2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)

A Temporal Clustering Approach for Social Recommender Systems

Sajad Ahmadian
Department of Computer Engineering
University of Zanjan
Zanjan, Iran
s.ahmadian@znu.ac.ir

Nima Joorabloo
School of Engineering
RMIT University
Melbourne, Australia
nima.joorabloo@student.rmit.edu.au

Mahdi Jalili
School of Engineering
RMIT University
Melbourne, Australia
mahdi.jalili@rmit.edu.au

Majid Meghdadi
Department of Computer Engineering
University of Zanjan
Zanjan, Iran
meghdadi@znu.ac.ir

Mohsen Afsharchi
Department of Computer Engineering
University of Zanjan
Zanjan, Iran
afsharchim@znu.ac.ir

Yongli Ren
School of Science
RMIT University
Melbourne, Australia
yongli.ren@rmit.edu.au

***Abstract*—**Recommender systems aim to suggest relevant items to users among a large number of available items. They have been successfully applied in various industries, such as e-commerce, education and digital health. On the other hand, clustering approaches can help the recommender systems to group users into appropriate clusters, which are considered as neighborhoods in prediction process. Although it is a fact that preferences of users vary over time, traditional clustering approaches fail to consider this important factor. To address this problem, a social recommender system is proposed in this paper, which is based on a temporal clustering approach. Specifically, the temporal information of ratings provided by users on items and also social information among the users are considered in the proposed method. Experimental results on a benchmark dataset show that the quality of recommendations based on the proposed method is significantly higher than the state-of-the-art methods in terms of both accuracy and coverage metrics.

Keywords*—recommender system, clustering, temporal, social information, graph*

I. Introduction

In recent years, with significant growth of information on the web, individual users can access diverse information online every day. Therefore, the necessity of having an intelligent system is vital to help the users to find their relevant items. Recommender systems (RSs) lead users to quickly find desirable items without being overwhelmed by irrelevant information. RSs have been used in many areas, including e-commerce, advertisement, news, and document management, to improve the cross-selling, user loyalty and fulfill customer requirements [1, 2].

RSs can be divided into three main categories including collaborative filtering (CF), content-based and hybrid methods [3]. CF methods collect preferences of similar users to provide recommendations to a target user. These methods often have several limitations such as data sparsity, cold start, and scalability [4]. On the other hand, content-based methods provide recommendations to a target user based on the item description, the user's history and their interests. These methods suffer from lack of diversity in recommending items to the users [5]. Finally, hybrid methods attempt to combine two or more recommender methods to improve the performance and overcome the drawbacks of any individual method [6].

Social recommender systems use additional information such as friendship, trust, and distrust relations between the users in the recommendation process [7, 8]. The main idea of the systems is that a social relation between two users indicates that they have similar interests. Therefore, the users who have relations with the target user can be considered as the neighbors set. Moreover, clustering approaches can be used to group users into clusters based on social relations between the users and then the clusters can be considered as the neighbor sets of the users. It is an important issue that the users' preferences depend on time and might change in different time periods [9]. Therefore, the time factor can play important role in social recommender systems to provide effective personalized recommendations and to increase the accuracy of predictions. However, the temporal information of the ratings is not considered in the traditional clustering methods.

In this paper, a social recommender system is proposed based on a temporal clustering approach. To this end, a user's network is constructed using the combination of the similarity values and trust relations between the users. The similarity values are calculated based on the temporal information of the ratings provided by the users. Moreover, a graph-based method is applied to find initial centers set of the clusters. Therefore, the initial centers set and also the number of initial clusters can be obtained automatically. Then, the final clusters are obtained using an iterated mechanism on the initial centers set. Finally, the identified clusters are used as the neighbor sets of the users to make recommendations. The proposed method is compared with time-based and clustering-based methods and the experiments results indicate that the proposed method can obtain better performance than the others.

IEEE/ACM ASONAM 2018, August 28-31, 2018, Barcelona, Spain

II. RELATED WORKS

Social recommender systems are advanced techniques which use social information to provide relevant recommendations for the users. In these systems, two types of the social relations can be used including explicit and implicit relations. The explicit relations are explicitly expressed by the users and can be represented as a social relations matrix [10]. On the other hand, the implicit relations can be implicitly obtained by using other information such as interactions between the users [11].

Several methods have been proposed for social recommender systems based on clustering approaches [12-15]. In [12], a clustering framework is proposed based on the combination of the trust networks and collaborative filtering method. Moreover, a fuzzy clustering approach is proposed to group the items into appropriate clusters. Guo et.al [13] proposed a multi-view clustering method for recommender systems which users are iteratively clustered from the views of both rating patterns and also social trust information. In [14], a recommender system is proposed which is based on a graph clustering approach using the combination of the trust information and users' ratings. In [15], a social recommender system is proposed based on an adaptive neighbor selection mechanism. To this end, a clustering approach is applied to group users into several clusters and then the identified clusters are used as the users' neighbors set in the recommendation process.

Time information for the ratings is useful criteria which can be used to track changes in user preferences and behavior over the time [16, 17]. Several studies have attempted to use the temporal information to improve the performance of the recommender systems [18-20]. In [18], an adaptive collaborative filtering algorithm is proposed which takes time into account when predicting users' behavior. A user-based collaborative filtering algorithm is proposed in [19] using temporal contextual information. To this end, a weight function is considered which is based on changes in the group user's preferences over time to increase prediction accuracy of collaborative filtering prediction algorithm. In [20], a model-based method is proposed to track the time changing behavior throughout the life span of the data. To this end, two leading collaborative filtering approaches are considered to exploit the relevant components of all data instances, while discarding only what is modeled as being irrelevant.

III. PROPOSED METHOD

In this section, a social recommender system is proposed which is called Social Recommender based on Temporal Clustering (in short SRTC). The proposed method consists of three main steps including 1) network construction, 2) temporal user clustering, and 3) recommendation. Fig. 1 shows the overview of the proposed method. In addition, the details of the steps of the proposed method are discussed in the following subsections.

A. *Network Construction*

In this step, a network is modeled for the users of the system. To this end, two different views of information are considered, including user-item rating matrix and trust relations between the users. The user-item rating matrix is used to define a temporal similarity function between the users. Let U and I be the users and items sets, respectively. To calculate the temporal similarity values, all of the ratings provided by the users in the user-item matrix are transformed into two groups including 'liked' and 'disliked'. For this purpose, we can use a threshold value for the ratings. For example, in a system with 5 point views of ratings, the transformation can be obtained by considering the ratings higher than 3 as 'liked' and the remaining ratings as 'disliked'. Then, the user-item matrix can be represented as 'liked' and 'disliked' time-weighted matrixes as follows:

$$L_{u,i} = \begin{cases} e^{-\lambda\times(TL-t)} & \text{if user } u \text{ likes item } i \\ 0 & otherwise \end{cases} \tag{1}$$

where L is the 'liked' time-weighted matrix, $u = 1, \dots, |U|$ and $i = 1, \dots, |I|$ are respectively the indexes of the user u and item i, t is the timestamp of rating provided by user u for item i, TL is the maximum value of timestamps in the system, and λ is a parameter to control the effect of time weights. On the other hand, the 'disliked' time-weighted matrix can be represented as follows:

$$D_{u,i} = \begin{cases} e^{-\lambda\times(TL-t)} & \text{if user } u \text{ dislikes item } i \\ 0 & otherwise \end{cases} \tag{2}$$

where D is the 'disliked' time-weighted matrix which is constructed based on the 'disliked' ratings provided by the users in the user-item matrix.

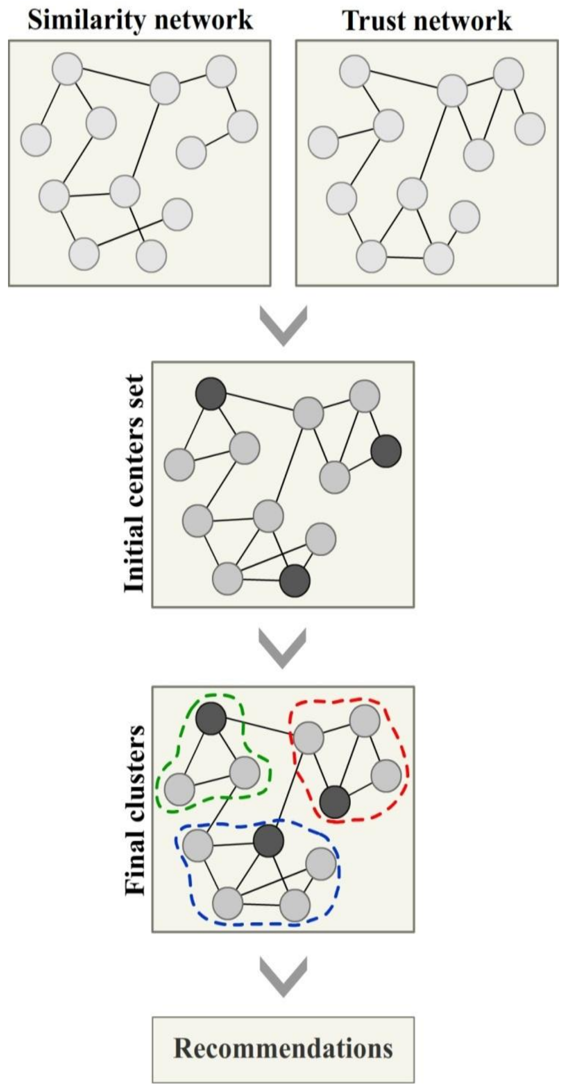

Fig. 1. The overview of the proposed method.

Moreover, the temporal similarity values between the users can be calculated based on the 'liked' and 'disliked' time-weighted matrixes as follows:

$$S_{u,v} = \sum_{i=1}^{|I|} A_i \tag{3}$$

where $S_{u,v}$ is the temporal similarity value between users u and v, and A_i is calculated as follows:

$$A_i = \begin{cases} L_{u,i} \times L_{v,i} & if\ L_{u,i} \neq 0\ and\ L_{v,i} \neq 0 \\ D_{u,i} \times D_{v,i} & if\ D_{u,i} \neq 0\ and\ D_{v,i} \neq 0 \\ -(L_{u,i} \times D_{v,i}) & if\ L_{u,i} \neq 0\ and\ D_{v,i} \neq 0 \\ -(D_{u,i} \times L_{v,i}) & if\ D_{u,i} \neq 0\ and\ L_{v,i} \neq 0 \\ 0 & otherwise \end{cases} \tag{4}$$

where L and D are the time-weighted matrixes for 'liked' and 'disliked' ratings which are calculated using Eqs. (1) and (2), respectively.

On the other hand, the trust relations between the users are used as the social information to construct the users' network. To this end, the explicit trust relations are considered which have been expressed by the users, explicitly. A trust relation between two users indicates that they have similar interests about the items. The trust values can be calculated as follows [21]:

$$T_{u,v} = \frac{d_{max} - d_{u,v} + 1}{d_{max}} \tag{5}$$

where $d_{u,v}$ is the trust propagation distance between the users u and v, and d_{max} is the maximum allowable propagation distance between the users which can be calculated as follows [22]:

$$d_{max} = \frac{\ln(n)}{\ln(k)} \tag{6}$$

where n and k are respectively the size and the average degree of the trust networks in a social recommender system.

Finally, the similarity and trust values between the users are combined to calculate the final similarity weights of the users' network as follows:

$$w_{u,v} = \begin{cases} \frac{2 \times S_{u,v} \times T_{u,v}}{S_{u,v} + T_{u,v}} & if\ S_{u,v} > 0\ and\ T_{u,v} > 0 \\ T_{u,v} & if\ S_{u,v} \leq 0\ and\ T_{u,v} > 0 \\ S_{u,v} & if\ S_{u,v} > 0\ and\ T_{u,v} \leq 0 \\ 0 & otherwise \end{cases} \tag{7}$$

A network can be constructed based on the similarity weights between the users. The network is represented as a graph $G = (V, E, W)$ which V is the set of users in the system, E is the edges between the users in the network, and W is the weights of the edges calculated by Eq. (7). The constructed network will be used in the next step to apply a clustering approach for the users.

B. Temporal User Clustering

In this step, a temporal clustering method is proposed to cluster the users into appropriate groups. The method consists of two phases. In the first phase, a graph-based algorithm [23] is applied on the users' network (i.e. graph G) to identify the initial centers set of the clustering method. The algorithm is based on calculating the graph density. The density of a subgraph $S \subseteq V$ is calculated as follows:

$$\rho(S) = \frac{\sum_{e \in E(S)} w_e}{|S|} \tag{8}$$

where $E(S)$ denotes the edges set of the subgraph S and $w_e \in W$ is the weight of edge e. Then, the candidate nodes $\tilde{A}(S)$ to remove from the graph are identified based on their weighted degrees using a threshold value. The weighted degree of node $i \in S$ can be calculated as follows:

$$wd_S(i) = \sum_{e_{ij} \in E(S)} w_{e_{ij}} \tag{9}$$

Then, the algorithm proceeds on the remaining nodes if the resulted subgraph is non-empty. Moreover, it guarantees that the final subgraph contains at least k nodes which can be considered as the initial centers set. The main idea behind this algorithm is to find an initial centers set which its nodes have maximum distances together. The pseudo code of the algorithm for finding initial centers set is represented in Algorithm 1.

Algorithm 1. Finding initial centers set.

Inputs: $\boldsymbol{G = (V, E, W)}$, $\boldsymbol{k > 0}$, $\boldsymbol{\varepsilon > 0}$.
Output: Initial centers set ($\tilde{\boldsymbol{S}}$).
Set $\boldsymbol{S = V}$ and $\tilde{\boldsymbol{S}} = \boldsymbol{V}$;
while $\boldsymbol{S} \neq \emptyset$ do
    Calculate the density of $\boldsymbol{S}$ (i.e. $\boldsymbol{\rho(S)}$) using Eq. (8);
    Set $\tilde{\boldsymbol{A}}(\boldsymbol{S}) = \emptyset$;
    for all $\boldsymbol{i \in S}$ do
        Calculate the weighted degree of node $\boldsymbol{i}$ (i.e. $\boldsymbol{wd_S(i)}$) using Eq. (9);
        if $\boldsymbol{wd_S(i) \geq (2 + 2\varepsilon) * \rho(S)}$ then
            $\tilde{\boldsymbol{A}}(\boldsymbol{S}) = \tilde{\boldsymbol{A}}(\boldsymbol{S}) \cup \{\boldsymbol{i}\}$;
        end if
    end for
    Sort all $\boldsymbol{i} \in \tilde{\boldsymbol{A}}(\boldsymbol{S})$ descending based on their $\boldsymbol{wd_S(i)}$;
    Set $\boldsymbol{r} = \frac{\boldsymbol{\varepsilon}}{\boldsymbol{1+\varepsilon}} \times |\tilde{\boldsymbol{A}}(\boldsymbol{S})|$;
    Select $\boldsymbol{top\_r}$ nodes from $\tilde{\boldsymbol{A}}(\boldsymbol{S})$ as $\boldsymbol{A(S)}$;
    Set $\boldsymbol{S = S - A(S)}$;
    if $|\boldsymbol{S}| \geq \boldsymbol{k}$ and $\boldsymbol{\rho(S)} < \boldsymbol{\rho}(\tilde{\boldsymbol{S}})$ then
        $\tilde{\boldsymbol{S}} = \boldsymbol{S}$;
    end if
end while
return $\tilde{\boldsymbol{S}}$;

An iterative process is applied on the initial centers set to find final clusters as the second phase of the temporal clustering algorithm. To this end, each user is assigned to the nearest cluster center based on the initial centers set (i.e. $\tilde{S}$). Then, new center set is obtained by using an iterative process based on lines 5-6 of Algorithm 2. Moreover, the clusters whose associated members are less than a threshold value (i.e. m) will be merged with the other clusters based on lines 7-11 of Algorithm 2. This merging process is necessary because the clusters with a small number of users may lead to reduce the accuracy of rating prediction process. Finally, the obtained clusters are used as final clusters of the users. The pseudo code of the clustering algorithm is represented in Algorithm 2.

Algorithm 2. Finding final clusters.

Inputs: $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{W})$, $\tilde{\boldsymbol{S}}$, and $\boldsymbol{m} > \boldsymbol{0}$.
Output: Users' clusters ($\boldsymbol{C}$).
1: *Set* $\boldsymbol{k}' = |\tilde{\boldsymbol{S}}|$;
2: *Set* $\boldsymbol{p_j} = \tilde{\boldsymbol{S}}_{\boldsymbol{j}}, \forall \boldsymbol{j} = \boldsymbol{1}, \dots, \boldsymbol{k}'$;
3: *Let* $\boldsymbol{p_j}, \forall \boldsymbol{j} = \boldsymbol{1}, \dots, \boldsymbol{k}'$ be initial center corresponding to $\mathbf{j}$-th cluster $\boldsymbol{C_j}$;
4: *Associate* each non-selected user to nearest cluster;
5: *Select* new centers $\boldsymbol{p}'_{\boldsymbol{j}} = \boldsymbol{arg\,max}_{v_i \in C_j} \boldsymbol{sum}(\boldsymbol{v_i})$, $\boldsymbol{j} = \boldsymbol{1}, \dots, \boldsymbol{k}'$, and $\boldsymbol{sum}(\boldsymbol{v_i}) = \sum_{v_t \in C_j, v_t \neq v_i} \boldsymbol{w}(\boldsymbol{v_i}, \boldsymbol{v_t})$;
6: if $\boldsymbol{p_j} = \boldsymbol{p}'_{\boldsymbol{j}}, \forall \boldsymbol{j} = \boldsymbol{1}, \dots, \boldsymbol{k}'$ then go to line 7, else $\boldsymbol{p_j} = \boldsymbol{p}'_{\boldsymbol{j}}, \forall \boldsymbol{j} = \boldsymbol{1}, \dots, \boldsymbol{k}'$ and go to line 4;
7: for all $\boldsymbol{C_j}, \boldsymbol{j} = \boldsymbol{1}, \dots, \boldsymbol{k}'$ do
8:    if $|\boldsymbol{C_j}| < \boldsymbol{m}$ then
9:       *Merge* the members of $\boldsymbol{C_j}$ to other clusters;
10:    end if
11: end for
12: for all users $\boldsymbol{u} \in \boldsymbol{V}$ do
13:    *Let* $\boldsymbol{C_u}$ be the cluster that user $\boldsymbol{u}$ belong to;
14: end for
15: return $\boldsymbol{C}$;

C. Recommendation

In this section, the users' clusters are used to predict unseen items for the users in the recommendation process. To this end, the users who are in the target user's cluster are considered as the neighbors set. Therefore, the rating of unseen item i for the target user u can be calculated as follows:

$$P_{u,i} = \bar{r_u} + \frac{\sum_{v \in C_u} w_{u,v}(r_{v,i} - \bar{r_v})}{\sum_{v \in C_u} |w_{u,v}|} \tag{10}$$

where $\bar{r_u}$ is the average of ratings provided by user u, C_u is the cluster that the user u belongs to, $w_{u,v}$ denotes the similarity weight between the users u and v which is calculated using Eq. (7), and $r_{v,i}$ is the rating of item i provided by user v.

IV. Experimental results

In this section, the proposed method (i.e. SRTC) is compared with the other recommender methods to evaluate the quality of predictions based on evaluation metrics. To this end, three clustering-based recommender methods including K-means collaborative filtering (KMCF), trust-aware clustering collaborative filtering (TRACCF) [12], and multi-view K-medoids (MV) [13] are used in the performed experiments. In addition, three time-based recommender methods including collaborative filtering with temporal contextual information (CFTCI) [19], adaptive time-based collaborative filtering (ATCF) [18], and time-based singular value decomposition (timeSVD++) [20] are used to evaluate the effectiveness of the proposed method in comparison to time-based methods. The details of the experiments are presented in the following subsections.

A. Dataset

In the experiments, Epinions[1] dataset is used to verify the effectiveness of the proposed method in comparison with the other recommender methods. In this dataset, the opinions of the users about existing items are used as numerical ratings in the range of 1 (min) to 5 (max). This dataset contains 49,290 users and 139,738 items. Moreover, the trust relations among the users are used as social information in Epinions dataset which the values of them are 0 or 1. Each rating has a timestamp which shows the time of providing the rating by a user for a specific item. All of the ratings in this dataset are transformed into two binary forms including 'liked' and 'disliked' ratings to perform the proposed method. To this end, each rating higher than 3 is labeled as 'liked', otherwise is labeled as 'disliked'.

B. Evaluation measures

Three evaluation metrics including mean absolute error (MAE), root mean squared error (RMSE) and catalog coverage (CC) are used in the experiments to evaluate the performance of the proposed method in comparison with the other methods. MAE and RMSE are two accuracy-based metrics to measure the quality of the predicted ratings by a recommender system. To calculate these metrics, the predicted ratings can be compared with the real ratings and their differences taken into account as the prediction error. Therefore, these metrics are calculated as follows:

$$MAE = \sum_{i=1}^{n} \frac{|r_i - p_i|}{n} \tag{11}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(r_i - p_i)^2} \tag{12}$$

where r_i is the real rating and p_i is the predicted rating for item i, and n indicates the total number of ratings predicted by using a recommender method. It should be noted that, the lower values of MAE and RMSE metrics indicate the higher performance for a method.

Catalog coverage (CC) is another metric which is used in the experiments. This metric calculates the percentage of distinct items in the top-*N* recommendation lists of users, which is calculated as follows:

$$CC = \frac{I_r}{n} \tag{13}$$

where I_r is the total number of distinct items in the recommendations list of users.

C. Results

In this section, several experiments are performed to evaluate the performance of the proposed method. To this end, 80% of ratings per each user are randomly selected as training set and the remaining data is used as test set. Moreover, the values of parameters k and ε in Algorithm 1 and also parameter m in Algorithm 2 are set to $k = 5, \varepsilon = 1$ and $m = 40$ as default values which give good results in general. The experiments are performed based on two types

[1] http://www.trustlet.org/datasets/download_epinions

of users including *All users* and *Cold start users* (the users who have less than 5 ratings).

The results of experiments based on MAE, RMSE, and CC metrics are reported in Tables 1 for *All users* view. As you can see from these results, the proposed method significantly outperforms other recommender methods based on all of the used evaluation metrics. Therefore, it can be concluded that, incorporating temporal information into clustering-based methods leads to improve the performance of the social recommender systems in terms of the accuracy and coverage metrics. The experiments are repeated for the *Cold start users* and the results are reported in Table 2. The results indicate that the proposed method obtains better performance than the other time-based and clustering-based recommender methods based on all of the used evaluation metrics. Therefore, the proposed method can alleviate cold start problem for the social recommender systems in comparison with the other methods.

TABLE I. EXPERIMENT RESULTS ON EPINIONS DATASET BASED ON ALL USERS. THE TOP-PERFORMING ALGORITHM IS DENOTED BY **BOLD** FONT.

Methods	Evaluation Measures		
	MAE	*RMSE*	*CC*
KMCF	0.952	1.295	0.318
TRACCF	0.931	1.202	0.385
MV	0.905	1.153	0.413
CFTCI	0.895	1.197	0.265
ATCF	0.927	1.234	0.284
timeSVD++	0.983	1.323	0.359
SRTC	**0.751**	**0.986**	**0.458**

TABLE II. EXPERIMENT RESULTS ON EPINIONS DATASET BASED ON COLD START USERS. THE TOP-PERFORMING ALGORITHM IS DENOTED BY **BOLD** FONT.

Methods	Evaluation Measures		
	MAE	*RMSE*	*CC*
KMCF	1.115	1.412	0.264
TRACCF	1.024	1.311	0.312
MV	0.973	1.238	0.364
CFTCI	0.958	1.287	0.182
ATCF	0.996	1.374	0.197
timeSVD++	1.139	1.496	0.298
SRTC	**0.867**	**1.112**	**0.395**

Parameter λ is used in Eqs. (1) and (2) to control the effect of rating timestamp in calculating time-weighted ‘liked’ and ‘disliked’ matrixes. The effect of different values of λ is evaluated on the performance of the proposed method in terms of MAE, RMSE and CC metrics. Figs. 2 and 3 show the effect of different values of λ based on these metrics for *All users* and *Cold start users*, respectively. As it can be seen from these results, increasing this parameter leads to increase in the values of MAE and RMSE metrics. Moreover, the values of CC metric decrease when the value of parameter λ is increased for both of the *All users* and *Cold start users*. Therefore, increasing the value of λ has a negative effect on the accuracy and coverage metrics of the recommendation process. Therefore, the value of parameter λ is set to λ=0.001 in the experiments.

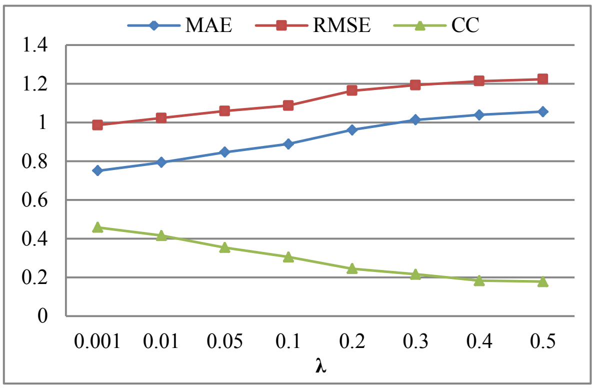

Fig. 2. The effect of different values of parameter λ on MAE, RMSE and CC metrics for *All users*.

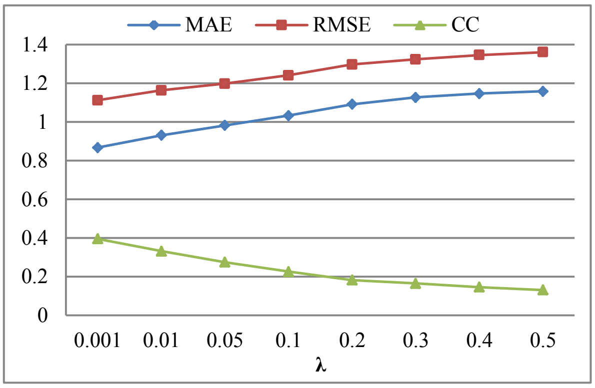

Fig. 3. The effect of different values of parameter λ on MAE, RMSE and CC metrics for *Cold start users*.

V. CONCLUSION

In this paper, a social recommender system is proposed based on a temporal clustering approach to incorporate the effects of time of ratings provided by the users. For this purpose, a temporal similarity measure is introduced, which assigns higher weights to recent ratings than the old ones. Moreover, a users’ network is constructed using the temporal similarity values and social relations between the users. Then, the users’ network is used to group users into appropriate clusters by the temporal clustering approach. To this end, a graph-based method is applied to find initial centers set, automatically. Finally, an iterative process is considered to find final clusters and use them as the neighborhoods in the recommendation process. Experimental results using a benchmark dataset show that the proposed method outperforms other recommender methods in terms of the accuracy and coverage metrics.

ACKNOWLEDGMENT

Mahdi Jalili is supported by Australian Research Council through project No. DP170102303. Sajad Ahmadian would like to thank RMIT University for hospitality during his research visit at RMIT. This research has been mainly conducted during his visit to RMIT University.

REFERENCES

[1] W. Lihua, L. Lu, L. Jing, and L. Zongyong, "Modeling user multiple interests by an improved GCS approach", Expert Systems with Applications, vol. 29, pp. 757-767, 2005.

[2] M. Eirinaki, J. Gao, I. Varlamis, and K. Tserpes, "Recommender Systems for Large-Scale Social Networks: A review of challenges and solutions", Future Generation Computer Systems, vol. 78, pp. 413-418, 2018.

[3] L. U, Y. Chai, and J. Chen, "Improved personalized recommendation based on user attributes clustering and score matrix filling", Computer Standards & Interfaces, vol. 57, pp. 59-67, 2018.

[4] A. Javari, J. Gharibshah, and M. Jalili, "Recommender systems based on collaborative filtering and resource allocation", Social Network Analysis and Mining, vol. 4, pp. 1-11, 2014.

[5] R. Forsati, M. R. Meybodi, "Effective page recommendation algorithms based on distributed learning automata and weighted association rules", Expert Systems with Applications, vol. 37, pp. 1316–1330, 2010.

[6] A. Javari, M. Jalili, "A probabilistic model to resolve diversity–accuracy challenge of recommendation systems", Knowledge and Information Systems, vol. 44, pp. 609-627, 2015.

[7] F. Qian, S. Zhao, J. Tang, and Y. Zhang, "SoRS: Social recommendation using global rating reputation and local rating similarity", Physica A: Statistical Mechanics and its Applications, vol. 461, pp. 61-72, 2016.

[8] J. Li, C. Chen, H. Chen, and C. Tong, "Towards Context-aware Social Recommendation via Individual Trust", Knowledge-Based Systems, vol. 127, pp. 58-66, 2017.

[9] F. Rezaeimehr, P. Moradi, S. Ahmadian, N. N. Qader, and M. Jalili, "TCARS: Time- and Community-Aware Recommendation System", Future Generation Computer Systems, vol. 78, pp. 419-429, 2018.

[10] P. Moradi, S. Ahmadian, "A reliability-based recommendation method to improve trust-aware recommender systems", Expert Systems with Applications, vol. 42, pp. 7386-7398, 2015.

[11] D. H. Alahmadi, X. J. Zeng, "ISTS: Implicit social trust and sentiment based approach to recommender systems", Expert Systems with Applications, vol. 42, pp. 8840-8849, 2015.

[12] C. Birtolo, D. Ronca, "Advances in Clustering Collaborative Filtering by means of Fuzzy C-means and trust", Expert Systems with Applications, vol. 40, pp. 6997-7009, 2013.

[13] G. Guo, J. Zhang, and N. Y. Smith, "Leveraging multiviews of trust and similarity to enhance clustering-based recommender systems", Knowledge-Based Systems, vol. 74, pp. 14-27, 2015.

[14] P. Moradi, S. Ahmadian, and F. Akhlaghian, "An effective trust-based recommendation method using a novel graph clustering algorithm", Physica A: Statistical Mechanics and its Applications, vol. 436, pp. 462-481, 2015.

[15] S. Ahmadian, M. Meghdadi, and M. Afsharchi, "A social recommendation method based on an adaptive neighbor selection mechanism", Information Processing & Management, vol. 54, pp. 707-725, 2018.

[16] N. N. Liu, M. Zhao, E. Xiang, and Q. Yang, "Online evolutionary collaborative filtering", in Proceedings of the fourth ACM conference on Recommender systems, pp. 95-102, 2010.

[17] Y. Ren, M. Tomko, F. D. Salim, J. Chan, C. L. A. Clarke, and M. Sanderson, "A Location-Query-Browse Graph for Contextual Recommendation", IEEE Transactions on Knowledge and Data Engineering, vol. 30, pp. 204-118, 2018.

[18] R. Rafeh, A. Bahrehmand, "An adaptive approach to dealing with unstable behaviour of users in collaborative filtering systems", Journal of Information Science, vol. 38, pp. 205–221, 2012.

[19] B. Karahodza, H. Supic, and D. Donko, "An approach to design of time-aware recommender system based on changes in group user's preferences", in International Symposium on Telecommunications (BIHTEL), Sarajevo, Bosnia and Herzegovina, 2014.

[20] Y. Koren, "Collaborative filtering with temporal dynamics", Communications of the ACM, vol. 53, pp. 89–97, 2010.

[21] P. Massa, P. Avesani, "Trust-aware recommender systems", RecSys '07 Proceedings of the 2007 ACM conference on Recommender systems, pp. 17–24, 2007.

[22] W. Yuan, D. Guan, Y. K. Lee, S. Lee, and S. J. Hur, "Improved trust-aware recommender system using small-worldness of trust networks", Knowledge-Based Systems, vol. 23, pp. 232–238, 2010.

[23] B. Bahmani, R. Kumar, and S. Vassilvitskii, "Densest subgraph in streaming and mapreduce", In 38th international conference on very large data bases, vol. 5, pp. 454–465, 2012.