending on user rating, the preferences of each user are predicted. Then, by aggregation of individual predictions, it is recommended to the group.

In the system that has been introduced in [16], the authors believe that social interaction within group members is the important thing in group recommender systems. For this reason, they presented coFeel [16] as an interface for using emotions to raise social interaction in group recommender systems. In the article [17], the authors propose a new group recommender system that can deal with technological constraints like bandwidth limitations.

Rating prediction in group recommender systems can be done with one of the following three approaches [18]:

1. Creation of group preferences models and prediction of non-rated items for each group using the model.
2. Prediction of ratings for items that have not been rated by the users and merging individual recommendations generated for members of a group.
3. Aggregating of predictions generated for each user to recommend to a group.

The architecture of the system changes according to the chosen method for predicting the ratings. In fact, based on the approach used, the predictive process takes a different input and produces a completely different output [18].

The system that has been proposed in [19] is a system that leverages similarity, knowledge factors, and trust to increase the quality of the group recommender system. This system used the MovieLens dataset [20] that does not have any trust factor. Due to the absence of trust factors in the dataset, the authors used fuzzy computational models [21] to compute the trust factor. At first, they classified each item into the groups of liked, unliked, or neutral. After the classification of each item, a trust factor was defined. Then a similarity-based trust was calculated and the similarity-based recommendation for the entire group was generated by employing the group recommendation the least-misery [22] strategy.

In this stage, the effectiveness of the group recommendation was reduced. To solve this problem, the authors used some other factors that affect the computation of trust between users. So, knowledge-based trusted user-item ratings have been computed. In the end, to generate the knowledge-based recommendation, the least-misery strategy was applied. The approach applied to 20 users randomly selected from the dataset and compared to the least-misery strategy reference [23]. The results showed the efficiency of the proposed method [19] in comparison to reference [23].

3. Proposed method

In this section, the proposed method and the implementation phases are described in details

3.1. Multi-View Group Recommender System (MVGRS)

In the MVGRS method, the group recommender system is investigated from two points of view of user preferences (rating matrix) and social connection (trust). Two types of information are extracted from users in the system:
Type 1: The interest scores of users to items are specified by rating.
Type 2: The trust of users in other users that are connected to them, and it is a part of the dataset.

According to the user preferences that are shown in the form of rating for items, group priorities can be generated by one of the following three approaches:

- Constructing a group model and predicting for each group
- Merging individual recommendations
- Aggregating individual predictions

The rating prediction plays an essential role in the recommender systems. In fact, the rating for non-rated items should be predicted for each user or each group [18], and the idea that is used in the MVGRS method is to aggregate the individual predictions for group recommendation. For a fair comparison, the reference [18] is first tested with the rating pattern of the FilmTrust dataset [24], then is compared with the MVGRS.

3.2. Implementation of the proposed method

The MVGRS method consists of two phases of training and testing. It is explained in detail in the following. An overviewof the MVGRS method is presented in Fig. 1.

3.2.1. Training phase

It consists of five steps, which are described below.

Step 1: Create rating matrix and trust matrix

The input of this phase is the FilmTrust dataset. The goal is to construct user-item-rating matrix and user-user-trust matrix which are briefly called rating matrix and trust matrix, respectively.

Step 2: Generate a graph for the trust matrix and compute the shortest path between the two nodes

The input of this step is the trust matrix. The goal is to find a trust distance.

A trust graph is generated from the trust matrix. Its nodes symbolize users and its edges symbolize the trust of each user in another user. Since the run-time of the MVGRS is as important as other features, the shortest path between two nodes is taken into account. By using the Euclidian distance, the shortest path between the two nodes is computed. The similarity between them is computed which has been shown in Eq. (1), and then the trust distance between the two nodes is obtained by using Eq. (2).

$$S_{i,j} = \frac{1}{P_{i,j}} \quad (1)$$

$$D_{i,j} = 1 - S_{i,j} \quad (2)$$

Where:

$P_{i,j}$: The shortest distance between the two nodes i, j

$D_{i.j}$: The trust distance between each two nodes

$S_{i.j}$: The similarity between the two nodes

Step 3: Cluster with Complete Linkage method

The input of this step is the trust distance. The purpose is to create a trust-based cluster. Using the trust distance obtained in the previous step, the trust matrix is clustered with the Complete Linkage method. The number of clusters is initially assumed to be 100. But this clustering matrix is also tested to obtain appropriate clustering with various numbers of clusters.

Step 4: Cluster the rating matrix using the K-means method

The input of this step is the rating matrix. This matrix is clustered by k-means clustering.

Step 5: Determine a cluster based on rating and trust

The input of this phase is two clusters types:

I. Rating-based cluster
II. Trust-based cluster

The goal of this step is to combine these two types of clustersand create a new cluster based on trust and rating simultaneously. The diagram of this phase has been shown inFig. 2. The idea of this step is as follows. First, for each rating cluster, the paired trust cluster is found, and the two clusters are combined. In this way, to find the paired cluster, a similarity measure is defined, as shown in Eq. (3).

$$S = Similarity(\mathrm{C}_r, \mathrm{C}_t) = \frac{|\mathrm{C}_r \cap \mathrm{C}_t|}{|\mathrm{C}_t|} \quad (3)$$

Where:

C_t : Trust cluster

C_r : Rating cluster

S : Similarity measure between the trust cluster and the rating cluster

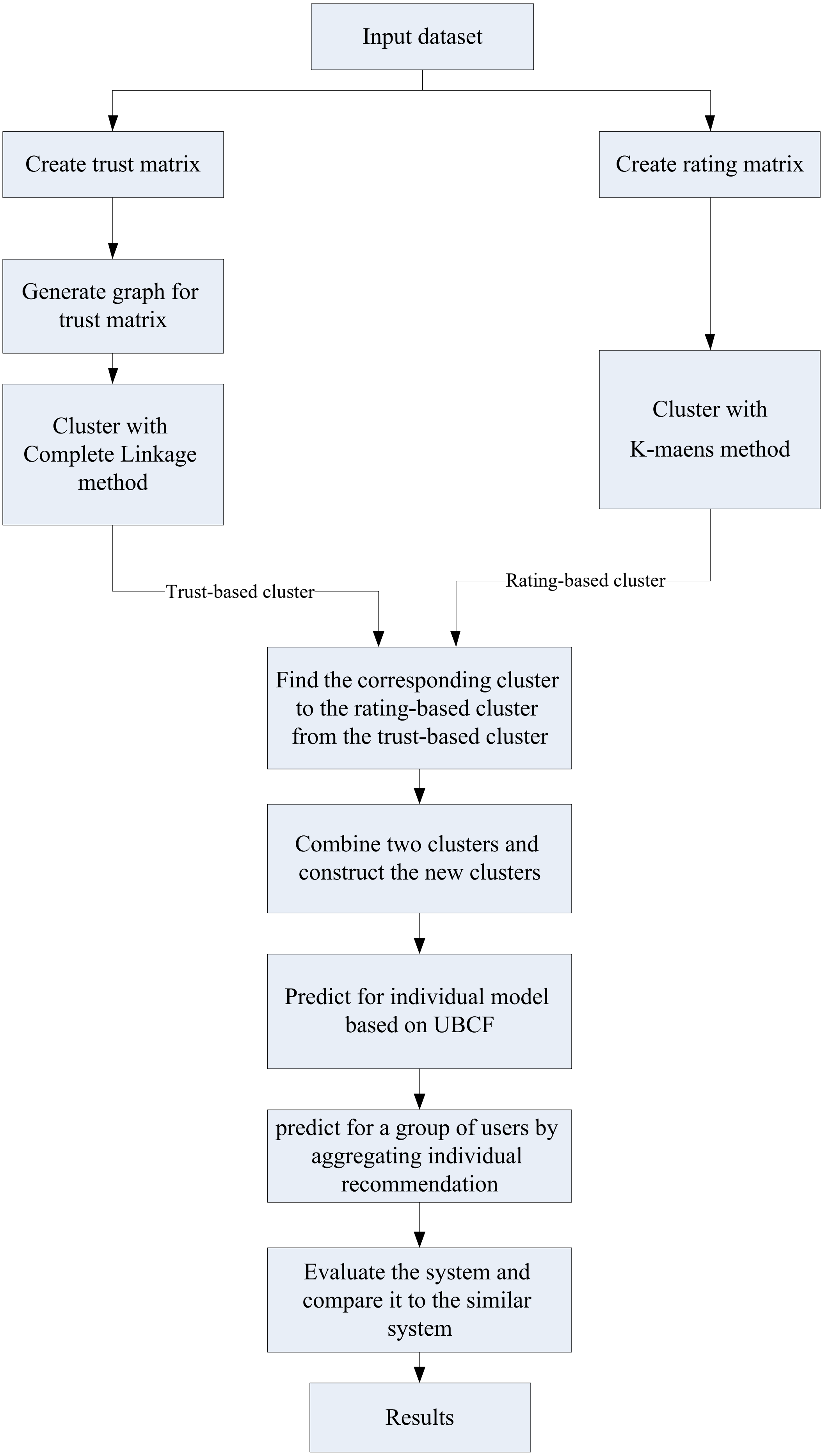

Fig. 1. An overview of the MVGRS method

The similarity is computed for each rating cluster and its corresponding trust clusters. The numbers of computed similarities are based on the number of corresponding trust clusters of each rating cluster. Then, these similarities are compared to each other and, the cluster with the maximum similarity to the rating cluster is selected. Finally, for each rating cluster, a paired trust cluster is found. Each rating cluster is combined with the paired trust cluster, which result in a new cluster.

3.2.2. Testing phase

In this section, predictions are performed using test data. This section has some steps which are described below.

Step 1: Predict for individual model based on User-Based Collaborative Filtering (UBCF)

The combined cluster resulting from the integration of the rating and trust cluster is the input of this step. The goal is to estimate the individual prediction for users, which is computed as the weighted average of the items. In this step, the rating of the test data that is available in each cluster is predicted using the UBCF method. In this way, the weighted average of the items is computed and proposed the mean of the data.

There is a difference between the UBCF method used in the MVGRS and the usual UBCF method. The difference is in selecting the nearest neighbor. In MVGRS, the nearest neighbor is selected from the combined cluster that the user is located in.

Step 2: Predict rating for a group of users by aggregating individual recommendation

In this step, the individual recommendations are considered as the inputs, which were determined in the previous phase based on the combined cluster.

The target is the recommendation to a group of users of each cluster. The rating has been predicted for each item that is non-rated by group members. The predictions are averaged and then the averaged predictions are suggested to the group of test users in the cluster.

Step 3: Compute RMS Error

In this section, the error of the MVGRS is evaluated by RMSE and compared to the reference recommender system [18].

4. Results

In this section, the error of the MVGRS method is compared to the reference recommender system [18] on the FilmTrust dataset.

4.1. Simulation Environment

This system is implemented in R software version 3.4.3. The "igraph" library has been used to generate the trust graph in step 2 of the training phase and the "recommender lab" library has been used to construct an individual recommendation model with the UBCF method in the testing phase [25].

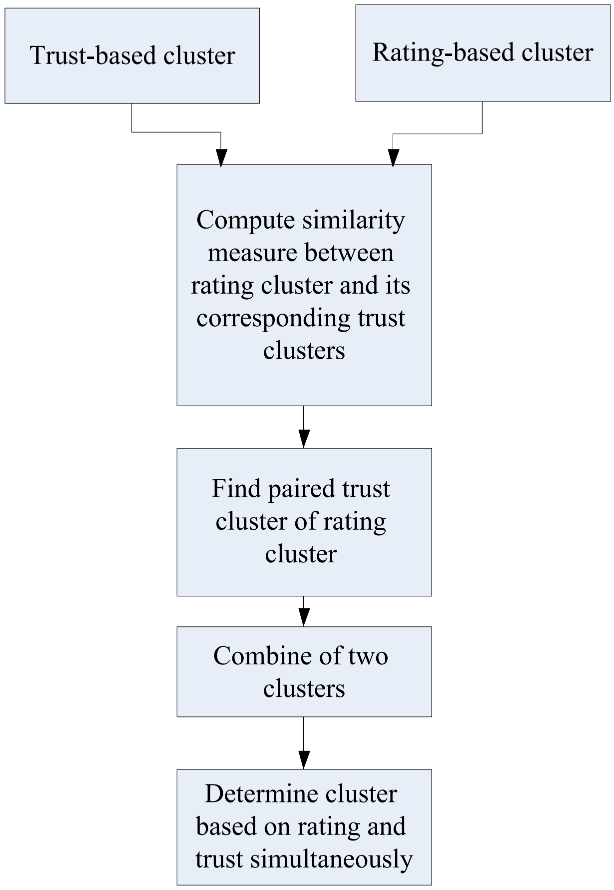

Fig. 2. The fifth step in details

4.2. Dataset

In this paper, the real dataset has been used to examine the proposed method. This dataset is the FilmTrust dataset, which is about movies and allows users to share movie ratings and also explicitly specify other users as trusted neighbors. In this dataset, the rankings are in the range of 0.5 to 4 and can be incremented by 0.5 at each stage. It is worth mentioning that trust is asymmetric in the dataset. This means that if the user trusts another user, this is not necessarily a two-way trust. In the dataset, there are 1508 users who have rated 2071 items. The number of ratings available for this dataset is 35497 [24]. The summary of the dataset is shown in table 1.

Table 1. Summary statistics of the dataset [24]

Dataset	Users	Items	Ratings	Trust
FilmTrust	1508	2071	35,497	2853

4.3. Experimental results

In this section, the numbers of trust clusters are considered 100, 90, and 80 and the number of rating clusters are considered 50, 60, and 40. The proposed method was run 10 times with different training data and test data. The error of the MVGRS method and the group recommendation system based on aggregating of individual recommendations [18] are compared in Fig. 3, Fig. 4 and Fig. 5. The experiments show the better results when trusts are considered in clusters for group recommendations.

In Fig. 6 the comparison of the RMSE for the various numbers of the rating clusters and the trust clusters has been shown.

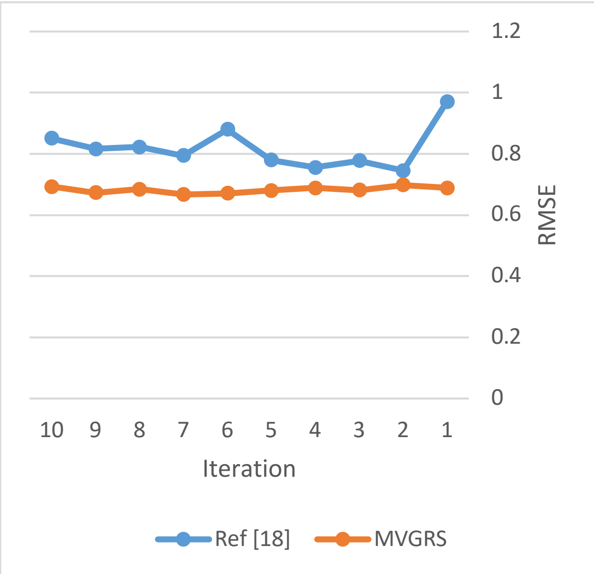

Fig. 3. Comparison of MVGRS error and group recommender system [18] with the number of trust clusters 100 and the number of rating clusters 50.

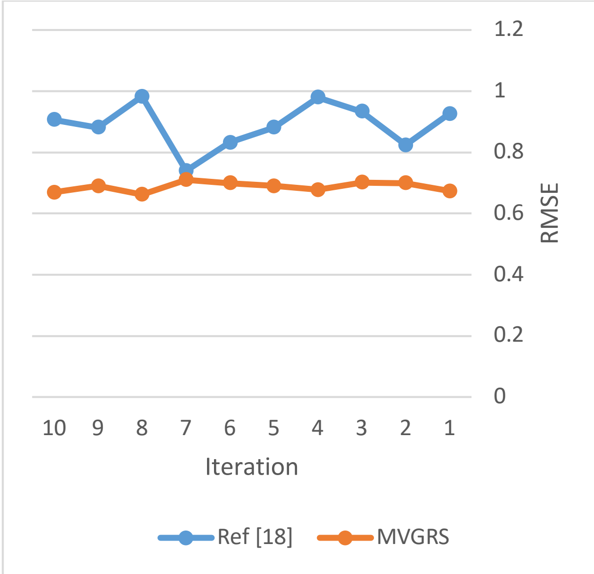

Fig. 4. Comparison of MVGRS error and group recommender system [18] with the number of trust clusters 90 and the number of rating clusters 60

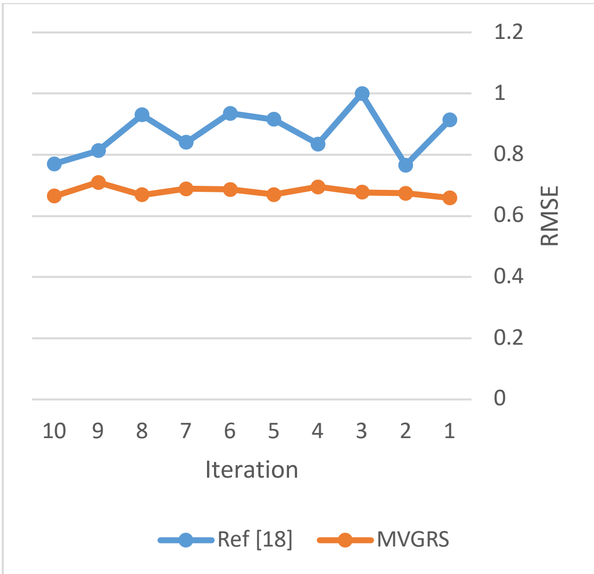

Fig. 5. Comparison of MVGRS error and group recommender system [18] with the number of trust clusters 80 and the number of rating clusters 40

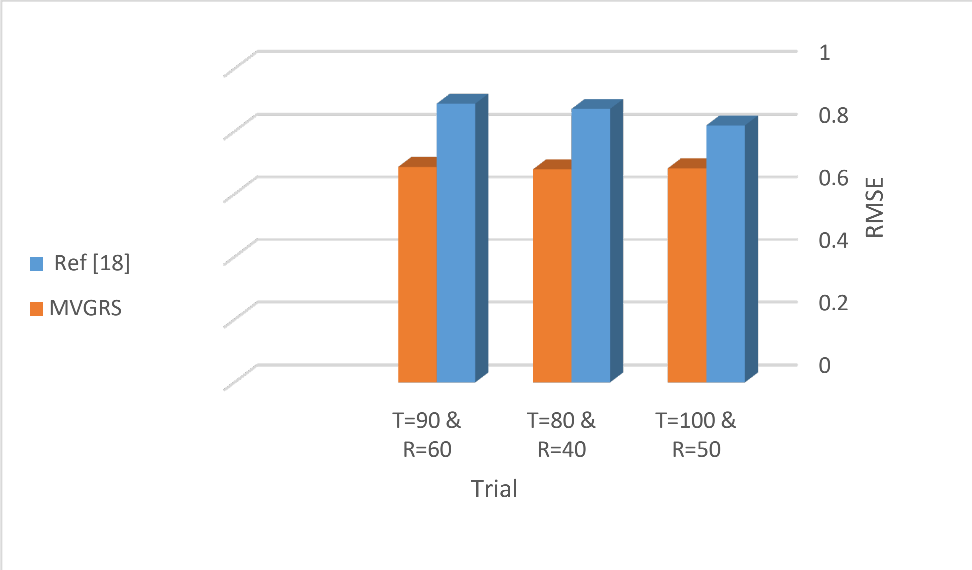

Fig. 6. Compare the RMSE in the different number of clusters for the rating cluster and trust cluster

5. Conclusion and future studies

The new combined clustering based on rating and trust is presented in MVGRS, which has not been presented in any group recommender system previously. Experimental results show that MVGRS can reduce error significantly with respect to the group recommender system that evaluates its system from just a single point of view. The results show that the MVGRS on average reduced RMSE 0.20171897 in the best case and 0.13668361 in the worst case compared to the same method.

Clustering accuracy affects the total performance of the group recommender system, as future work, we can apply the other clustering methods, such as evolutionary algorithm-based clustering, to gain better results.

Acknowledgments

We would like to show our gratitude to Dr. Ghorban Kheradmandian for sharing his pearls of wisdom with us during the course of this research.

References

[1] J.A. Konstan and J. Riedl, "Recommender systems: from algorithms to user experience", *User Model User-Adap Inter*, pp. 22-101, 2012, https://doi.org/10.1007/s11257-011-9112-x.

[2] J. Lu, D. Wu, M. Mao, W. Wang and G. Zhang, "Recommender System Application Developments: A Survey", *Decision Support Systems*, vol. 74, pp. 12-32, 2015, https://doi.org/10.1016/j.dss.2015.03.008.

[3] S. Kant and T. Mahara, "Merging user and item based collaborative filtering to alleviate data sparsity", *International Journal of System Assurance Engineering and Management*, vol. 9, pp. 173–179, 2016, https://doi.org/10.1007/s13198-016-0500-9.

[4] A. Gordillo, E. Barra and J. Quemada, "A Hybrid Recommendation Model for Learning Object Repositories", *in IEEE Latin America Transactions*, vol. 15, pp. 462-473, 2017, DOI:10.1109/TLA.2017.7867596.

[5] T. Horvath and A. Carvalh, "Evolutionary Computing in Recommender system: a review of recent research", *Natural Computing*, pp. 1-22, 2017, https://doi.org/10.1007/s11047-016-9540-y.

[6] J. Bobadilla, F. Ortega, A. Hernando and A. Gutierrez, "Recommender Systems Survey", *Knowledge-based System*, vol. 46, pp. 109-132, 2013, https://doi.org/10.1016/j.knosys.2013.03.012.

[7] G. Suganeshwari, and S.P. Syed Ibrahim, "A Survey on Collaborative Filtering Based Recommendation System", *Proceedings of the 3rd International Symposium on Big Data and Cloud Computing Challenges (ISBCC'16), Smart Innovation, Systems and Technologies, Springer, Cham,* vol. 49, pp. 503-518, 2016, https://doi.org/10.1007/978-3-319-30348-2_42.

[8] M. Sadeghi and S.A. Asghari, "Recommender Systems Based on Evolutionary Computing: A Survey", *Journal of Software Engineering and Applications*, vol. 10, pp. 407-421, 2017, DOI: 10.4236/jsea2017.105023.

[9] S. Pero and T. Horvath, "Opinion-Driven Matrix Factorization for Rating Prediction", *User Modeling, Adaptation, and Personalization, UMAP 2013, Lecture Notes in Computer Science, Springer*, vol. 7899, pp. 1-13, 2013, https://doi.org/10.1007/978-3-642-38844-6_1 .

[10] S. Berkovsky, R. Taib and D. Conway, "How to Recommend?: User Trust Factors in Movie Recommender Systems", *Proceedings of the 22nd International Conference on Intelligent User*

Interfaces, pp. 287-300, 2017, https://doi.org/10.1145/3025171.3025209.

[11] G. Guo, J. Zhang, N. Yorke-Smith, "Leveraging multiviews of trust and similarity to enhance clustering-based recommender systems", *Knowledge-Based Systems*, vol. 74, pp. 14–27, 2015, https://doi.org/10.1016/j.knosys.2014.10.016.

[12] J. Masthoff, "Group recommender systems: aggregation, satisfaction and group attributes", *Recommender systems handbook, Springer*, pp. 743-776, 2015, https://doi.org/10.1007/978-1-4899-7637-6_22.

[13] Z. Guo, C. Tang, H. Tang, Y. Fu and W. Niu, "A Novel Group Recommendation Mechanism from the Perspective of Preference Distribution", *In IEEE Access*, vol. 6, pp. 5865-5878, 2018, DOI: 10.1109/ACCESS.2018.2792427.

[14] J. Park and K. Nam, "Group recommender system for store product placement", *Data Min Knowledge Discovery*, vol. 33, pp. 204–229, 2019, https://doi.org/10.1007/s10618-018-0600-z .

[15] J. McCarthy, "Pocket Restaurant Finder: A situated recommender system for groups", *In Workshop on Mobile Ad-Hoc Communication at the 2002 ACM Conference on Human Factors in Computer Systems*, 2002.

[16] Y. Chen, and P. Pu, "Cofeel: Using emotions to enhance social interaction in group recommender systems", *In Alpine Rendez-Vous (ARV), Workshop on Tools and Technology for Emotion-Awareness in Computer Mediated Collaboration and Learning*, 2013.

[17] L. Boratto and S. Carta, "State-of-the-Art in Group Recommendation and New Approaches for Automatic Identification of Groups", *Information Retrieval and Mining in Distributed Environments, Studies in Computational Intelligence, Springer*, vol. 324, pp. 1-20, 2010, https://doi.org/10.1007/978-3-642-16089-9_1 .

[18] L. Boratto and S. Carta, "The rating prediction task in a group recommender system that automatically detects groups: architectures, algorithms, and performance evaluation", *Journal of Intelligent Information Systems, Springer*, vol. 45, pp. 221–245, 2015, https://doi.org/10.1007/s10844-014-0346-z .

[19] N. Choudhary and K.K. Bharadwaj, "Leveraging Trust Behaviour of Users for Group Recommender Systems in Social Networks", *Integrated Intelligent Computing, Communication and Security*, vol. 771, pp. 41–47, 2019, https://doi.org/10.1007/978-981-10-8797-4_5 .

[20] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context." *Acm transactions on interactive intelligent systems (tiis),* vol. 5, pp. 1-19, 2016, https://doi.org/10.1145/2827872 .

[21] V. Kant, K.K. Bharadwaj, "Fuzzy Computational Models of Trust and Distrust for Enhanced Recommendations", *International Journal of Intelligent Systems,* vol. 28, 332–365, 2013, https://doi.org/10.1002/int.21579 .

[22] M. O'Connor, D. Cosley, J.A. Konstan and J. Riedl, "PolyLens: A Recommender System for Groups of Users", *In: Prinz W., Jarke M., Rogers Y., Schmidt K., Wulf V. (eds) ECSCW 2001, Springer*, pp. 199-218, 2001, https://doi.org/10.1007/0-306-48019-0_11 .

[23] L. Baltrunas, T. Makcinskas and F. Ricci, "Group recommendations with rank aggregation and collaborative filtering", *In Proceedings of the fourth ACM conference on Recommender systems,* pp. 119-126, 2010, https://doi.org/10.1145/1864708.1864733 .

[24] G. Guo, J. Zhang, N. Yorke-Smith, "A novel bayesian similarity measure for recommender systems", *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2619–2625, 2013.

[25] M. Hahsler, "recommenderlab: Lab for Developing and Testing Recommender Algorithms", *R package version 3.4.3*, 2018.

Maryam Sadeghi received her B.S.c degree in Computer Engineering- Software from Islamic Azad University, Karaj, Iran in 2014 and her M.S.c degree in Computer Engineering- Artificial Intelligence from Kharazmi University, Tehran, Iran in 2018. Her main research interests include Data Mining, Recommender Systems, Machine Learning, and Artificial Intelligence.
Email: std_m.sadeghi@khu.ac.ir

Seyyed Amir Asghari was born in Lashte Nesha, Gilan, Iran, on June 26, 1984. He received his B.S., M.S., and Ph.D. degrees in Computer Engineering from Amirkabir University of Technology, Tehran, Iran, in 2007, 2009, and 2013 respectively. His interests include reliable and fault tolerant embedded system design, real-time system design, and operating systems. He is currently the associate professor of the faculty of engineering in Kharazmi University of Tehran.
Email:asghari@khu.ac.ir

Mir Mohsen Pedram received his Ph.D. and M.Sc. degree in Electrical Engineering from the Tarbiat Modarres University, Tehran, Iran, and his B.Sc. degree in Electrical Engineering from Isfahan University of Technology, Isfahan, Iran. He is currently an Associate Professor in the Department of Electrical and Computer Engineering at Kharazmi University. His main areas of research are Intelligent Systems, Machine Learning, Data Mining, and Cognitive Science.
Email: pedram@khu.ac.ir

Paper Handling Data:

Submitted: 12-28-2020
Received in revised form: 03-18-2022
Accepted: 03-26-2022
Affiliation of the Corresponding author: Seyyed Amir Asghari, Faculty of Engineering, Kharazmi University, Tehran, Iran