Available online at www.garmian.edu.krd

Passer Journal

Passer 4 (Issue 2) (2022) 180-187

http://passer.garmian.edu.krd/

A Temporal and Social Network-based Recommender using Graph Clustering

Nawroz Fadhil Ahmed[1,2], Karwan Mohammad Hamakarim[3], Zana Azeez Kakarash[2,4,*]

[1]*Department of Information Technology, Kurdistan Technical Institute, Sulaymaniyah, Kurdistan Region, Iraq*

[2]*Department of Engineering, Faculty of Engineering and Computer Science, Qaiwan International University, Sulaymaniyah, Iraq*

[3]*Department of Information Technology, College of Science and Technology, University of Human Development, Kurdistan Region, Iraq*

[4]*Department of Computer Science, Kurdistan Technical Institute, Sulaymaniyah, Kurdistan Region, Iraq*

Received 29 May 2022; revised 22 September 2022;
accepted 22 September 2022; available online 26 September 2022

DOI: 10.24271/PSR.2022.344758.1134

ABSTRACT

Recommendation Systems (RSs) have significant applications in many industrial systems. The duty of a recommender algorithm is to operate available data (users/items contextual data and rating (or purchase) the consumption history for items), as well as to provide a recommendation list for any target user. The recommended items should be selected so that the target user is compelled to give them positive reviews. In this manuscript, we propose a novel of RS algorithm that makes advantage of user-user trust relationships, rating histories, and their frequency of occurrence. We also provide a brand new overlapping community detection algorithm. The information about the users' community structure is used to handle the cold-start and sparsity problems. We compare the performance of the proposed RS algorithm with a number of state-of-the-art algorithms on the extended Epinions dataset, which has both information on trust relations and the timing of the ratings. Numerical simulations reveal the superiority of the proposed algorithm over others. We also investigate how the algorithms perform when only cold-start users and items are considered. As a cold-start user (item) we consider those that have made (received) less than five ratings. The experiments show significant outperformance of the proposed algorithm over others, which is mainly due to the use of information on overlapping community structures between users.

Keywords: Recommendation systems, social networking, science of networking, overlapping community structure, trust relations, interaction time.

1. Introduction

Big data mining has become one of the major topics in various research areas including social sciences. A large volume of digital data is often produced by users when they purchase (or rate) items. Such information along with users/items contextual data can be used to obtain users-items interactions, model users' preferences and make proper item recommendation for them. Recommendation Systems (RSs) have been developed to perform such tasks[1]. RSs have always been one of the core computer science topics in recent years, which is mainly due to availability of large-scale social networks data, and the industry need. RSs have various applications in both academia and industry and that have been effectively applied in variety of domains such as recommended video[2], webpage[3], friend[4,5], mobile applications[6], OLAP sessions[7], travel package[8], tourism[9], e-commerce[10], B2C controls[11], business process models[12], job[13], rout[14] and learning objects[15].

* Corresponding author
E-mail address: zana.azeez.k@gmail.com (Instructor).
Peer-reviewed under the responsibility of the University of Garmian.

One can classify RSs algorithms depending on the data used in their design. The algorithms that use solely users /or items contextual information are called content-based RSs, while those that are based on users-items interaction data (i.e., purchase (or rating) history of users on items) are called Collaborative Filtering (CF). There are also hybrid RSs algorithms that use both data types[16]. In a variety of cases, users-items interaction data is the only available information, and thus CF algorithms are the only choices. These algorithms are divided into two types which are model-based and memory-based types. Model-based CF algorithms, such as those based on matrix factorization[17, 18] and singular value decomposition, in the first it consider a proper model for the data, and then use a learning algorithm to train the unknown parameters of the model. Memory-based methods, such as user- or item-based CF, seeks to find specific patters in the interaction data by identifying prference of users /or similarity between items[19]. For example, in user-based CF recommendation, a set of the most similar users is first extracted for any target user. Then, the recommendation list is constructed among the items purchased by these similar users. This approach

works on the assumption that similar users have similar tastes and preferences, and a target user likely provides positive ratings to the items that have already positively rated by his/her similar users. The same argument can also be made on the items[20,51,52]; item-based CF algorithms assume that the similar items are likely to be rated similarity (positive or negative) by users.

However, CF recommendation algorithms have been successfully applied to numerous of applications and are indeed the most widely used RSs in the industry, they faced a number drawbacks including rating sparsely, cold-start, scalability, and efficiency[21,54]. The lacking and cold-start problem happens for new users or items for which there are not enough rating history, and thus similarity the test is either impossible or the results are in unreliable scores. ther number of techniques have been used in the literature to handle these issues. Possible approaches to deal with the cold-start problem can be to use social network structure and community structure between users (or items). One can also use other information such as contextual data, friendship, membership and social trust relationships. Moreover, the trust-based CF[22-24] is to demonstrate trust between users. In trust-based algorithms, trustworthy of users is also used in how to consider their rating history in the computations (i.e., the higher the trustworthy of a user to a target user, the higher the influence of ratings of that user in the recommendation list). Purchase (or rating) time is another important factor that, if available, can be used to improve efficiency of recommendation lists. Traditional RSs ignored this type of factor and treat all the ratings equally, regardless of their occurrence time. However, users might change their preferences over time. This can also happen for items, where they might change popularity over time. For example, a movie that was popular some time ago might not still be has the same popularity. Incorporating the rating times in the design of RSs have been considered in a number of works.

Recently, several algorithms have been proposed to consider time or trust in their computations using community detection methods to discover variations of users' interest[23-27, 53]. Using the community detection methods can help RSs to enhance the accuracy of the algorithms and better deal with the cold-start and data sparsity problems. In our previous work (called DGCTARS)[26], we showed that employing community detection methods improves the performance of trust-aware recommender systems. Also, in[27,55] the authors proposed an overlapping community detection method called TOTAR to enhance the results of collaborating filtering systems. In this paper we consider both trust relations and occurrence times of the ratings to design RSs with high precision. The proposed recommendation algorithm (called TATRS) first construct user-user similarity graph. Then, it identifies the sparsest subgraph and obtains community structure of users, for which time and trust information are used. The final step is to make the recommendation[56]. The novelty of the proposed method over state-of-the-art recommenders such as TOTAR and DGCTARS are as follows:

- The proposed method uses a novel overlapping community detection method in its discovery process, in a way that DGCTARS uses non-overlapping graph clustering method to create group the users. In overlapping community detection, a user may belong to more than one community and thus more relevant users are utilized in prediction process.
- The proposed method considers both time of ratings (as a rating context) and trust statements in its recommendation process, while DGCTARS does not employ the changes in users' interests over different periods of time.
- The suggested method employs an imputation method integrating with a reliability measure for estimating the missing ratings. This method is used for overcoming the data- sparsity problem by reducing the sparsity of the user-item rating matrix.
- The proposed method uses a subgraph finding algorithm to identify cluster seeds. Using this algorithm, cluster seeds are identified in dense regions of the graphs, which leads form high qualify communities. The aim of the subgraph finding algorithm is to find a set of graph nodes that are located in high dense regions of the graph and are as far apart with each other as possible. Using this seeding strategy leads to producing cohesive clusters and identifying ground-truth communities.
- The proposed method automatically identifies the numeral of communities, while TOTAR needs predefining the number of communities.

The is to mention that the proposed method has been applied on a benchmark dataset and compared its performance with state-of-the-art recommenders. Our simulations results reveal effectiveness of the proposed algorithm.

The paper is organized as follows.
The Section2 is deals with Research Related Work for this study. Proposed Algorithm mentioned in Section3. The experiments are explained in section4. While The conclusions of the study is indicated in the last section.

2. Related Work

2.1 Trust-aware Recommendations

Trust-based RSs have been recently introduced to the community of computer science. In trust-based RS algorithms, the ratings of socially connected users to the target user are also considered in the recommendation process. Recent studies have shown that incorporating social factors or trust statements in RSs leads to improve the recommendation quality. Previously, several trust-based CF approaches have been proposed to overcome sparsity data and cold-start problems as well as to increase recommendable items[22, 26, 28-30,57]. Trust statements can be explicitly collected from users or can be implicitly inferred from users' behaviors (such as rating information).

In implicit methods, a trust network is constructed for each user based on her/his ratings in the system. For example, Hwang and Chen introduced a method to directly derive the trust relations from users' rating data[31]. Liu and Lee suggested a specific approach which does not directly use the trust information; instead the number of exchanged messages among the users of the system are take into account to construct the trust network[32].

Alahmadi and Zeng presented a framework to apply short texts posted by users' friends in microblogs as an additional data source to build the trust network[33]. According to common sense the closer two ratings are, the most significant value of that rating, and the more trustworthy the user will be. On the other hand, explicit methods propose the trust statements from pre-established social connections among users of a social network. While the explicit trust statements are directly specified by the users, they are more accurate and reliable than implicit ones in determining social relationships among users[34-37].

some of the recent works are proposed to consider users' online social networks is to designing trust-aware recommender systems. In[38] it has been demonstrate that a user can constructs his/her social connections with someone who has similar tastes. Gharibshah and Jalili have conducted a study of the relation between recommender systems and connectedness of users-items bipartite interaction network[39]. Guo et al. proposed a method which merged the ratings of users' trusted neighbors with the other information sources to identify their preferences[40]. Moradi and Ahmadian suggested a framework to combine both implicit and explicit trust statements[28]. They used reliability measures to evaluate how their implicit trusts are correctly estimated. In[26] a graph clustering method was used to create group users based on their trust network. Jiang et al. introduced a framework to incorporate interpersonal influences of users in social network with their personal preferences to improve the accuracy of social recommendation[42,53].

2.2 Time-aware Recommendations

Purchase (or rating) time is one of the most important contextual information that can be used, if available, to design RSs with high precision[43]. The main motivation for time-aware recommenders is that in real-world applications the interests of users may change through the time. The most traditional recommender algorithms are exploits the data as static and not to consider this issue into account. Both memory-based and model-based CF approaches have been recently adopted to take into account user's drifts. In time window algorithms only a number of the most recent ratings are considered in the recommendation process. In time decay-based methods, temporal effect of the data is handled by boosting recent ratings and penalizing the older ratings. For example,[44] used a decay function to compute the importance and usefulness of both the tag and time of the interaction. Model-based CF methods have also been adopted with temporal effects. For instance, Randle et al. proposed a matrix factorization method to include temporal effects in RSs[45]. Daneshmand et al. proposed a model-based time-aware recommender system using statistical diffusion to model dependencies between items[46].

3. Proposal Method

In this paper, a time- and trust-aware recommendation algorithm is proposed. The algorithm is based on overlapping community structure of users' network and a novel method is proposed to discover such overlapping communities. The proposed RS method consists of three main phases including (*i*) construction of time-weighted trust network, (*ii*) community detection, and (*iii*) prediction of the ratings. In the first step, a trust network is constructed for the target user based on both temporal information and explicit trust statements. In the second phase, the overlapping community detection method is applied to create group of similar users into clusters. Also, in this angle a novel method is developed to calculate the similarity between users (items) that are placed in the same cluster. Finally, in the third phase, for each unseen item, a rating is predicted as well as a list top-N items is recommended to the target user. We denote the proposed method by Time- And Trust-aware Recommendation System (TATRS).

3.1 Constructing Time-Weighted Trust Network

The aim of this step is to construct users' trust network based on implicit and explicit trust relationships. Implicit trust values can be obtained based on the users' ratings on items. In real-world recommendation methods users rated only a few numbers of items and thus resulted in a sparse user-item matrix. The tackle this issue, the following equation is used to predict unseen items and impute them to the user-item matrix:

$$P_{a,i} = \bar{r}_a + \frac{\sum_{u \in N_{a,i}} sim_{a,u}\left(r_{u,i} - \bar{r}_u\right)}{\sum_{u \in N_{a,i}} sim_{a,u}} \quad (1)$$

where $\bar{r}_a$ is the mean of ratings for the target user a, $N_{a,i}$ is the set of neighbors of the target user, and $sim_{a,u}$ is a similarity value between users a and u that is computed as:

$$sim_{a,u} = \frac{\sum_{i \in A_{u,u}}\left(r_{a,i} - \bar{r}_a\right)\left(r_{u,i} - \bar{r}_u\right)}{\sqrt{\sum_{i \in I_{a,u}}\left(r_{a,i} - \bar{r}_a\right)^2}\sqrt{\sum_{i \in I_{a,u}}\left(r_{u,i} - \bar{r}_u\right)^2}} \quad (2)$$

where $I_{a,u}$ is the set of items commonly rated by both a and u.

As the similarity values are computed using the above equations, we use the method proposed in[28] to obtain the reliability of the prediction, which indeed evaluate the quality of the predicted rating values. If the reliability value for a predicted rating is higher than a threshold value φ, then the predicted rating is added to user-item matrix.

To construct the users' trust network, the (enriched) dataset is divided into several subsets of different periods of time (i.e., $t = \{1,2,\dots,T\}$). The interactions are represented as the time-weighted user-item matrix $W = [w_{ij}]$ using the following equation:

$$w_{ij} = e^{-(T-t)/\theta} \quad if \;\; t > 0; \quad otherwise \; w_{ij} = 0 \quad (3)$$

where $\theta > 0$ is a pre-defined constant. It should be noted that if there is no interaction information, t is set to 0. The following equation is used to obtain the value of the time-weighted rating from user i to item j is obtained using the following equation:

The next step is to obtain the time-weighted similarity values between users, which is extracted from user–item interaction data, as:

$$U_{a,u} = \frac{\sum_{i \in A_{u,u}}\left(tr_{a,i} - \bar{tr}_a\right)\left(tr_{u,i} - \bar{tr}_u\right)}{\sqrt{\sum_{i \in I_{a,u}}\left(tr_{a,i} - \bar{tr}_a\right)^2}\sqrt{\sum_{i \in I_{a,u}}\left(tr_{u,i} - \bar{tr}_u\right)^2}} \quad (4)$$

where $\bar{tr}_u$ is the mean of the time-weighted ratings given by u and computed as:

$$tr_{ij} = r_{ij} \times w_{ij} \quad if \quad r_{ij}, g_{ij} > 0; \; otherwise \; tr_{ij} = 0 \tag{5}$$

The values of U can be further used as implicit trust statements between users. Then, the (dense) weighted connectivity matrix between users is calculated as:

$$H_{u,v} = \gamma \times U_{u,v} + (1-\gamma) \times T_{u,v} \tag{6}$$

where $\gamma \in [0,1]$ and T_{uv} denotes the explicit trust value between users u and v. The time-weighted trust network can be constructed using H in which each user is a node in the graph and H_{uv} indicates the weight value between nodes u and v.

3.2 Overlapping Community Detection

The community detection method proposed here including (1) Identifying initial community centers, (2) Finding community members and (3) Merging communities' steps. The first step, takes the idea of Moradi et al.[26] to find the sparsest subgraph of which the nodes are used as the set of initial cluster seeds for the proposed overlapping community detection algorithm. Thus, the nodes of this subgraph should have maximum dissimilarities with one other. Let's denote the candidate nodes as $\widetilde{A}$, which is obtained using Algorithm 1. For an undirected and unweighted graph $G = (V, E)$ with V as set of nodes and E as set of edges, density of a subgraph S$\subseteq V$ is defined as $\rho(S) = \frac{\sum_{e \epsilon E(S)} W_e}{|S|}$, where w_e is the weight of edge e. Different methods employ different seeding, expanding and merging strategies.

Algorithm 1. The approximate sparsest subgraph finding

Input: $\boldsymbol{G = (V, E), K > 0, and \; r > 0}$.
Output: List of candidate nodes.
Algorithm:
1: $S, \widetilde{\boldsymbol{S}} \leftarrow \boldsymbol{V}$;
2: while $S \neq \emptyset$ do
3: $\widetilde{\boldsymbol{A}}(\boldsymbol{S}) \leftarrow \{\boldsymbol{i \epsilon S | wd_s(i) \geq \theta * \rho(s)}\}$;
4: *Let* $\boldsymbol{A(S)} \subseteq \widetilde{\boldsymbol{A}}(\boldsymbol{S})$, with $|\boldsymbol{A(S)}| = r * |\widetilde{\boldsymbol{A}}(\boldsymbol{S})|$;
5: $\boldsymbol{S \leftarrow S \backslash A(S)}$;
6: *If* $\boldsymbol{|S| \geq k \; and \; \rho(s) < \rho(\tilde{s}) \; then}$
7: $\widetilde{\boldsymbol{S}} \leftarrow S$;
8: end if
9: end while
10: *end for;* 11: *Return* $\widetilde{\boldsymbol{S}}$;

As the initial cluster seeds are obtained, the nodes are iteratively assigned to the cluster centers that maximizes the following equation[47]:

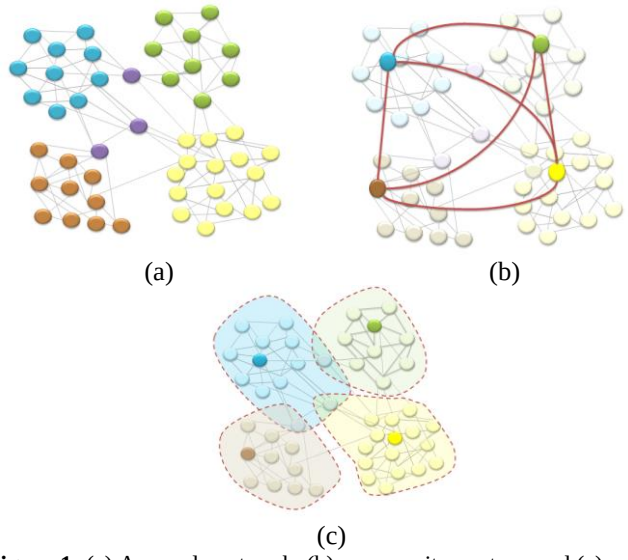
$$f_c = \frac{k_c^{in}}{\left(k_c^{in} + k_c^{out}\right)^{\alpha}} \tag{7}$$

Where α is value in the range of [1,2], k_c^{in} is the total internal degree of c-th community (i.e., the number of links within the community) while, k_c^{out} is the total external degree of this community (i.e., the number of links from nodes located in community c to nodes of other communities). This process is iteratively repeated over all communities until there is no further change in the community members and steady state is achieved. Finally, initial communities are formed. Then those of communities with small number of users are merged with their nearest clusters. The proposed recommendation method is based on collaborative filtering meaning that information of neighboring users is used to identify their taste. The users that are in the same cluster are considered as neighbors. If a user does not have a sufficient number of neighbors, the computation might not be reliable, and thus small clusters should appropriately merge to minimize this effect.

The pseudo-code of the proposed community detection method is presented in Algorithm 2.

Algorithm 2. The proposed overlapping community detection method

Input: Time-weighted user trust network.
Output: List of communities.
Algorithm:
1: *Find* initial community centers $\widetilde{\boldsymbol{S}}$;
2: $\boldsymbol{k'} = |\widetilde{\boldsymbol{S}}|$;
3: *Set* $\boldsymbol{p_j = \tilde{S}_j}, \forall \boldsymbol{j = 1, \dots, k'}$;
4: *Let* $\boldsymbol{p_j}, \forall \boldsymbol{j = 1, \dots, k'}$ be initial center associated with $\boldsymbol{j}$-th cluster $\boldsymbol{C_j}$;
5: *for* each $\boldsymbol{p_j}$ do
6: A= *Identify neighbors* of $\boldsymbol{C_j}$,
7: $\boldsymbol{C_j^{new}}$= Iteratively assign the nodes of A to $\boldsymbol{C_j}$ if maximizes the Eq. (7).
8: *If* $\boldsymbol{C_j^{new} \neq C_j}$ go to line 8;
9: *end for*;
10: *for* all $\boldsymbol{C_j, j = 1, \dots, k'}$ do
11: if $\boldsymbol{|C_j| < m}$ then *Merge* $\boldsymbol{C_j}$ members to the other communities;
12: *end for;* 13: *Return* $\boldsymbol{C}$;



(a) (b) (c)

Figure 1: (a) A sample network, (b) community centers, and (c) identified communities.

Fig. 1 illustrates the proposed community detection method. Fig. 1(a) shows the time-weighted users trust network. The initial founded centers are shown in in Fig. 1(b). Fig. 1(c) shows how the proposed method finds the overlapping communities. Some nodes might have the same membership degree to more than one community, indicating that the communities might overlap.

3.3 Rating Prediction

The aim of this step is to recommend a set of items to the target user such that the user likely provides high ratings to the recommended items. To this end, we use the information of the target user's community, identified in the previous step. The rating made by user u on an unseen item i is predicted using the following equation:

$$p_i(u) = \bar{r}(u) + \frac{\sum_{v \in C_u} H_{u,v}\left(r_i(v) - \bar{r}(v)\right)}{\sum_{v \in C_u} \left| H_{u,v} \right|} \quad (8)$$

where C_u denotes the associated community (or communities) that user u belongs to, $r_i(v)$ is a rating given to item i by user v, $\bar{r}(u)$ is the average rating given by user u and $H_{u,v}$ is a similarity function between users u and v which is calculated using Eq. (6). Finally, we recommend the top-N items based on the predicted rating values.

4. Experiments

A set of experiments are conducted to evaluate the performance of the proposed method. The experiments are performed on extended Epinions dataset, which contains both trust relationships and rating timestamps. This dataset contains 922,267 ratings from 22,166 users on 296,277 products. The suggested method (TATRS) is compared with a number of state-of-the-art methods including Dense Graph Clustering Collaborative Filtering (DGCCF)[26], User-based and Item-based k-means CF (KMCF-U and KMCF-I)[48], Item-based Fuzzy Clustering CF (IFCCF) [48], Trust-Aware Clustering CF (TRACCF)[48], TimeSVD++[45], Time-Adaptive Collaborative Filtering algorithm (TACF)[49] and TOTAR[27].

In this study the users' ratings are separated into different periods of time to evaluate time effects on user's drifts. The ratings are on a scale of 1–5, with 5 being the highest rating. The experiments were carried out on a machine with a core-i7 CPU, 8GB of RAM, and the algorithms were implemented using C# programming language.

4.1 Evaluation Measures

In this paper three evaluation measures are used to report the results. These measures include: Mean Average Error (MAE), Root Mean Square Error (RMSE), and Rate Coverage (RC). MAE referred to mean average error and it is defined as:

$$MAE = \frac{\sum_{i=1}^{N} |r_i - p_i|}{N} \quad (9)$$

where p_i and r_i are the predicated and actual ratings of an item i, respectively, and N denotes the total number of predicted ratings. While, RMSE demonstrates the contributions of the absolute errors between the predicted and actual ratings and it has defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(r_i - p_i)} \quad (10)$$

The RMSE squares the error before summing it and tends to penalize large errors; lower MAE and RMSE values show higher prediction accuracy. RC refers to rate coverage and defined as a fraction of ratings for user is able to predict and is defined as follows:

$$RC = \frac{\#number\ of\ \ predicted\ ratings}{\#number\ of\ \ all\ ratings} \quad (11)$$

4.2 Results on Prediction Performance

Tables 1-7 report the performance of the RS algorithms including the one proposed in this work (TATRS) in terms of their MAE, RMSE and RC. In these experiments, the parameters θ and γ are both set to 0.5. Tables 1, 2 and 3, respectively, show MAE, RMSE and RC of the algorithms for different values of N (N = 5, 10, 15, 20, 25 and 30) in the top-N recommendation task. The proposed RS algorithm show much lower MAE than other algorithms for all value of N except for N = 30. The other methods are not consistent in their behavior; while KMCF-U is the top-performer for N = 30 and second top-performer (after TATRS) for N = 5, it has third rank for N = 10, 20 and the last one for N = 15. RMSE results are also very similar to those of MAE (Table 2), where TATRS is the top-performer with the lowest RMSE for N = 45, 10, 15, 20, while KMCF-U has the best performance for N = 30. The algorithms have mixed performance in terms of RC measure (note that the higher the value of RC, the better the performance of the algorithm). TATRS has the highest RC in two cases (N = 10, 20), while DGCTARS is the top-performed for N = 5, 30 and KMCF-U for N = 15. Table 4 report the average values of MAE, RMSE and RC for N = 1-30. As it is seen, the proposed method is the top-performer on all performance measures.

Table 1: MAE of RS algorithms on the extended Epinions dataset for different values of N is top-N recommendation task (TATRS is the proposed RS algorithm).

Algorithm	Top5	Top10	Top15	Top20	Top30
TATRS	**0.1309**	**0.1583**	**0.1078**	**0.0203**	0.9232
DGCTARS	0.9486	0.9263	0.9778	0.5842	0.9301
IFCCF	0.8186	0.9092	0.7844	0.8565	0.8759
TRACCF	0.8594	0.9062	0.9059	0.9023	1.0527
KMCF-U	0.6523	0.8926	1.2730	0.7462	**0.6445**
KMCF-I	1.0457	0.7518	0.7681	0.9375	0.7333
TOTAR	2.9691	2.8431	1.9382	2.3648	2.1590
TimeSVD++	0.9131	0.9032	0. 8952	0.8924	0.8911
TACF	0.8012	0.7666	0.8500	0.8345	0.8767

Table 2: RMSE of the algorithms for different values of N in the top-N recommendation task.

Algorithm	Top5	Top10	Top15	Top20	Top30
TATRS	**0.1308**	**0.1583**	**0.1078**	**0.2401**	0.9231
DGCTARS	0.9891	0.9005	0.9163	0.9415	0.8863
IFCCF	1.0075	1.1355	0.9751	1.1151	1.1867
TRACCF	1.2551	1.2556	1.2595	1.0588	1.4060
KMCF-U	1.0290	1.1067	1.4845	1.0710	**0.7978**
KMCF-I	1.2618	1.0179	0.9806	1.3012	0.9168
TOTAR	3.4324	3.2962	2.8704	2.9368	2.7639
TimeSVD++	0.8777	0.8991	0.9091	0.8983	0.9191
TACF	0.9011	0.8871	0.8943	0.8454	0.7998

Table 3: RC results the extended Epinions dataset for different values of N in top-N recommendation task.

Algorithm	Top5	Top10	Top15	Top20	Top30
TATRS	0.6364	**0.6343**	0.1078	**0.6603**	0.6624
DGCTARS	**0.6404**	0.6253	0.9163	0.6421	**0.6677**
IFCCF	0.1667	0.1923	0.9751	0.2451	0.1923
TRACCF	0.0661	0.0771	1.2595	0.0585	0.0523
KMCF-U	0.0419	0.0333	**1.4845**	0.0617	0.0404
KMCF-I	0.1795	0.1667	0.9806	0.1702	0.1795
TOTAR	0.2000	0.278	0.3378	0.3674	0.2000
TimeSVD++	0.4553	0.0234	0.6543	0.2311	0.6954
TACF	0.0356	0.5432	0.9632	0.1165	0.0453

Table 4: RMSE, MAE and RC of the algorithms averaged over N in the top-N recommendation task.

Algorithm	RMSE	MAE	RC
TATRS	0.0145	0.0012	0.6453
DGCTARS	0.9415	0.5842	0.6420
IFCCF	1.1151	0.8565	0.2451
TRACCF	1.05884	0.9023	0.0585
KMCF-U	1.0701	0.7462	0.0617
KMCF-I	1.3012	0.9375	0.1702
TOTAR	3.4324	2.9691	0.2000
TimeSVD++	0.90066	0.89995	0.4119
TACF	0.86554	0.8258	0.34076

One of the main reasons to use information on community structure between users is to overcome the cold-start and sparsity problems. Next investigating the performance of the algorithms when cold-start users or items are considered in the prediction process. To this end, four subsets are extracted from the dataset, including (*i*) *Cold start users*: the set of users who have rated less than five items; (*ii*) *Heavy users*: to indicate the users who have provided more than 10 ratings; (*iii*) *Black sheep users* refer to users who are too isolated and have no or little connections with others. This situation makes the recommender system to be inefficient in producing preferences. In this paper, we have predicted that those users who have provided more than four ratings and the difference between their ratings on a specific item (with average rating above one) is greater than one is identified as black sheep users; (IV) *Niche items*: for the items that have received less than five ratings.

Table 5-7 show the performance of the algorithms when the above four subset of users (or items) are considered. As it is seen, the proposed method has the best performance for cold-start users, where its MAE and RMSE is significantly less than the other methods and its RC value is the highest. The power of TATRS in better handling the cold-start users comes from taking into account trust relations and overlapping community information of the users. Not surprisingly, TATRS does not show good performance for heavy users. This is mainly due to having only few fields in this subset of the data. It results in much less MAE and RMSE values for black sheep users and niche items. It has the second top performance in terms of RC in these two cases.

Table 5: MAE of the algorithms over different views of the dataset.

Algorithm	Cold users	Heavy users	Black sheep users	Niche Items
TATRS	**0.0001**	0.0096	**0.0224**	**0.0022**
DGCTARS	0.9628	0.9426	1.0049	1.0322
IFCCF	0.6529	**0.7976**	0.6811	0.9206
TRACCF	0.7724	1.1977	1.0505	1.1791
KMCF-U	1.0058	1.1325	1.2029	1.1038
KMCF-I	1.4872	0.7052	1.1643	0.9759
TOTAR	1.0097	0.9065	0.6342	2.9987
TimeSVD++	0.7245	0.3775	0.5254	0.4000
TACF	0.8435	0.5991	0.3454	1.3480

Table 6: RMSE of the algorithms over different views of the dataset.

Algorithm	Cold users	Heavy users	Black sheep users	Niche Items
TATRS	**0.0001**	0.1175	**0.2480**	**0.0210**
DGCTARS	1.2734	1.2951	1.3297	1.3567
IFCCF	0.9679	**1.0220**	0.9512	1.3663
TRACCF	1.0517	1.5467	1.3518	1.5858
KMCF-U	1.2322	1.5059	1.4407	1.3782
KMCF-I	1.7764	0.8764	1.4794	1.3763
TOTAR	1.1790	1.0075	0.6660	3.1793
TimeSVD++	0.9734	0.68670	0.7583	0.9643
TACF	0.9535	0.9124	1.2554	0.8000

Table 7: RC of the algorithms over different views of the dataset.

Algorithm	Cold users	Heavy users	Black sheep users	Niche Items
TATRS	**0.4951**	0.4925	0.6306	0.6419
DGCTARS	0.5562	0.4727	0.6029	**0.6530**
IFCCF	0.3226	0.2500	**0.2885**	0.2000
TRACCF	0.0865	0.0775	0.5841	0.0686
KMCF-U	0.0508	0.0525	0.067***6***	0.0737
KMCF-I	0.1932	**0.1905**	0.4091	0.1818
TOTAR	0.3655	**0.5787**	0.5356	0.2000
TimeSVD++	0.6235	0.6476	0.7500	0.5400
TACF	0.4875	0.1643	0.6779	0.4652

Table 8: Computational complexity of the recommendation methods. n: number of users, m: number of items, c: number of clusters, i: number of iterations, k: number of latent factors.

Algorithm	Time	Trust	Graph-based	Computational complexity
TATRS	Yes	Yes	Yes	$O(n^2m)$
DGCTARS	No	Yes	Yes	$O(n^2m)$
IFCCF	No	Yes	No	$O(inmc)$
TRACCF	No	Yes	No	$O(inmc + nm)$
KMCF-U	No	Yes	No	$O(inmc + nm)$
KMCF-I	No	Yes	No	$O(inmc + nm)$
TOTAR	Yes	No	Yes	$O(n^2m)$
TimeSVD++	Yes	No	No	$O(nmk + ink + nmk)$
TACF	Yes	No	Yes	$O(n^2m)$

4.3 Sensitivity Analysis

The proposed RS algorithm includes two adjustable parameters (γ and θ), and its performance depends on assigning proper values for these parameters. We next study how the performance of TATRS depends on different values of these two parameters. Figure 2 show the performance in terms of MAE. MAE and RMSE do not show significant changes as γ changes. However, they have the minimum for $\theta = 0.5$. Considering these results, we set $\theta = \gamma = 0.5$ in the experiments reported in the previous section.

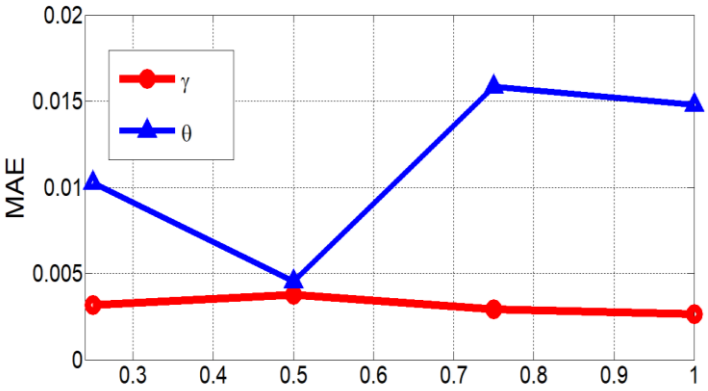


Figure 2: MAE of the proposed RS algorithm (TATRS) as a function of γ and θ parameters.

Conclusion

Recommendation systems include many potential applications in both academia and industry. However, they use information from various sources (e.g., users and/or items contextual data and rating (or purchase) history of users on items) in order to provide efficient item recommendations for any target user. Traditionally, recommendation systems were designed without considering rating times. However, the users might change their taste and/or items might lose (or gain) popularity in different time periods. Indeed, the time period that the rating has been made is an important factor that can be efficiently used in designing recommender algorithms. In this manuscript, we proposed a novel recommender algorithm that uses information on rating times as well as trust relations between users. Moreover, we proposed a novel overlapping community detection algorithm to discover the community structure between users. The information on the community structure is used to improve the accuracy of the recommendations and better deal with the data sparsity and cold-start problems. We applied the proposed algorithm on extended Epinions dataset and compared its performance with a number of state-of-the-art algorithms. The numerical simulations showed that the proposed algorithm resulted in recommendation lists with higher prediction accuracy. Also, the proposed algorithm outperformed others when only cold-start users (or items) were considered in the prediction process.

Conflict of interests

None

References

[1] E. Aslanian, M. Radmanesh, M. Jalili, Hybrid recommender systems based on content feature relationship, IEEE Transactions on Industrial Informatics, DOI (2016).
[2] A. Javari, M. Jalili, Cluster-based collaborative filtering for sign prediction in social networks with positive and negative links, ACM Transactions on Intelligent Systems and Technology (TIST), 5 (2014) 24.
[3] R. Mishra, P. Kumar, B. Bhasker, A web recommendation system considering sequential information, Decision Support Systems, 75 (2015) 1-10.
[4] C.C. Chen, S.-Y. Shih, M. Lee, Who should you follow? Combining learning to rank with social influence for informative friend recommendation, Decision Support Systems, 90 (2016) 33-45.
[5] Z. Zhang, Y. Liu, W. Ding, W. Huang, Q. Su, P. Chen, Proposing a new friend recommendation method, FRUTAI, to enhance social media providers' performance, Decision Support Systems, 79 (2015) 46-54.
[6] C. Xu, D. Peak, V. Prybutok, A customer value, satisfaction, and loyalty perspective of mobile application recommendations, Decision Support Systems, 79 (2015) 171-183.
[7] J. Aligon, E. Gallinucci, M. Golfarelli, P. Marcel, S. Rizzi, A collaborative filtering approach for recommending OLAP sessions, Decision Support Systems, 69 (2015) 20-30.
[8] Q. Liu, E. Chen, H. Xiong, Y. Ge, Z. Li, X. Wu, A Cocktail Approach for Travel Package Recommendation, IEEE Transactions on Knowledge and Data Engineering, 26 (2014) 278-293.
[9] M. Nilashi, O. bin Ibrahim, N. Ithnin, N.H. Sarmin, A multi-criteria collaborative filtering recommender system for the tourism domain using Expectation Maximization (EM) and PCA–ANFIS, Electronic Commerce Research and Applications, 14 (2015) 542-562.
[10] S.-l. Huang, Designing utility-based recommender systems for e-commerce: Evaluation of preference-elicitation methods, Electronic Commerce Research and Applications, 10 (2011) 398-407.
[11] S. Lee, Using data envelopment analysis and decision trees for efficiency analysis and recommendation of B2C controls, Decision Support Systems, 49 (2010) 486-497.
[12] Y. Li, B. Cao, L. Xu, J. Yin, S. Deng, Y. Yin, Z. Wu, An Efficient Recommendation Method for Improving Business Process Modeling, IEEE Transactions on Industrial Informatics, 10 (2014) 502-513.
[13] M. Reusens, W. Lemahieu, B. Baesens, L. Sels, Explicit versus Implicit information for job recommendation: A case study with the Flemish public employment services, Decision Support Systems, DOI http://doi.org/10.1016/j.dss.2017.04.002.
[14] Y.-M. Li, L.-F. Lin, C.-C. Ho, A social route recommender mechanism for store shopping support, Decision Support Systems, 94 (2017) 97-108.
[15] M. Salehi, I.N. Kamalabadi, M.B.G. Ghoushchi, An effective recommendation framework for personal learning environments using a learner preference tree and a GA, IEEE Transactions on Learning Technologies, 6 (2013) 350-363.
[16] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey, Knowledge-Based Systems, 46 (2013) 109-132.
[17] J. Liu, C. Wu, W. Liu, Bayesian Probabilistic Matrix Factorization with Social Relations and Item Contents for recommendation, Decision Support Systems, 55 (2013) 838-850.
[18] M. Ranjbar, P. Moradi, M. Azami, M. Jalili, An imputation-based matrix factorization method for improving accuracy of collaborative filtering systems, Engineering Applications of Artificial Intelligence, 46 (2015) 58-66.
[19] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, Proceedings of the 10th international conference on World Wide Web, ACM, Hong Kong, 2001, pp. 285-295.
[20] A. Nanopoulos, Item Recommendation in Collaborative Tagging Systems, IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 41 (2011) 760-771.
[21] X. Su, T.M. Khoshgoftaar, A survey of collaborative filtering techniques, Advances in artificial intelligence, 2009 (2009) 4.
[22] P. Massa, P. Avesani, Trust-aware recommender systems, 2007 ACM conference on Recommender systems, Minneapolis, Minnesota, USA, 2007, pp. 17-24.
[23] P. Moradi, F. Rezaimehr, S. Ahmadian, M. Jalili, A trust-aware recommender algorithm based on users overlapping community structure, International Conference on Advances in ICT for Emerging Regions, IEEE, 2016, pp. 162-167.
[24] M.M. Azadjalal, P. Moradi, A. Abdollahpouri, M. Jalili, A trust-based recommendation method based on Pareto dominance and confidence concepts, Knowledge-Based System, 116 (2017) 130-143.

[25] Y. Ding, X. Li, Time weight collaborative filtering, Proceedings of the 14th ACM international conference on Information and knowledge management, ACM, 2005, pp. 485-492.
[26] P. Moradi, S. Ahmadian, F. Akhlaghian, An effective trust-based recommendation method using a novel graph clustering algorithm, Physica A: Statistical Mechanics and its Applications, 436 (2015) 462-481.
[27] H. Feng, J. Tian, H.J. Wang, M. Li, Personalized recommendations based on time-weighted overlapping community detection, Information and Management, 25 (2015) 789-800.
[28] P. Moradi, S. Ahmadian, A reliability-based recommendation method to improve trust-aware recommender systems, Expert Systems with Applications, 42 (2015) 7386-7398.
[29] N. Ranjbar Kermany, S.H. Alizadeh, A hybrid multi-criteria recommender system using ontology and neuro-fuzzy techniques, Electronic Commerce Research and Applications, 21 (2017) 50-64.
[30] M. Ghavipour, M.R. Meybodi, An adaptive fuzzy recommender system based on learning automata, Electronic Commerce Research and Applications, 20 (2016) 105-115.
[31] C.-S. Hwang, Y.-P. Chen, Using trust in collaborative filtering recommendation, International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Springer, 2007, pp. 1052-1060.
[32] F. Liu, H. Lee, Use of social network information to enhance collaborative filtering performance, Expert Systems with Applications, 37 (2010) 4772–4778.
[33] D.H. Alahmadi, X.-J. Zeng, ISTS: Implicit social trust and sentiment based approach to recommender systems, Expert Systems with Applications, 42 (2015) 8840-8849.
[34] N. Lathia, S. Hailes, L. Capra, Trust-based collaborative filtering, Proceedings of trust management II, US: Springer, 2008, pp. 119-134.
[35] Y. Manolopoulus, A. Nanopoulus, A.N. Papadopoulus, P. Symeonidis, Collaborative recommender systems: combining effectiveness and efficiency, Expert Systems with Applications, 34 (2008) 2995-3013.
[36] H. Ingoo, J.O. Kyong, H.R. Tae, The collaborative filtering recommendation based on SOM cluster-indexing CBR, Expert Systems with Applications, 25 (2003) 413–423.
[37] J. Cho, K. Kwon, Y. Park, Q-rater: a collaborative reputation system based on source credibility theory, Expert Systems with Applications, 36 (2009) 3751-3760.
[38] A. Abdul-Rahman, S. Hailes, Supporting trust in virtual communities, 33th Hawaii International Conference on System Sciences, Hawaii, USA, 2000.
[39] J. Gharibshah, M. Jalili, Connectedness of users–items networks and recommender systems, Applied Mathematics and Computation, 243 (2014) 578-584.
[40] G. Guo, J. Zhang, D. Thalmann, Merging trust in collaborative filtering to alleviate data sparsity and cold start, Knowledge-Based Systems, 57 (2014) 57-68.
[41] X. Yang, Y. Guo, Y. Liu, Bayesian-Inference-Based Recommendation in Online Social Networks, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, 24 (2013) 642-651.
[42] M. Jiang, P. Cui, F. Wang, W. Zhu, S. Yang, Scalable Recommendation with Social Contextual Information, IEEE Transactions on Knowledge and Data Engineering, 26 (2014) 2789-2802.
[43] L. Xiong, X. Chen, T.-K. Huang, J. Schneider, J.G. Carbonell, Temporal Collaborative Filtering with Bayesian Probabilistic Tensor Factorization, Proceedings of the 2010 SIAM International Conference on Data Mining2010, pp. 211-222.
[44] N. Zheng, Q. Li, A recommender system based on tag and time information for social tagging systems, Expert Systems with Applications, 38 (2011) 4575-4587.
[45] Y. Koren, Collaborative filtering with temporal dynamics, Communications of the ACM, 53 (2010) 89-97.
[46] S.H. Daneshmand, A. Javari, S.E. Abtahi, M. Jalili, A time-aware recommender system based on dependency network of items, The Computer Journal, 58 (2015) 1255-1266.
[47] A. Lancichinetti, S. Fortunato, J. Kertesz, Detecting the overlapping and hierarchical community structure in complex networks, New Journal of Physics, 11 (2009) 033015.
[48] C. Birtolo, D. Ronca, Advances in Clustering Collaborative Filtering by means of Fuzzy C-means and trust, Expert Systems with Applications, 40 (2013) 6997–7009.
[49] R. Rafeh, A. Bahrehmand, An adaptive approach to dealing with unstable behaviour of users in collaborative filtering systems, Journal of Information Science, 38 (2012) 205-221.
[50] Kumar, M. S., & Prabhu, J. (2020). A hybrid model collaborative movie recommendation system using K-means clustering with ant colony optimisation. International Journal of Internet Technology and Secured Transactions, 10(3), 337-354.
[51] Hassan, B. A., & Rashid, T. A. (2021). A multidisciplinary ensemble algorithm for clustering heterogeneous datasets. Neural Computing and Applications, 33(17), 10987-11010.
[52] Hassan, B. A., Rashid, T. A., & Hamarashid, H. K. (2021). A novel cluster detection of COVID-19 patients and medical disease conditions using improved evolutionary clustering algorithm star. Computers in biology and medicine, 138, 104866.
[53] Hamarashid, H. K., Qader, S. M., Saeed, S. A., Hassan, B. A., & Ali, N. A. (2022). Machine Learning Algorithms Evaluation Methods by Utilizing R. UKH Journal of Science and Engineering, 6(1), 1-11.
[54] Kakarash, Z. A., Ezat, H. S., Omar, S. A., & Ahmed, N. F. (2022). Time Series Forecasting Based on Support Vector Machine Using Particle Swarm Optimization. International Journal of Computing, 21(1), 76-88. https://doi.org/10.47839/ijc.21.1.2520
[55] Pourbahrami, S., Balafar, M. A., Khanli, L. M., & Kakarash, Z. A. (2020). A survey of neighborhood construction algorithms for clustering and classifying data points. Computer Science Review, 38, 100315.
[56] Kakarash, Z. A., Karim, S. H. T., Ahmed, N. F., & Omar, G. A. (2021). New Topology Control base on Ant Colony Algorithm in Optimization of Wireless Sensor Network. Passer Journal of Basic and Applied Sciences, 3(2), 123-129.
[57] Omar, S. A., Othman, R. N., Ahmed, N. F., Ezat, H. S., & Kakarash, Z. A. effective data parallel optimization for quantifying the mathematical model of subsystems in multivariable systems.