I.J. Intelligent Systems and Applications, **2018, 10, 53-62**
Published Online October 2018 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2018.10.06

Temporal Community-Based Collaborative Filtering to Relieve from Cold-Start and Sparsity Problems

Anupama Angadi [1], Satya Keerthi Gorripati [2]
[1, 2] IT Department, GMR Institute of Technology, Rajam, Andhra Pradesh, India
E-mail: anupamaangadi.a@gmail.com[1], satyakeerthi.gsk@gmail.com[2]

P. Suresh Varma[3]
[3] Department of CSE, Adikavi Nannaya University, Rajahmundry, A.P, India
E-mail: vermaps@yahoo.com.

Received: 17 July 2017; Accepted: 20 December 2017; Published: 08 October 2018

Abstract—Recommender systems inherently dynamic in nature and exponentially grow with time, in terms of interests and behaviour patterns. Traditional recommender systems rely on similarity of users or items in static networks where the user/item neighbourhood is almost same and they generate the same recommendations since the network is constant. This paper proposes a novel architecture, called Temporal Community-based Collaborative filtering, which is an association of recommendation and the dynamic community algorithm in order to exploit the temporal changes in the community structure to enhance the existing system. Our framework also provides solutions to common inherent issues of collaborative filtering approach such as cold-start, sparsity and compared against static and traditional collaborative systems. The outcomes indicate that the proposed system yields higher values in quality standards and minimizes the drawbacks of the traditional recommender system.

Index Terms—Community Detection, Item-based, Collaborative Filtering, Neighbourhood Similarity, Recommender Systems, Temporal Data.

I. Introduction

With the increased usage of World Wide Web, information has increased at an exponential rate and the information overload problem has become increasingly severe for online users. For instance, we want to buy a laptop and search "laptop" in the Amazon, it returns lakhs of products. Recommender systems, tackles this issue (i.e., information overload problem) by suggesting information that is of potential interest to online users, have become familiar and popular [1,2,3,4]. Such systems are extensively implemented in various fields, including, movie recommendation on Netflix and product recommendation on Flipkart. For information customers, relevant recommendations allow them to find required information hidden in a great quantity of irrelevant data. For information providers, recommender systems not only help govern which information to provide, but also improve customer reliability because customers tend to cite those sites that best answer their demands. Recommender systems have caught much attention from various fields, such as psychology, natural philosophy and computer skill. For example, the champions of the Netflix prize contest, was a team of psychologists, physicists and computer scientists. Psychologists employ them as a locus for gathering up the patient behaviour before his/her discourse. Computer science persons, seek to exploit information about OSNs in the invention of marketing schemes. Biologist or physicist uses a biological/molecule network for understanding the function of the gene / molecule in identifying the disease/structure. Many methods are applied to build recommender systems, which can be generally separated into content-based filtering, collaborative filtering (CF) based methods, and hybrid methods [5].

The purpose of this paper is to model the temporal/dynamic community based recommender systems based on trusted and top similar neighbours in the community. We exploit this by comparing the user profile to some reference features with the aim to predict whether the user is interested in an unseen item.

The paper is organized as follows. Section II introduces the basics of recommender systems and community detection. Related work is discussed in Section III. Our proposed framework and experimental results are presented in Section IV and we concluded in Section V.

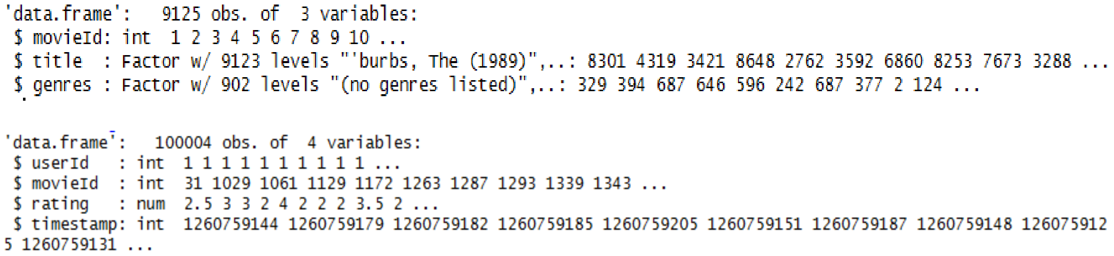
II. Basics Of Recommender Systems

A. Content based Recommender System

Content based filtering, recommends items based on the association between the content of the items and a user profile. Content, here refers to a set of features/attributes that describes the details. For a movie recommendation, a content based approach would be to recommend movies that are of highly relevant based on its characteristics such as genres (i.e., Action, Romance, Adventure, Comedy), actors, producers, directors, etc. The assumption is that users have some preferences for a certain type of item which were given in ratings data, and then we attempt to recommend a related detail to what the user has conveyed liking for.

MovieLens dataset contains ratings.dat, users.dat and movies.dat. A basic content-based recommender engine can be constructed based on movie genres only. It is likewise possible to make a more complicated engine by including several properties that have been determined to be more significant. This could be made out with a method known as Term Frequency-Inverse Document Frequency algorithm (TFIDF) [6].

The sample data from movies.dat and ratings.dat is represented as a data frame in R with observations is shown in Fig. 1.

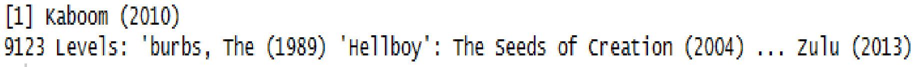


```
'data.frame':   9125 obs. of  3 variables:
 $ movieId: int  1 2 3 4 5 6 7 8 9 10 ...
 $ title  : Factor w/ 9123 levels "'burbs, The (1989)",..: 8301 4319 3421 8648 2762 3592 6860 8253 7673 3288 ...
 $ genres : Factor w/ 902 levels "(no genres listed)",..: 329 394 687 646 596 242 687 377 2 124 ...

'data.frame':   100004 obs. of  4 variables:
 $ userId   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ movieId  : int  31 1029 1061 1129 1172 1263 1287 1293 1339 1343 ...
 $ rating   : num  2.5 3 3 2 4 2 2 2 3.5 2 ...
 $ timestamp: int  1260759144 1260759179 1260759182 1260759185 1260759205 1260759151 1260759187 1260759148 126075912
5 1260759131 ...
```

Fig.1.Sample data set with observations

User gets recommendation based on similar items that are closest in similarity to a user’s profile, which is known as user’s preference for an item’s feature. If the user has a preference towards genres like Documentary, Horror, Musical and war the following movies will be recommended.



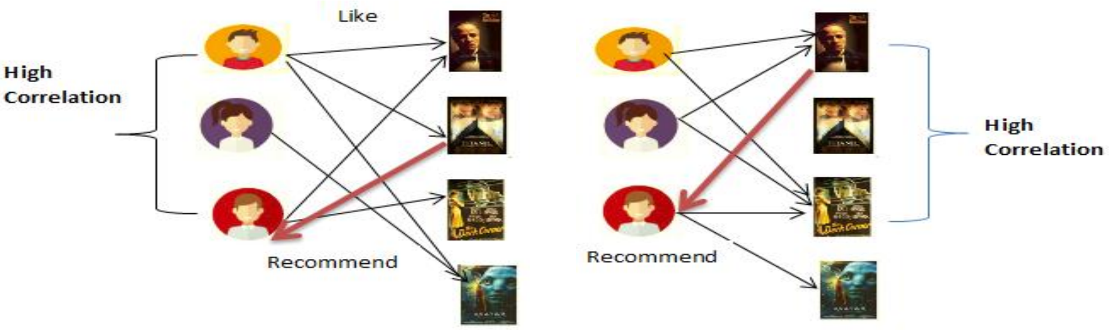
```
[1] Kaboom (2010)
9123 Levels: 'burbs, The (1989) 'Hellboy': The Seeds of Creation (2004) ... Zulu (2013)
```

Fig.2. Recommended rating for user1

B. *Collaborative Filtering*

Both user- and item- based approaches use the same approach of correlation of different objects [7, 8]. These objects are users if it is user based and item based if the object is an item. User-based collaborative filtering approach correlates users by excavating (similar) ratings and recommends unseen or new items that were preferred by similar users (see Fig.3). Item-based approach correlates the items by excavating (similar) ratings and then recommends similar or novel items. The main advantage of these approaches they are domain-independent and do not require content analysis and increases the quality of the recommendation.

However, CF approaches are defined by a number of drawbacks. First, the cold - start problem is due to the fact that Collaborative Filtering approaches depend on past behaviour patterns of the users. When new users or items are added, this drawback emerges since new users (cold-start user problem) or items (cold-start item problem) do not have sufficient ratings in order to make recommendations. Second, sparsity of the past user actions in a network. Finally, malicious attacks on centralized architectures and recommender systems [26, 30].



(a) User-based CF

(b) Item-based CF

Fig.3. Recommendations are indicated with red arrows using the User and Item based CF

C. *Hybrid Collaborative Filtering*

A hybrid recommender system is made by mixing two or more CF approaches described before to build a more robust system. By merging various recommender systems, we can eliminate the drawbacks of one system with the benefits of another system and thus build a more reliable robust system.

D. *Community Detection*

A community is a collection of vertices that share same characteristic or interest and can therefore be treated as a single entity in any respect[9]. The community structure of the network simply provides a barrier in the number of entities in order look into the structure of the network. For instance, it is much easier to understand the peoples' pulse by considering the voting patterns rather than looking at each individual voter's actions separately.

On the subject of complex networks, a network is said to possess community structure if the vertices of the network can be easily bifurcated into sets of nodes such that each such set is densely connected internally and sparsely connected externally. If a node contributes in one only one subgroup (community) then it is termed as disjoint node. On the other hand, if a node contributes in more than subgroup, then it is termed as overlapped node. In spite of many community detection algorithms, there is a still essential for a new one for multipurpose.

Disjoint community detection: Discovering communities from static network was first proposed by [10]. It is built with a modularity metric which is the fraction of the edges that fall within the given groups minus the expected, aiming to obtain best bifurcations [11]. To enhance the gain of modularity Parimi et al. has proposed Louvian algorithm [12]. Rosvall and Bergstrom have proposed LPA, considered as a solution to the simple problem of static community detection. The main notion behind LPA is to spread out the labels of node throughout the network using a technique and build communities through the process of label propagation [13]. The mentioned algorithms are not capable to handle dynamic nature of the network as well as overlapping communities [28]. To ensure this property several algorithms were proposed to exploit the evolutions of communities which require the time stamped data in the input [14, 15, 16].

III. Related Works

Content based recommender systems have their origin in information retrieval. Existing content based recommender systems focus on recommending item based on textual information, keywords available in the documents which are frequently measured by TFIDF weight, where TF denotes the frequency of the keyword in the document and IDF denotes the inverse document frequency of the keyword.

Collaborative filtering is the most common technique to form recommender systems [17]. It can predict the user preferences from users past behavioural patterns [7, 8, 27]. The premise of collaborative filtering is that if users have matching pattern in the past, they are more likely to agree with each other in the future. CF approaches are categorized into a memory-based and model-based[32]. In this paper, we propose a memory-based CF.

There has been much research on social recommender systems can get the advantage of analysis taken from social networks, correlation theories, community detection and trust for recommendations, since social networks and recommender systems based on the same hypothesis called "Friend-of-Friend" [29]. Even thus, the recent trend is on integrating community detection and recommender systems in order to offer more personalized recommendations to a group of users belong to the same community. The first step is to identify closed groups and in the second step it recommends unseen items to the target user based on community.

Many authors have proposed recommender systems using static community detection [18, 19] which can reveal user's preferences using collaborative and hybrid approaches on the YouTube, Twitter and other social networks.

IV. Proposed Framework

The proposed framework, called temporal community based collaborative recommender system, denoted by TCBCR, and is based on three major steps as shown in Fig. 4.

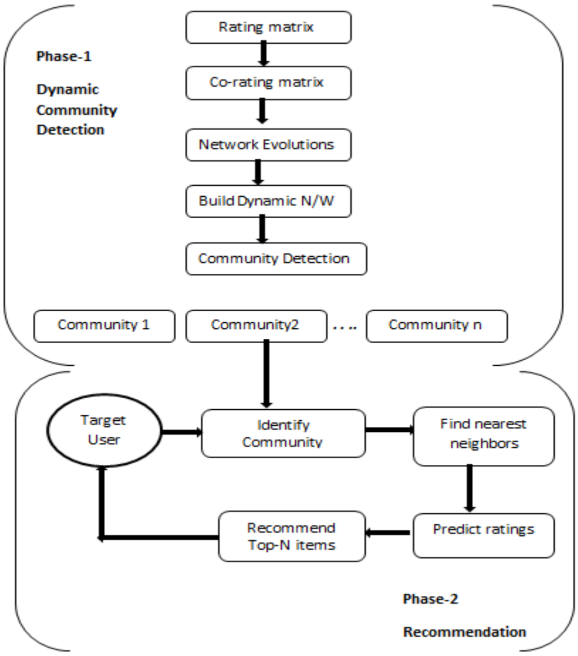


Fig.4. TCBCR architecture

Phase-1

A. *Pre-processing*

This step consists of pre-processing and evolutionary community detection. The input for phase-1 is the rating matrix which contains the ratings given by the users when the user does not rate an item we consider that it is not thus far watched by him. These evolutions of the users' preferences over time period are considered to build a dynamic network. Nodes (or Vertices) are treated as items and hence the communities are the groups of items. Based on the co-rating relationship an edge gets created between items. Indeed, relationship between nodes exists when a user gives the similar rating to both items in the same timestamp. With the aim to trace the evolutionary changes over time, we have adapted the method to build a temporal network Fig.5. An edge (interaction) gets made if two nodes must interact minimum number of times over a period of time. In a period of time T, if a specific number of interactions M met an edge will be added otherwise removed. These evolutions can be represented as snapshot where each one agrees to a change in the network structure over a specific period of time [20].

B. Dynamic Community Detection

Once a network is built in the pre-processing step, we can use those snapshots as input to dynamic community detection algorithms. Based on the network edges, a community can be defined as a set of items, where the interactions are extremely high (i.e., strongly connected) and tends to possess similar interests/preferences over a period of time. To assure the community structure, we adopt to use the LabelRankT algorithm. This option is justified that label propagation algorithms are efficient, popular in nature and detects communities in static and dynamic networks. It can also deal with both disjoint and the overlapped community structures. The following are the phases of the LabelRankT [14].

1. Input: snapshots of a graph.
2. Track the updated nodes in a graph due to changes in edges, they have attached to since the previous time stamps.
3. For the updated nodes, reinitialize their label distributions as in LabelRank.
4. Iteratively update only changed nodes and allot them to the corresponding communities as in LabelRank.

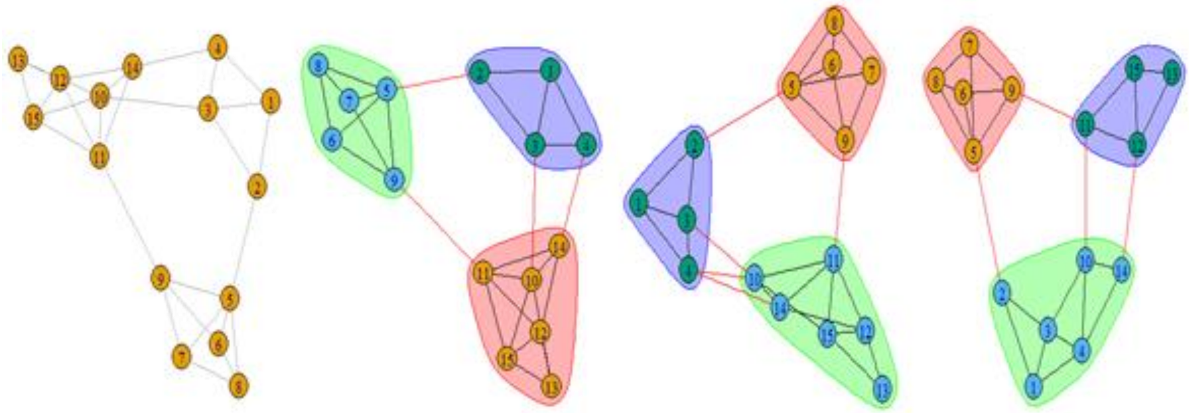


Fig.5. A small example of a temporal network, where a node changes its membership of the community based on evolutions over a period.

Phase-2

Communities hold certain hidden categories which help the recommender system to predict the users' future preferences. A highest rated item should be distinguished for every active user which is well-known as target item and candidate items are those which are not rated. Our objective is to predict the interest of the active user on the candidate items based on items that belong to the same community of the target item. By doing this our approach is not considering the whole dataset (like traditional item-based CF) instead it depends only on the community to which the candidate item belongs to. The list of *top-k* trusted neighbours with high predictions will be recommended to the active user. In our framework items with high correlation and direct interactions will be chosen as neighbours for predictions [21]. Directed nodes or between nodes are identified based on the distance. A distance matrix will be created between every node to all other nodes using a shortest path algorithm. On the other hand correlation between every item is computed using the Pearson correlation similarity measure. Formally the interest prediction is defined in (1) [22].

$$P_{u,i} = \frac{\sum_{j\in C} s(i,j) r_{u,j}}{\left|\sum_{j\in C} s(i,j)\right|} \tag{1}$$

where u is the active user and i is the target item and C is the set of items belong to the community of i, $r_{u,j}$ is the rating given by the active user and $s(i,j)$ similarity between item i and j.

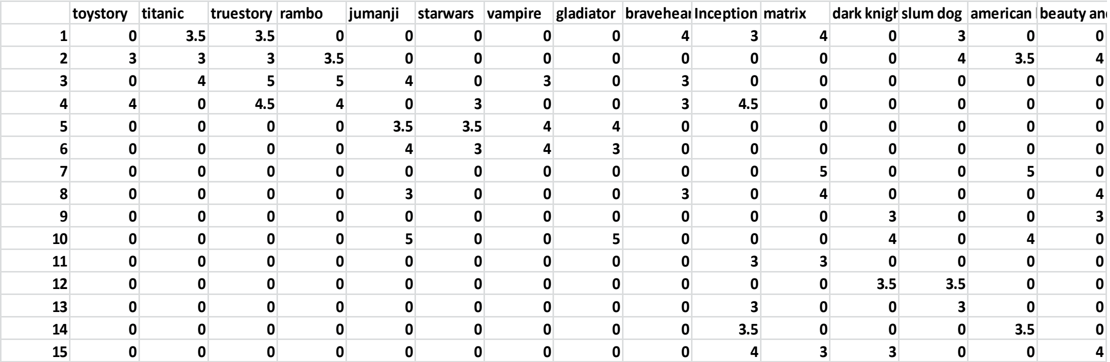
C. The strength of the proposed approach to relieve from a cold start, sparsity

The suggested approach takes in two advantages relative to other methods. First, with the inclusion of community approach it can ameliorate the cold start problem. Items with fewer ratings are termed as cold items. Based on the interactions, community approach force to locate all the nodes in any one of the community so there is no issue of cold items. Second, it provides solution for sparsity by predicting ratings of all the users.

D. *An Illustrative Example*

This step shows a step by step trace of dynamic community based recommender system to produce a prediction for a given unrated item. Suppose there are fifteen users and fifteen items, represented by u_k and i_j respectively, where $k, j \in [1,15]$ in a assured system. Each user may rate limited items by giving a rating between 1 and 5 as shown in Table 1.

Table 1. Rating data given by 15 users



	toystory	titanic	truestory	rambo	jumanji	starwars	vampire	gladiator	braveheai	Inception	matrix	dark knigł	slum dog	american	beauty an
1	0	3.5	3.5	0	0	0	0	0	4	3	4	0	3	0	0
2	3	3	3	3.5	0	0	0	0	0	0	0	0	4	3.5	4
3	0	4	5	5	4	0	3	0	3	0	0	0	0	0	0
4	4	0	4.5	4	0	3	0	0	3	4.5	0	0	0	0	0
5	0	0	0	0	3.5	3.5	4	4	0	0	0	0	0	0	0
6	0	0	0	0	4	3	4	3	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	5	0	0	5	0
8	0	0	0	0	3	0	0	0	3	0	4	0	0	0	4
9	0	0	0	0	0	0	0	0	0	0	0	3	0	0	3
10	0	0	0	0	5	0	0	5	0	0	0	4	0	4	0
11	0	0	0	0	0	0	0	0	0	3	3	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	3.5	3.5	0	0
13	0	0	0	0	0	0	0	0	0	3	0	0	3	0	0
14	0	0	0	0	0	0	0	0	0	3.5	0	0	0	3.5	0
15	0	0	0	0	0	0	0	0	0	4	3	3	0	0	4

First, identify the users' patterns from the ratings data set. Second, based on the frequency of interactions the network is partitioned into communities shown in Fig.6 and Fig.7. In our framework interaction between the items happen when a minimum one user gives the same rating in same timestamp which can be computed using co-rating occurrences. The co-rating matrix is converted into a binary matrix in order to establish a network [20].

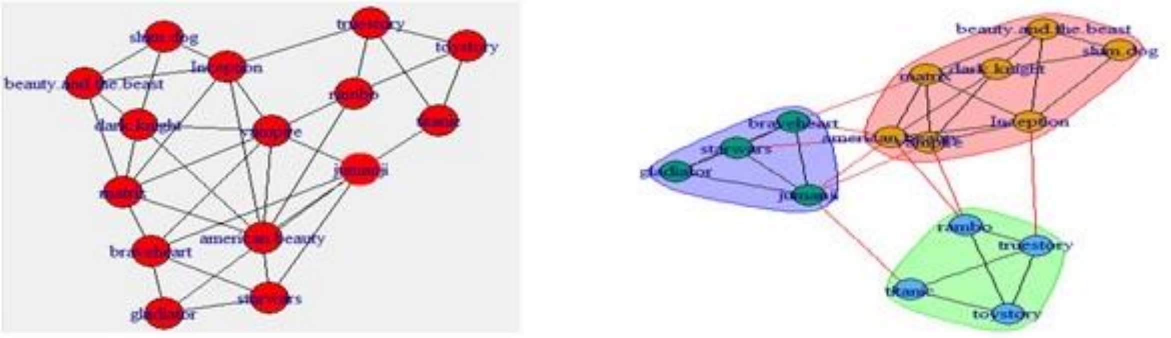
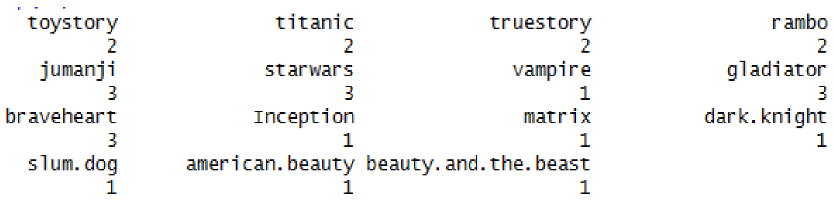


Fig.6. Obtained network and its communities for the data in Table 1

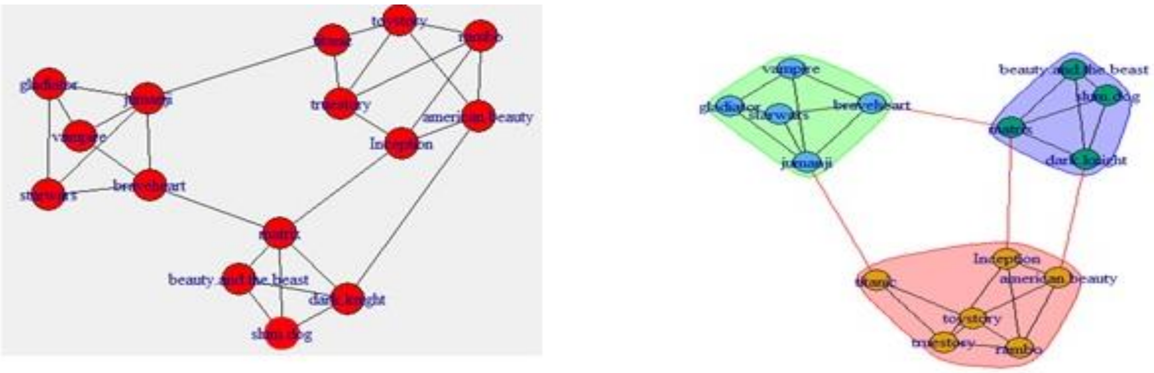


```
     toystory           titanic         truestory             rambo
            2                 2                 2                 2
      jumanji          starwars           vampire         gladiator
            3                 3                 1                 3
   braveheart         Inception            matrix       dark.knight
            3                 1                 1                 1
     slum.dog   american.beauty beauty.and.the.beast
            1                 1                 1
```

Fig.7. A list of vertices and its assigned communities

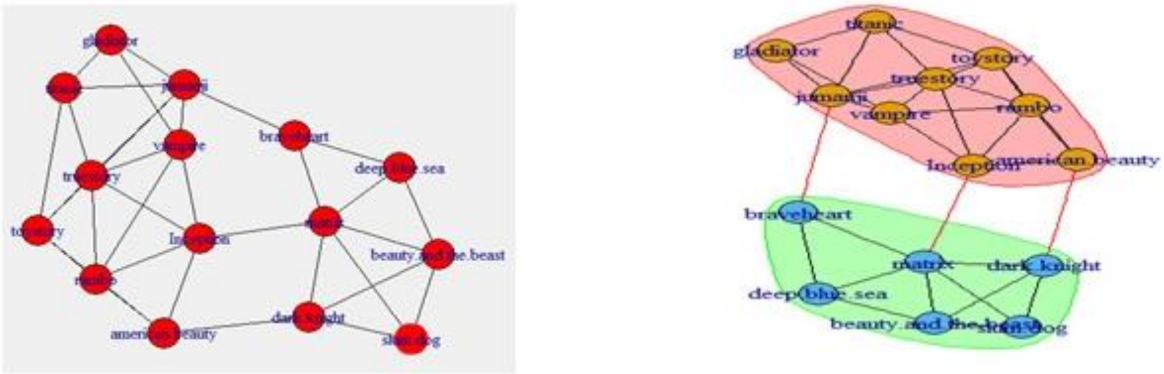
The growing popularity of social networks focusing on online communities became popular in recent years and most of the literature has been covered finding communities in static networks. However, temporal or dynamic aspects of communities in recommender systems have received a little consideration. Real-world networks are not static so in this framework we are bridging that gap to handle with the dynamic aspects of community-based recommender system [23].

Algorithms that identify communities in temporal networks uses several snapshots or timestamps of the network, taking into account the changes of the network that happen naturally between consecutive timestamps. In fact, a snapshot gets created for every change in the network and are determined by days or months based on pre-processing phase. The changes of the network happen only in the interactions which occurs when more than M interactions over a period of T days. The control of adding and deleting of edges depend on the values of M and T .The edge gets removed if it has fewer interactions. The resulting network with these evolutions (edges additions and removal) is represented at different time stamps are shown in Fig. 8.



```
   toystory            titanic          truestory            rambo
          1                  1                  1                1
    jumanji           starwars            vampire        gladiator
          2                  2                  2                2
 braveheart          Inception             matrix      dark.knight
          2                  1                  3                3
   slum.dog american.beauty beauty.and.the.beast
          3                  1                  3
```

Fig.8. (a) Network evolutions and list of vertices at time stamp t=2



```
   toystory            titanic          truestory            rambo
          1                  1                  1                1
    jumanji         big.hero.6            vampire        gladiator
          1                  3                  1                1
 braveheart          Inception             matrix      dark.knight
          2                  1                  2                2
   slum.dog american.beauty beauty.and.the.beast    deep.blue.sea
          2                  1                  2                2
```

Fig.8. (b) Network evolutions and list of vertices at time stamp t=3

Quality measure: For judging the strength of disjoint and overlapped communities [24,28] we use modularity as a metric shown in (2). This measures the difference between the number of edges found in community to the expected number of edges in a randomly generated community structure [17]. In our proposed framework we use (2) as we are concentrating on disjoint communities.

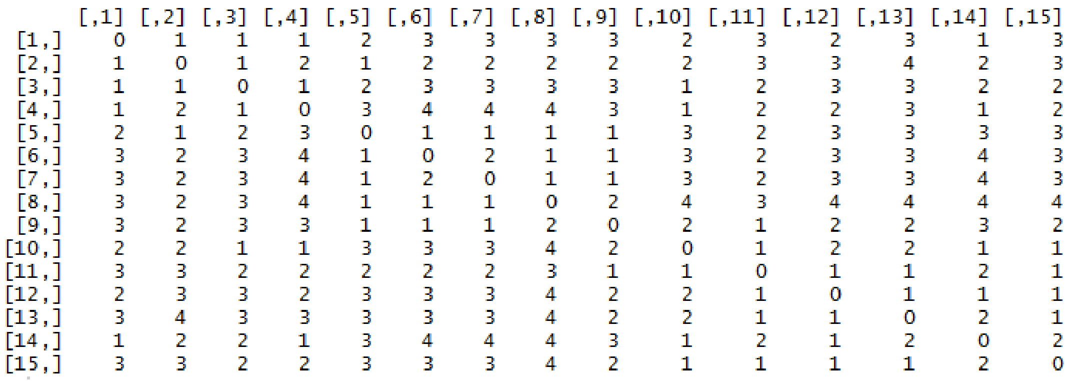
$$Q = \frac{1}{2M}\sum_{u,v}\left(A_{u,v} - \frac{k_u k_v}{2M}\right)S_{uv} \qquad (2)$$

Where M is the number of edges, A is the adjacency matrix and k_u, k_v represents the degree of node u and v. If a node u belongs to first community, then $s_u = 1$, or -1 otherwise. For visualization, calculating modularity metric we have used Gephi tool and it has shown 0.7982 for our approach [25]. Gephi is an interactive visualization platform for complex networks and large scale graph processing [31].

Third, highly correlated and trusted users are considered for neighbourhood similarity. Recommender systems, mostly consider the similarity between items of the nearest neighbour in order to predict the unseen ratings [26]. In our framework the vertices or nodes with less distance are seen as trusted users, since our family or best friends are mostly separated with 0 or 1-hop distance. The computation of the correlation shown in (3) between items is determined using Pearson correlation and the distance between every node to all the other nodes is computed using a shortest path algorithm. In the graph shown in Fig.8, the similarity computation and length of the shortest path is shown in Table 2(a) and Table 2(c).

$$P_{u,v} = \frac{\sum_{i\in I_{u,v}}(r_{u,i}-\overline{r}_u)(r_{v,i}-\overline{r}_v)}{\sqrt{\sum_{i\in I_{u,v}}(r_{u,i}-\overline{r}_u)^2}\sqrt{\sum_{i\in I_{u,v}}(r_{v,i}-\overline{r}_v)^2}} \qquad (3)$$

Table 2(a). The Length of shortest distance path between every vertex to all other vertices in the form of matrix



```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15]
 [1,]    0    1    1    1    2    3    3    3    3     2     3     2     3     1     3
 [2,]    1    0    1    2    1    2    2    2    2     2     3     3     4     2     3
 [3,]    1    1    0    1    2    3    3    3    3     1     2     3     3     2     2
 [4,]    1    2    1    0    3    4    4    4    3     1     2     2     3     1     2
 [5,]    2    1    2    3    0    1    1    1    1     3     2     3     3     3     3
 [6,]    3    2    3    4    1    0    2    1    1     3     2     3     3     4     3
 [7,]    3    2    3    4    1    2    0    1    1     3     2     3     3     4     3
 [8,]    3    2    3    4    1    1    1    0    2     4     3     4     4     4     4
 [9,]    3    2    3    3    1    1    1    2    0     2     1     2     2     3     2
[10,]    2    2    1    1    3    3    3    4    2     0     1     2     2     1     1
[11,]    3    3    2    2    2    2    2    3    1     1     0     1     1     2     1
[12,]    2    3    3    2    3    3    3    4    2     2     1     0     1     1     1
[13,]    3    4    3    3    3    3    3    4    2     2     1     1     0     2     1
[14,]    1    2    2    1    3    4    4    4    3     1     2     1     2     0     2
[15,]    3    3    2    2    3    3    3    4    2     1     1     1     1     2     0
```

Table 2(b). Pearson correlation visualization using heatmap in R

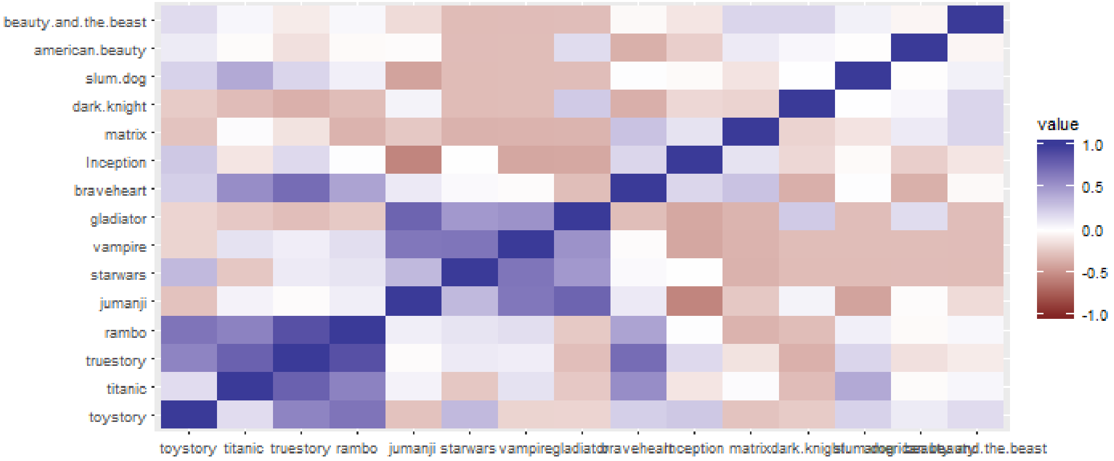
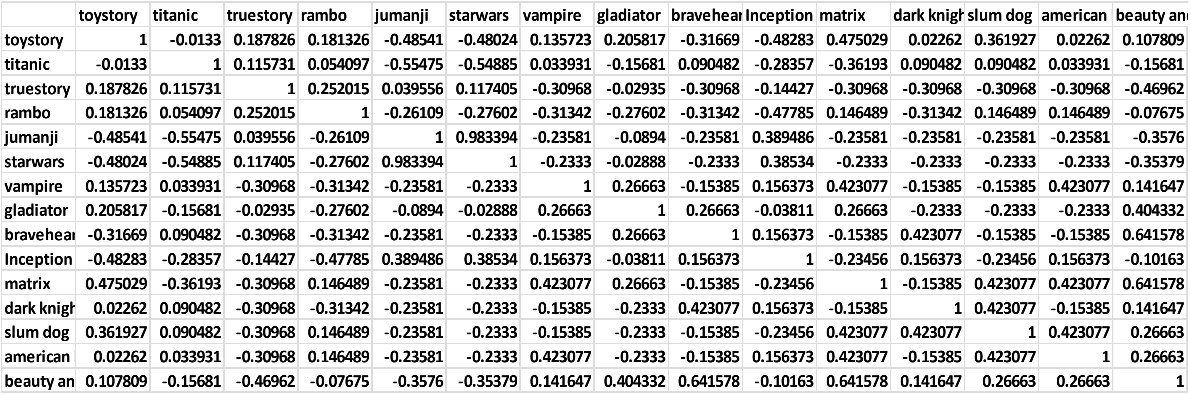


Table 2(c). Pearson correlation similarities between items



	toystory	titanic	truestory	rambo	jumanji	starwars	vampire	gladiator	bravehear	Inception	matrix	dark knigh	slum dog	american	beauty an
toystory	1	-0.0133	0.187826	0.181326	-0.48541	-0.48024	0.135723	0.205817	-0.31669	-0.48283	0.475029	0.02262	0.361927	0.02262	0.107809
titanic	-0.0133	1	0.115731	0.054097	-0.55475	-0.54885	0.033931	-0.15681	0.090482	-0.28357	-0.36193	0.090482	0.090482	0.033931	-0.15681
truestory	0.187826	0.115731	1	0.252015	0.039556	0.117405	-0.30968	-0.02935	-0.30968	-0.14427	-0.30968	-0.30968	-0.30968	-0.30968	-0.46962
rambo	0.181326	0.054097	0.252015	1	-0.26109	-0.27602	-0.31342	-0.27602	-0.31342	-0.47785	0.146489	-0.31342	0.146489	0.146489	-0.07675
jumanji	-0.48541	-0.55475	0.039556	-0.26109	1	0.983394	-0.23581	-0.0894	-0.23581	0.389486	-0.23581	-0.23581	-0.23581	-0.23581	-0.3576
starwars	-0.48024	-0.54885	0.117405	-0.27602	0.983394	1	-0.2333	-0.02888	-0.2333	0.38534	-0.2333	-0.2333	-0.2333	-0.2333	-0.35379
vampire	0.135723	0.033931	-0.30968	-0.31342	-0.23581	-0.2333	1	0.26663	-0.15385	0.156373	0.423077	-0.15385	-0.15385	0.423077	0.141647
gladiator	0.205817	-0.15681	-0.02935	-0.27602	-0.0894	-0.02888	0.26663	1	0.26663	-0.03811	0.26663	-0.2333	-0.2333	-0.2333	0.404332
bravehear	-0.31669	0.090482	-0.30968	-0.31342	-0.23581	-0.2333	-0.15385	0.26663	1	0.156373	-0.15385	0.423077	-0.15385	-0.15385	0.641578
Inception	-0.48283	-0.28357	-0.14427	-0.47785	0.389486	0.38534	0.156373	-0.03811	0.156373	1	-0.23456	0.156373	-0.23456	0.156373	-0.10163
matrix	0.475029	-0.36193	-0.30968	0.146489	-0.23581	-0.2333	0.423077	0.26663	-0.15385	-0.23456	1	-0.15385	0.423077	0.423077	0.641578
dark knigh	0.02262	0.090482	-0.30968	-0.31342	-0.23581	-0.2333	-0.15385	-0.2333	0.423077	0.156373	-0.15385	1	0.423077	-0.15385	0.141647
slum dog	0.361927	0.090482	-0.30968	0.146489	-0.23581	-0.2333	-0.15385	-0.2333	-0.15385	-0.23456	0.423077	0.423077	1	0.423077	0.26663
american	0.02262	0.033931	-0.30968	0.146489	-0.23581	-0.2333	0.423077	-0.2333	-0.15385	0.156373	0.423077	-0.15385	0.423077	1	0.26663
beauty an	0.107809	-0.15681	-0.46962	-0.07675	-0.3576	-0.35379	0.141647	0.404332	0.641578	-0.10163	0.641578	0.141647	0.26663	0.26663	1

The trouble of recommending items for any user using the ratings data shown in Table 1 is that most of the entries are sparse for the unseen/unknown items, we have to predict the ratings using Eqn. 1. For the active user $u1$ the target item is a *matrix* its community is 1 and the candidate items are *vampire, dark knight, american beauty* and *beauty and the beast*. The user believes his own ratings we keep his ratings as it is, unseen ratings are predicted and ratings are shown in Table 3. The recommended items for the user $u1$ are *vampire, beauty and the beast* and *gladiator*.

Table 3. The predicted ratings for the First user

Neighborhood similarity			Movie Ratings of the candidate items														
Vampire	0.423077	Dot product =>	0	0	3	0	4	4	0	0	0	0	0	0	0	0	0
Dark knight	-0.15385		0	0	0	0	0	0	0	0	3	4	0	3.5	0	0	3
American beauty	0.423077		0	3.5	0	0	0	0	5	0	0	4	0	0	0	3.5	0
Beauty and the beast	0.141647		0	4	0	0	0	0	0	4	3	0	0	0	0	0	4
P_{u_1}			0	3.5	3.5	0	4	4	5	4	4	3	4	3.5	3	0	4.5

The real-world networks are affected by changes in the underlying structure (i.e., addition or deletion of edges or vertices). Tracking these changes shows an effective role in Bioinformatics, Information Science and Marketing, etc. As an instance, the development of recommender systems can give efficient predictions where new ratings from users to items arrive constantly. For the new evolution shown in Fig.8(b) for the same user $u1$ has the target item *matrix*, but its community has changed to 2 and the updated candidate items are *braveheart, dark knight, beauty and beast, slum dog* and *dark blue sea.* The recommended items also change based on the new neighbours list.

E. Experimental Study

We evaluate the effectiveness of our framework using movieLens dataset available on the website (http://movieLens.umn.edu). This dataset contains 943 users on 1682 movie recommendations. The score is ranges from 1 to 5 and average rating of the user is at least 20 movies. MovieLens data are represented as a quadruple contains the following [user, item, rating, timestamp].

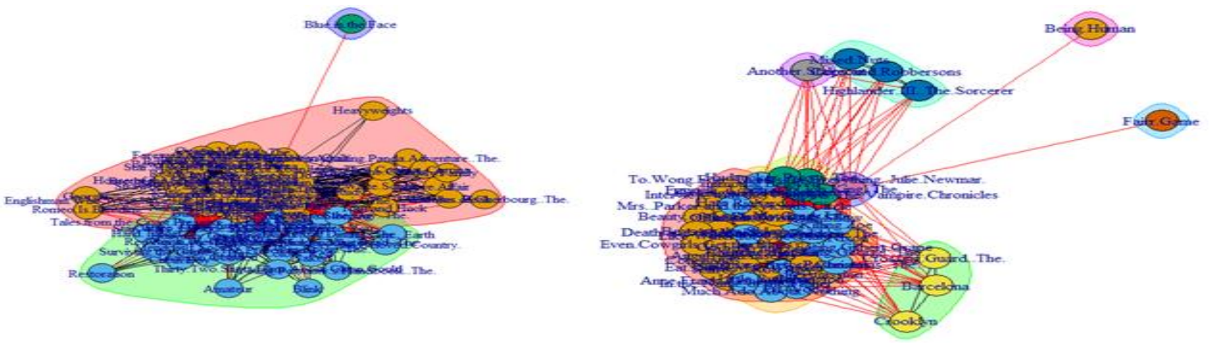


Fig.9. Major communities identified in the movieLens dataset over a period of time, where a node is a movie and link represents the co-relation between movies.

The data set should consider the evolutions over time, we propose to use 10-fold cross validation technique to determine the quality of the recommendations. We propose to validate the dataset in one scenario, in which the first scenario selects 90% of user ratings are occurrences in the training set and the residual ones will be used in the testing set.

There are many well-known metrics in the literature, the Mean Absolute Error and the Root Mean Square Error, precision, coverage and F-measure to measure the prediction accuracy. Our proposed framework uses (4) and (5) as quality measures.

$$(RMSE) = \sqrt{\frac{1}{T}\sum_{i,j}(R_{ij} - \overline{R}_{ij})^2} \tag{4}$$

$$precision = 1 - \frac{RMSE}{4} \tag{5}$$

where R_{ij}, is the rating given by user i to item j, $\overline{R}_{ij}$ is the predicted rating and T denotes the total number of tested.

The proposed approach is compared against static approach and traditional IBCF algorithms the quality measures were shown in Table 4 and Fig. 10.

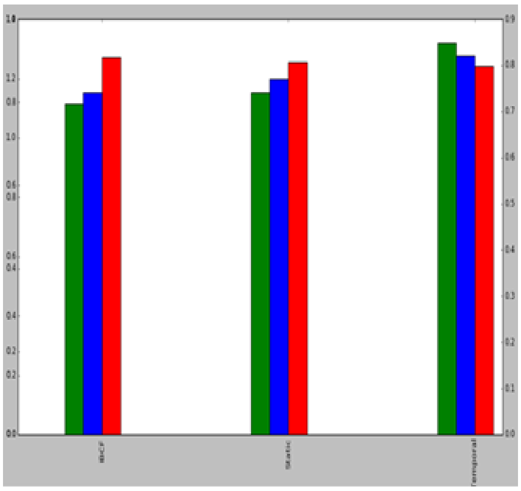


Fig.10. Accuracy comparisons

Table 4. Accuracy comparisons

Algorithms	RMSE	Precision	Recall
Item-based CF	1.271	0.824	0.716
Static	1.254	0.857	0.742
Temporal	1.243	0.912	0.849

V. Conclusion

In this paper, we proposed a Temporal Community-based Collaborative Filtering approach that merges community detection and recommendation to relieve from common problems of collaborative filtering approach. The proposed approach is able to deal with real-world network evolutions which explore the users' preferences over time. The experiment results show that our approach overtook than static community algorithms based on the $top-k$ neighbourhood of the entire dataset and traditional collaborative filtering approaches. As a forthcoming study, we will explore the neighbourhood similarity based on trusted users pertaining to the same community.

References

[1] Gregory, Steve. "Finding overlapping communities in networks by label propagation." *New Journal of Physics* , Vol.12, No. 10, p. 103018, 2010.

[2] Koren Y, "Collaborative filtering with temporal dynamics.", In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, New York, pp 447–456, 2009.

[3] Massa P, Avesani P, "Trust-aware recommender systems.", In: Proceedings of the 2007 ACM conference on recommender systems. ACM, New York, pp 17–24, 2007.

[4] Golbeck, Jennifer. "Generating predictive movie recommendations from trust in social networks." ,*Trust Management* , pp. 93-104,2006.

[5] Adomavicius, Gediminas, and Alexander Tuzhilin,"Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions.", *IEEE transactions on knowledge and data engineering,* Vol.17, No. 6,pp. 734-749,2005.

[6] Wu, Ho Chung, Robert Wing Pong Luk, Kam Fai Wong, and Kui Lam Kwok, "Interpreting tf-idf term weights as making relevance decisions." ,*ACM Transactions on Information Systems (TOIS),* Vol. 26, No. 3, 2008.

[7] Koren, Yehuda, "Factorization meets the neighborhood: a multifaceted collaborative filtering model.", In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining,* pp. 426-434. ACM, 2008.

[8] Su, Xiaoyuan, and Taghi M. Khoshgoftaar. "A survey of collaborative filtering techniques.", *Advances in artificial intelligence* , p.4,2009.

[9] Gregory, Steve, "Finding overlapping communities in networks by label propagation.", *New Journal of Physics, Vol.* 12, No.10, p. 103018, 2010.

[10] Girvan, Michelle, and Mark EJ Newman., "Community structure in social and biological networks.", *Proceedings of the national academy of sciences* , Vol.99, No. 12,pp. 7821-7826, 2002.

[11] Chen, Mingming, Konstantin Kuzmin, and Boleslaw K. Szymanski, "Extension of modularity density for overlapping community structure.", In *Advances in Social Networks Analysis and Mining (ASONAM), IEEE/ACM International Conference* ,pp. 856-863. IEEE, 2014.

[12] Parimi, Rohit, and Doina Caragea, "Community detection on large graph datasets for recommender systems.", In *Data Mining Workshop (ICDMW), IEEE International Conference,* pp. 589-596, 2014.

[13] Rosvall, Martin, and Carl T. Bergstrom, "An information-theoretic framework for resolving community structure in complex networks.", *Proceedings of the National Academy of Sciences, Vol* 104, No. 18,pp. 7327-7331,2007.

[14] Xie, Jierui, Mingming Chen, and Boleslaw K. Szymanski,"LabelrankT: Incremental community detection in dynamic networks via label propagation." In *Proceedings of the Workshop on Dynamic Networks Management and Mining,* pp. 25-32. ACM, 2013.

[15] Aston, Nathan, and Wei Hu, "Community detection in dynamic social networks.", *Communications and Network* , Vol.6, No. 02,p.124,2014..

[16] Nguyen, Nam P., Thang N. Dinh, Sindhura Tokala, and My T. Thai, "Overlapping communities in dynamic networks: their detection and mobile applications.", In *Proceedings of the 17th annual international conference on Mobile computing and networking,* ACM,pp. 85-96, 2011.

[17] Chen, Mingming, Konstantin Kuzmin, and Boleslaw K. Szymanski, "Extension of modularity density for overlapping community structure.", In *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference* , pp. 856-863, 2014.

[18] Zhao, Gang, Mong Li Lee, Wynne Hsu, Wei Chen, and Haoji Hu., "Community-based user recommendation in uni-directional social networks.", In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management,* ACM, pp. 189-198, 2013.

[19] Qiang, Hou, and Gai Yan, "A method of personalized recommendation based on multi-label propagation for overlapping community detection.", In *System Science, Engineering Design and Manufacturing Informatization (ICSEM), 3rd International Conference* IEEE, vol. 1, pp. 360-364, 2012.

[20] Abdrabbah, Sabrine Ben, Raouia Ayachi, and Nahla Ben Amor, "Collaborative filtering based on dynamic community detection.", *Dynamic Networks and Knowledge Discovery, p.* 85,2014.

[21] Deshpande, Mukund, and George Karypis, "Item-based top-n recommendation algorithms.", *ACM Transactions on Information Systems (TOIS),*Vol. 22, No. 1,pp. 143-177,2004.

[22] Abdrabbah, Sabrine Ben, Raouia Ayachi, and Nahla Ben Amor, "A dynamic community-based personalization for e-Government services.", In *Proceedings of the 9th International Conference on Theory and Practice of Electronic Governance,* ACM ,pp. 258-265, 2016.

[23] Hopcroft, John, Omar Khan, Brian Kulis, and Bart Selman, "Tracking evolving communities in large linked networks.", *Proceedings of the National Academy of Sciences* , Vol.101, No. suppl 1 ,pp. 5249-5253,2004.

[24] Angadi, Anupama, and P. Suresh Varma. "Overlapping community detection in temporal networks.", *Indian Journal of Science and Technology* ,Vol.8, No. 31, 2015.

[25] Bastian, Mathieu, Sebastien Heymann, and Mathieu Jacomy, "Gephi: an open source software for exploring

and manipulating networks." ,ICWSM 8, pp. 361-362, 2009

[26] Berkovsky, Shlomo, Ronnie Taib, and Dan Conway, "How to Recommend?: User Trust Factors in Movie Recommender Systems.", In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, pp. 287-300. ACM, 2017.

[27] Kushwaha, Nidhi, et al. "A Lesson learned from PMF based approach for Semantic Recommender System." *Journal of Intelligent Information Systems,* pp.1-13, 2017.

[28] Moradi, Parham, et al. "A trust-aware recommender algorithm based on users overlapping community structure." *Advances in ICT for Emerging Regions (ICTer), 2016 Sixteenth International Conference on*. IEEE, 2016.

[29] Nesrine, Gouttaya, et al. "Improving the Proactive Recommendation in Smart Home Environments: An Approach Based on Case Based Reasoning and BP-Neural Network." *International Journal of Intelligent Systems and Applications, Vol* 7, No.7, pp.29,2015.

[30] Sharma, Richa, Sharu Vinayak, and Rahul Singh. "Guide Me: A Research Work Area Recommender System." *International Journal of Intelligent Systems and Applications,* Vol.8, No.9, pp.30, 2016.

[31] Mohan, Anuraj, and G. Remya. "A Review on Large Scale Graph Processing Using Big Data Based Parallel Programming Models." *International Journal of Intelligent Systems and Applications,* Vol.9, No.2, pp.49, 2017.

[32] Maina, Elizaphan M., Robert O. Oboko, and Peter W. Waiganjo. "Using Machine Learning Techniques to Support Group Formation in an Online Collaborative Learning Environment." *International Journal of Intelligent Systems & Applications, Vol.* 9, No. 3, 2017.

Authors' Profiles

Dr.A.Anupama received M.Tech in Computer Science Technology from Andhra University, and Ph.D. in Computer Science and Engineering, from Adikavi Nannaya University, India. Her research interests include trust evaluation models, game theory, recommender systems and algorithms in online social networks. Most of her publications and guidance to the students relate to community detection algorithms and recommender systems.

Mr. G.Satya Keerthi received M.Tech in Computer Science Technology from GITAM University and pursuing ph.D from Andhra University. His research areas of interests are social network analysis, machine learning and cloud computing.

Dr. P.Suresh Varma is presently working as Professor in the Department of Computer Science, Adikavi Nannaya University, Rajahmundry, India. He published several research papers in national and international journals and presented papers at various conferences, seminars and workshops. He guided a large number of students for their M.Tech, MCA degrees in Computer Science and Engineering and Information Technology. His current research interests are Communication Networks, Data Mining and Cloud Computing.

How to cite this paper: Anupama Angadi, Satya Keerthi Gorripati, P. Suresh Varma, "Temporal Community-Based Collaborative Filtering to Relieve from Cold-Start and Sparsity Problems", International Journal of Intelligent Systems and Applications(IJISA), Vol.10, No.10, pp.53-62, 2018. DOI: 10.5815/ijisa.2018.10.06