International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 15 (2017) pp. 5022-5030

Community-Based Collaborative Filtering to Alleviate the Cold-Start and Sparsity Problems

Satya Keerthi Gorripati and Valli Kumari Vatsavayi

[1]*IT Department, GMR Institute of Technology, Rajam, Andhra Pradesh, India.*

Orcid Id: 0000-0002-1127-1055

[2]*Department of Computer Science and Systems Engineering, Andhra University, Visakhapatnam, Andhra Pradesh, India.*

Orcid Id: 0000-0002-7252-8301

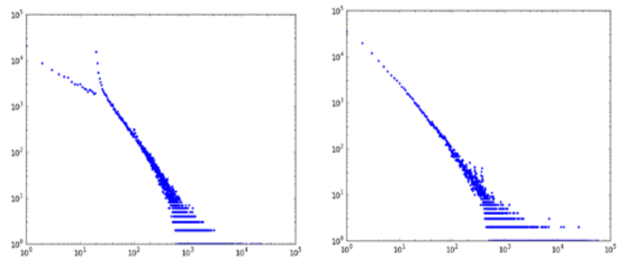
Abstract

Recommender systems help users in the discouraging task of selecting through large quantities of information in order to select relevant information or items. It relies on most similar users or items, when the information is large huge number of neighbors gain importance where the goal is to obtain a set of users with whom a target user is likely to match. Forming communities allows us to reveal like-minded users and also reduce the challenges of collaborative filtering like data-sparsity and cold-start problems. This paper proposes a community based collaborative filtering approach based on high correlation and shortest neighbor in the community. We carried out experiments on m1-1m dataset available on MovieLens datasets and experimental results indicate that the quality of the recommendation method is improved compared with traditional algorithms.

Keywords: Community Detection; cold start; collaborative filtering;co-occurances; recommendation.

INTRODUCTION

Advances in web 2.0 applications have significantly changed users' styles of online activities from sharing and interactions to searching and browsing. The options available grow up exponentially as shown in Fig. 1, and make it difficult for users to find the relevant information which is well-known as the information overload problem. Recommender systems have become a promising tool to manage this by suggesting the items that are potentially of their pursuits. Recommender systems are developed and heavily implemented in a recent e-commerce applications such as Amazon, Netflix and FilmAffinity to cope up with information overload problem [7,8] by providing users with personalized and quality recommendation to find items such as books, well-known news articles, web-pages, music tracks etc. that are likely to interest of a user.



(a) (b)

Figure 1: (a) Log log plot of movie ratings per user (b) number of ratings per movie

Several types of recommenders are proposed and can be broadly classified into three major categories, namely collaborative, content-based and hybrid filtering approaches [9]. In this framework, we focus on collaborative filtering, which recommends user preferences/interests based on remaining users sharing similar tastes. The primary idea of collaborative systems is to use the opinions of a community and it is viewed as one of the most used techniques due to its high accuracy and efficiency. Typically, the recommendation process starts when users show their view/preferences by rating items. The system recommends a set of items for the active user by analysing the proximity of active users' preference by mining the category of users having similar taste.

From another side, the community detection drags the attention of many researchers, especially in a web application to reveal the internal patterns of large networks. Panoply of community algorithms exists in literature to understand users' collective behaviour. The community detection techniques aim to find closed (or tight) subgroups among subjects such that the quantity of interaction inside the group is comparatively more than the interaction outside the group [10,11,12]. Static community detection algorithms have been explored in collaborative filtering based on the idea that community structure reveals the users' with common taste and will enhance the performance of recommendation [13,14]. However, these are not capable to trade with the shortest paths between the users in the community of real-world networks.

The purpose of this paper is to model the community based recommender systems based on shortest distance and top similarity of the neighbors in the community. We exploit the users' conduct of the ratings data and more specifically by looking firstly at the items which have been rated by each user in order to link items to each other.

The paper is organized as follows. Section 2 introduces the basics of recommender systems and community detection. Related work is provided in Section 3. Our proposed architecture is shown in Section 4 and Section 5 is dedicated to the experimental subject.

BASICS OF RECOMMENDER SYSTEMS

A .Content Based Recommender System

A content based recommender works with information supplied by the user explicitly or implicitly (clicking on a link). The rating is to be considered as explicit when he/she explicitly presses either thumb up or thumb down on situations such as Pandora and YouTube. In implicit, we don't ask users to give any ratings – we just observe their liking patterns and behavior from repeated visits and time spent looking at a web page and so on. Content Information retrieval systems are subclass of information filtering, where keywords are employed to describe the items. A user profile is built with these attributes to indicate the case of item user like, the concept of Term Frequency (TF) and Inverse Document Frequency (IDF) is widely applied to determine the importance of an item [15].

Articles	Baseball	Soccer	security
Doc 1	21	59	0
Doc 2	24	115	8
Doc 3	4	28	8
Doc 4	8	48	4
Doc 5	17	49	19
Doc 6	40	24	25
DF	5,000	50,000	10,000

Figure 2: The top 3 links if we search for "Sports" on Google

Consider Fig. 2, to find the relevant document; First, compute the document frequency for every document in the corpus. TF is calculated for each entry, the term baseball and Doc1 gives the entry 2.322 using the equation $1+\log_{10} 21$ (if the corpus size is 1 million). Second, an Inverse document frequency score calculated using the equation $\log_{10}$ (corpus size/DF). Next, the sum of the squared root of values of each attribute in the vector is used to compute the length of the vector and each term vector is divided by the vector length to get the normalized vector. Finally, Cosine similarity is used to recognize the similarities between articles. From Fig.2 Doc1 and Doc2 are comparatively more similar.

B. Collaborative Filtering

Collaborative filtering (CF) [9] is widely used technique for recommender systems which is based on users' rating information. It suggests items based on similarities between the active user's profile and other users, which is well-known as User-based collaborative filtering. These users are likely to be as marked the active user whose ratings are applied to predict the active users' rating value. To predict the rating of user 7 (active user) for item 2, scan the items for all the users who rated that item. For N users, a user similarity matrix is computed for each pair of items and chooses k most similar users to the active user. Finally, the predicted ratings are computed for the active user and recommend the items that have a highest predicted rating value.

	u1	u2	u3	u4	u5	u6	u7	u8
i1						1		
i2			2	2				
i3	5	4			5		2	
i4			?				4	5
i5								
i6	?				?	3		
i7					2			
i8		3		3		1	2	1

Figure 3: Items 3,4 and 8 are used to predict the ratings of user 7.

On the other hand, when the relationships are explored between the similarities of items rather than users, then it is known as Item-based collaborative filtering. To predict the rating of user 1 (active user) for the item 1(target item), identify the users who have rated for item 1. For N items, an item similarity matrix is computed since similar peers have almost similar preferences and compute the similarity for each of those pairs.

	i1	i2	i3	i4	i5	i6	i7	i8
u1			5			?		
u2			4					3
u3	2			?				
u4	2							3
u5			5			?	2	
u6	1					3		1
u7			2	4				2
u8				5				1

Figure 4: Users 3, 4 and 6 are used to predict the rating of item 1.

In collaborative filtering we rely on "most similar" users/items. Based on the data many measures like KNN, Cosine, Euclidean or Manhattan, Pearson can be used. If the data is sparse consider Cosine similarity, if it is dense use either Euclidean or Manhattan, Pearson is used in case of subject to grade inflation. The major drawbacks of collaborative filtering are Cold start, Popularity effect, Sparseness and Trust.

C. Hybrid Collaborative Filtering

Numerous hybrid approaches have been introduced based on the combination of content-based filtering and collaborative filtering recommendation methods. It uses the ratings history of users and the items which are capable to address the Cold start and Sparseness of both collaborative and content filtering.

D. Community Detection

The aim of Community detection technique is to find a group of vertices that share some characteristics and interest and can therefore be considered as a single entity in some respect. The community structure of the network simply provides a partition in the number of entities in order to understand the network. Community detection is one of the most significant and challenging tools for analysing complex networks in the area of research. It is the process of discovering groups in a network where individuals group memberships are not explicitly given.

The target of standard community detection problem is to partition a network into disjoint communities in a static network where a node belongs to just one community. On the other hand the nodes of overlapped community structures go to multiple communities.

Disjoint community detection: The first idea of using static networks is to discover communities that were proposed by Girvan and Newman [16]. It is based on a modularity metric which is the fraction of the edges that fall within the given groups minus the expected, aiming to obtain optimum partitions. To optimize the gain of modularity Guillaume et al. has proposed Louvian algorithm [18]. Rosvall and Bergstrom have presented LPA, considered as a solution to the simplest problem of static community detection. The main notion behind LPA is to pass around the labels of node throughout the network using a technique and build communities through the process of label propagation [19]. The mentioned algorithms are not able to detect overlapping communities where a node can belong to more than one community at the same time. To ensure this property, Palla proposed Clique-percolation Method to extract communities based on finding all k-cliques in the graph, which requires the size of the cliques in the input.

RELATED WORK

Many Collaborative filtering methods have been proposed in the literature to resolve the drawbacks of CF Cold start, data sparsity and trust. It is called collaborative because nearest neighborhood people collaborate to come up with recommendations. Community detection is the solution to understand users' collective behavior instead of focusing on the whole lot of neighbors.

In [1], authors have built a social network of movies from the Internet. Movie Database using Louvian community method in order to provide personalized recommendations. The proposed work in [2] mined topics as well as communities are based on probabilistic model by integrating user guidance with social network. The authors in [3] gave a solution to cold-start problem based on homophily in social networks by capturing similarities from different dimensions to help the recommendations. The approach in [4] reduces the impact of malicious users by keeping them in a separate community and user predictions calculated based on trusted communities. By combining link and content information from OSN a joint NMF model was proposed which discovers the friends from interested friends to ease the problem of information load [5]. To improve the scalability, coverage and cold start an approach [6] is developed based on collaborative filtering using map-reduce framework.

PROPOSED ARCHITECTURE

The proposed framework, called community based collaborative recommender system, denoted by CBCR, and is based on three major steps as shown in Fig. 5.

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 15 (2017) pp. 5022-5030

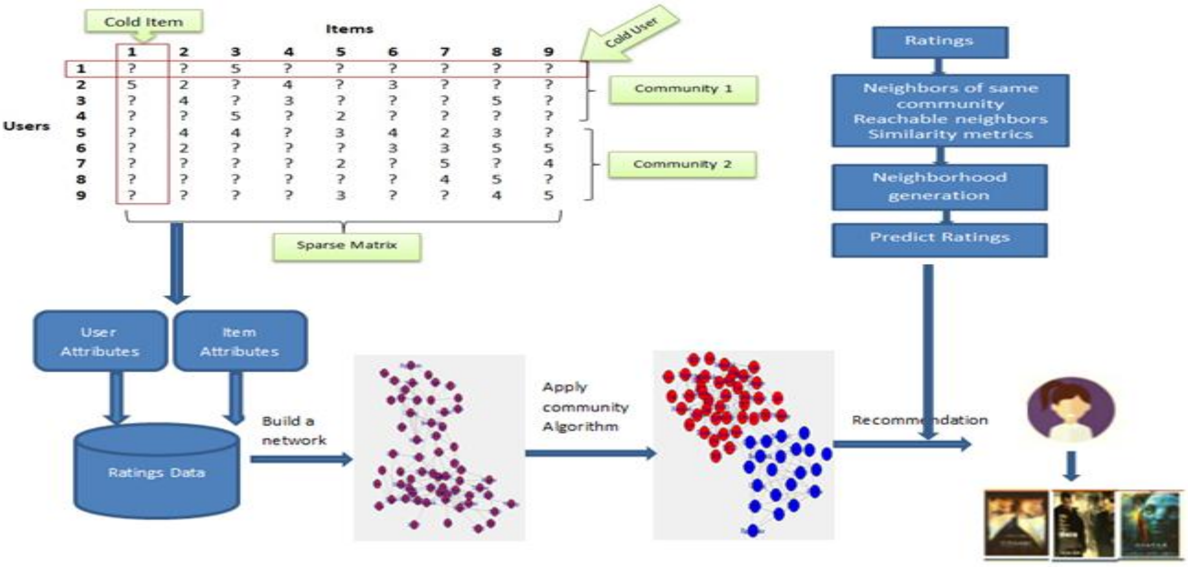


Figure 5: CBCR architecture

A. The Pre-processing Step

This step consists of building a network of the users' preferences (i.e., the set of items ranked). We assume that the item is not watched by the user if he/she does not rate an item [20]. In this framework, the nodes are the items and hence the communities are the subgroups of items. Indeed, the items have interaction with each other if they have co-rating relationship (i.e., at least one common rating between items). With the objective to prepare the list of relationships, we have adapted the method to extract user ratings based on co-occurrences whose ratings are above 3 which in turn assigns a count between pair of items. A network is built based on either adjacency matrix or edge list. Co-occurrence matrix then converted into adjacency matrix by replacing a non-zero value with one and a zero represents no interaction.

B. Community Detection Step

Once a network is built in the pre-processing step, we can use it as input to community detection algorithms. A community can be defined as a set of items where the interactions are frequent (i.e., strongly connected) that tend to have similar interests/preferences. The community algorithms are not only restricted to items it is also applicable on the feature called genres (eg., Horror, Comedy, Romance etc.) in the recommender systems dataset. To ensure the community structure, we choose to use the label propagation algorithm. This choice is justified that label propagation algorithms are popular, efficient in nature and detects communities in static network. Moreover, the algorithms can deal with both disjoint and the overlapped community structures. The phases of the label propagation algorithm may be described via three phases [11].

1. Initialize the node ids as labels for all nodes in the network
2. At each iteration every node updates its label based on the labels to the one carried by the most of its neighbors.
3. At the end of the algorithm, nodes with same label form a community.

The idea behind Label propagation algorithms to simulate the propagation of labels in the network based on denser connections as shown in Fig. 6.

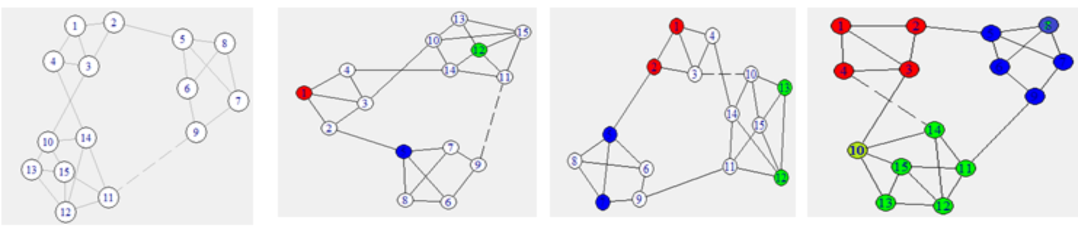


Figure 6: An example network is shown where each node has a distinct label (i.e., node id) and at every step each node changes its label.

C. Recommendation Step

In this step, the learned patterns from communities will be exploited to help the recommender system to predict the active users' future preferences based on certain categories given by the communities. Firstly, a highest rated item (i.e., target item) should be identified for each user. The items which are not rated by the active user and belong to the community of the target item are treated as candidate items. The list of $top-k$ neighborhood and trusted (i.e., direct interaction) users contains the k candidate items that have the highest predicted preferences. The nearest neighborhood is well known as $top-k$, where the top k most similar users will be used. However, since in this framework we focus on the Sparsity and Cold start, the top-k method is less effective than the threshold (i.e., shortest distance) method. Therefore, we combine both methods to select the neighbors for the active users.

Finally, nearest neighbors ratings belongs to the target items' community are aggregated to produce a prediction for the active user u on a target j that has not rated. The advantage of this approach is that we only rely on the items communities extracted from the network whereas the traditional item-based approach considers all items in the ratings data set into consideration. Formally the preference prediction as in Eqn. (1)

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

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.

D. Strength of the Proposed Approach

The core problems of CF approach was addressed in the proposed approach by amalgamating community algorithm. Cold-start items are forcibly falling into communities based on co-rating interactions so there is no issue of cold-start problem. Sparsity can be predicted by using rating predictor to predict all sparse items.

E. An Example

In this step, we intend to exemplify step by step trace of community based recommender system to generate a prediction for a given item. Suppose there are twenty users and twenty items, denoted by u_k and i_j respectively where $k, j \in [1, 20]$ in a certain system. Each user may rate a few items by giving a rating between [1, 5] as shown in Table 1.

Table 1: Rating data given by 20 users

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]	[,11]	[,12]	[,13]	[,14]	[,15]	[,16]	[,17]	[,18]	[,19]	[,20]
[1,]	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
[2,]	NA	NA	NA	NA	NA	NA	NA	NA	NA	4	NA	NA	NA	NA	NA	NA	5	NA	NA	NA
[3,]	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
[4,]	NA	NA	NA	NA	NA	NA	NA	NA	NA	4	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
[5,]	NA	NA	4	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
[6,]	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
[7,]	3.0	NA	NA	NA	NA	NA	NA	NA	NA	3	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
[8,]	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
[9,]	4.0	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	4	NA	NA	NA
[10,]	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
[11,]	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
[12,]	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
[13,]	5.0	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
[14,]	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
[15,]	NA	NA	NA	NA	4.5	4.0	NA	NA	NA	3	NA	NA	NA	NA	NA	3.5	3	NA	NA	NA
[16,]	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
[17,]	NA	NA	NA	NA	NA	4.5	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
[18,]	NA	NA	NA	NA	3.0	4.0	3	NA	3	NA	NA	NA	NA	NA	NA	NA	4	3	NA	NA
[19,]	3.0	3	3	3	NA	3.0	3	NA	3	3	3	NA	NA	5	NA	5.0	NA	NA	NA	NA
[20,]	3.5	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

The First step of the method is to identify users' patterns from the ratings data and more specifically by looking firstly at the items which have are rated by the user as shown in Table 1. Second, is to build a network based on the interaction between the items, which are modelled using co-ratings relationship. Indeed, two items interact with each other if at least one user gives the same rating and rating above 3 to both of them. According to Table 2(a), co-ratings are the count of similarities between the items; social graphs are build based on edge list or adjacency matrix. In order to obtain the adjacency matrix each non-zero entry in the co-rating matrix is replaced with one and an entry zero is left as it is (i.e., no interaction) shown in Table 2(b). In addition users may specify adjacent users as trusted neighbors.

Table 2 :

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20
V1	5	1	1	1	0	1	1	0	1	2	1	0	0	1	0	1	1	0	0	0
V2	1	1	1	1	0	1	1	0	1	1	1	0	0	1	0	1	0	0	0	0
V3	1	1	2	1	0	1	1	0	1	1	1	0	0	1	0	1	0	0	0	0
V4	1	1	1	1	0	1	1	0	1	1	1	0	0	1	0	1	0	0	0	0
V5	0	0	0	0	2	2	1	0	1	1	0	0	0	0	0	1	2	1	0	0
V6	1	1	1	1	2	4	2	0	2	2	1	0	0	1	0	2	2	1	0	0
V7	1	1	1	1	1	2	2	0	2	1	1	0	0	1	0	1	1	1	0	0
V8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V9	1	1	1	1	1	2	2	0	2	1	1	0	0	1	0	1	1	1	0	0
V10	2	1	1	1	1	2	1	0	1	5	1	0	0	1	0	2	2	0	0	0
V11	1	1	1	1	0	1	1	0	1	1	1	0	0	1	0	1	0	0	0	0
V12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V14	1	1	1	1	0	1	1	0	1	1	1	0	0	1	0	1	0	0	0	0
V15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V16	1	1	1	1	1	2	1	0	1	2	1	0	0	1	0	2	1	0	0	0
V17	1	0	0	0	2	2	1	0	1	2	0	0	0	0	0	1	4	1	0	0
V18	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	1	1	0	0
V19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(a) co-ratings matrix

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20
V1	1	1	1	1	0	1	1	0	1	1	1	0	0	1	0	1	1	0	0	0
V2	1	1	1	1	0	1	1	0	1	1	1	0	0	1	0	1	0	0	0	0
V3	1	1	1	1	0	1	1	0	1	1	1	0	0	1	0	1	0	0	0	0
V4	1	1	1	1	0	1	1	0	1	1	1	0	0	1	0	1	0	0	0	0
V5	0	0	0	0	1	1	1	0	1	1	0	0	0	0	0	1	1	1	0	0
V6	1	1	1	1	1	1	1	0	1	1	1	0	0	1	0	1	1	1	0	0
V7	1	1	1	1	1	1	1	0	1	1	1	0	0	1	0	1	1	1	0	0
V8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V9	1	1	1	1	1	1	1	0	1	1	1	0	0	1	0	1	1	1	0	0
V10	1	1	1	1	1	1	1	0	1	1	1	0	0	1	0	1	1	0	0	0
V11	1	1	1	1	0	1	1	0	1	1	1	0	0	1	0	1	0	0	0	0
V12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V14	1	1	1	1	0	1	1	0	1	1	1	0	0	1	0	1	0	0	0	0
V15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V16	1	1	1	1	1	1	1	0	1	1	1	0	0	1	0	1	1	0	0	0
V17	1	0	0	0	1	1	1	0	1	1	0	0	0	0	0	1	1	1	0	0
V18	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	1	1	0	0
V19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(b) adjacency matrix

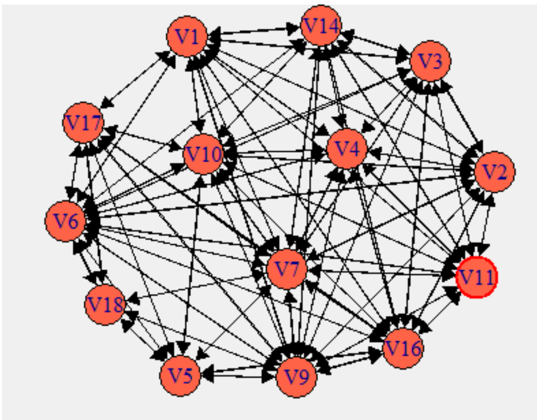
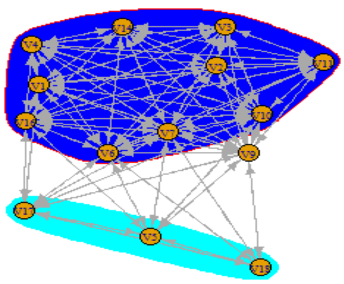


Figure 7: Obtained network using adjacency matrix

Third, community structures are identified using label propagation algorithm which is based on extremely related items that tends to possess the same interests of several users shown in Fig. 8. The framework can address the cold-start problem by using social connection information when the rating data are not sufficient and the second problem Sparsity can be reduced when a network is partitioned into sub graphs.



V1	V2	V3	V4	V5	V6	V7	V9	V10	V11	V14	V16	V17	V18
1	1	1	1	2	1	1	1	1	1	1	1	2	2

Figure 8: A list contains the vertex id and its assigned community

Fourth, neighborhood similarity is computed based on highly correlated and top k trusted users. The selection of k must be guarded, if is small it might get recommendations that are all over the place and if it is big recommendations might be general. Correlation between items can be calculated using many approaches (Pearson, Spearman, Cosine, Euclidean etc.) Pearson correlation coefficient approach is widely used one in the literature, so we have considered this approach which was shown in Table 3(b) and trusted users are identified using Depth First Search shown in Table 3(a).

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

Table 3: (a) The Length of shortest distance path between every vertex to all other vertices in the form of matrix (b) Pearson correlation similarities between items.

(a)

	V1	V2	V3	V4	V5	V6	V7	V9	V10	V11	V14	V16	V17	V18
V1	0	1	1	1	2	1	1	1	1	1	1	1	1	2
V2	1	0	1	1	2	1	1	1	1	1	1	1	2	2
V3	1	1	0	1	2	1	1	1	1	1	1	1	2	2
V4	1	1	1	0	2	1	1	1	1	1	1	1	2	2
V5	2	2	2	2	0	1	1	1	1	2	2	1	1	1
V6	1	1	1	1	1	0	1	1	1	1	1	1	1	1
V7	1	1	1	1	1	1	0	1	1	1	1	1	1	1
V9	1	1	1	1	1	1	1	0	1	1	1	1	1	1
V10	1	1	1	1	1	1	1	1	0	1	1	1	1	2
V11	1	1	1	1	2	1	1	1	1	0	1	1	2	2
V14	1	1	1	1	2	1	1	1	1	1	0	1	2	2
V16	1	1	1	1	1	1	1	1	1	1	1	0	1	2
V17	1	2	2	2	1	1	1	1	1	2	2	1	0	1
V18	2	2	2	2	1	1	1	1	2	2	2	2	1	0

(b)

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]	[,11]	[,12]	[,13]	[,14]
[1,]	1.00	0.29	0.07	0.29	-0.18	-0.10	0.12	0.12	0.05	0.29	0.29	0.17	0.02	-0.13
[2,]	0.29	1.00	0.57	1.00	-0.07	0.33	0.69	0.69	0.33	1.00	1.00	0.81	-0.11	-0.05
[3,]	0.07	0.57	1.00	0.57	-0.11	0.11	0.36	0.36	0.10	0.57	0.57	0.44	-0.16	-0.08
[4,]	0.29	1.00	0.57	1.00	-0.07	0.33	0.69	0.69	0.33	1.00	1.00	0.81	-0.11	-0.05
[5,]	-0.18	-0.07	-0.11	-0.07	1.00	0.67	0.33	0.33	0.21	-0.07	-0.07	0.42	0.52	0.52
[6,]	-0.10	0.33	0.11	0.33	0.67	1.00	0.58	0.58	0.17	0.33	0.33	0.55	0.30	0.47
[7,]	0.12	0.69	0.36	0.69	0.33	0.58	1.00	1.00	0.15	0.69	0.69	0.53	0.25	0.69
[8,]	0.12	0.69	0.36	0.69	0.33	0.58	1.00	1.00	0.15	0.69	0.69	0.53	0.25	0.69
[9,]	0.05	0.33	0.10	0.33	0.21	0.17	0.15	0.15	1.00	0.33	0.33	0.47	0.32	-0.13
[10,]	0.29	1.00	0.57	1.00	-0.07	0.33	0.69	0.69	0.33	1.00	1.00	0.81	-0.11	-0.05
[11,]	0.29	1.00	0.57	1.00	-0.07	0.33	0.69	0.69	0.33	1.00	1.00	0.81	-0.11	-0.05
[12,]	0.17	0.81	0.44	0.81	0.42	0.55	0.53	0.53	0.47	0.81	0.81	1.00	0.09	-0.08
[13,]	0.02	-0.11	-0.16	-0.11	0.52	0.30	0.25	0.25	0.32	-0.11	-0.11	0.09	1.00	0.45
[14,]	-0.13	-0.05	-0.08	-0.05	0.52	0.47	0.69	0.69	-0.13	-0.05	-0.05	-0.08	0.45	1.00

According to Table 1, for the active user u_5 the target item is i_3 and its community is 1 as shown in Fig. 8. The candidate items $i_1, i_2, i_4, i_6, i_7, i_9, i_{10}, i_{11}, i_{14}, i_{16}$ belong to the community of the target item and preference prediction for those items calculated using Eqn 1. Items $i_6, i_9, i_{16}, i_{17}, i_{18}$ were chosen as top-5 neighborhood for the target item based on shortest distance and high correlation. However, if the traditional collaborative method is considered this neighborhood list may change to $i_6, i_{12}, i_{13}, i_{14}, i_7$ since it considers only highly correlated neighbors, which is purely based on rating patterns without considering preference. In our framework these preferences are computed based on co-occurrence ratings. For the user u_5 the predicted values are shown in Table 4 and top predictions will be preferred as recommendation for the user those are identified as $i_1, i_5, i_6, i_{15}, i_{16}$.

Table 4: Predicted values for user u_5

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]	[,11]	[,12]	[,13]	[,14]	[,15]	[,16]	[,17]	[,18]	[,19]	[,20]
[1,]	5.34	3	2.5	2	3.6	4.17	3	2	2.55	3	2	3	2	2	3.5	3.66	3	2	2	1.5

F. Experimental Study

We propose to use the m1-1m dataset in movieLens which contains in total 1 million ratings collected by 6000 users on 4000 movies available through the movieLens website (https://grouplens.org/datasets/movielens/). After pre-processing the graph size reduces and the ratings are made on a 5-star scale. Each user has rated at least 20 movies. Each movie is represented by a node and we suppose that a community is defined as a set of nodes. A user rates a movie; this means that he is interested in watching it. MovieLens are represented as a sequence of events in the following way:

User U_1 rates movie I_1 with 4

User U_2 rates movie I_2 with 3, etc

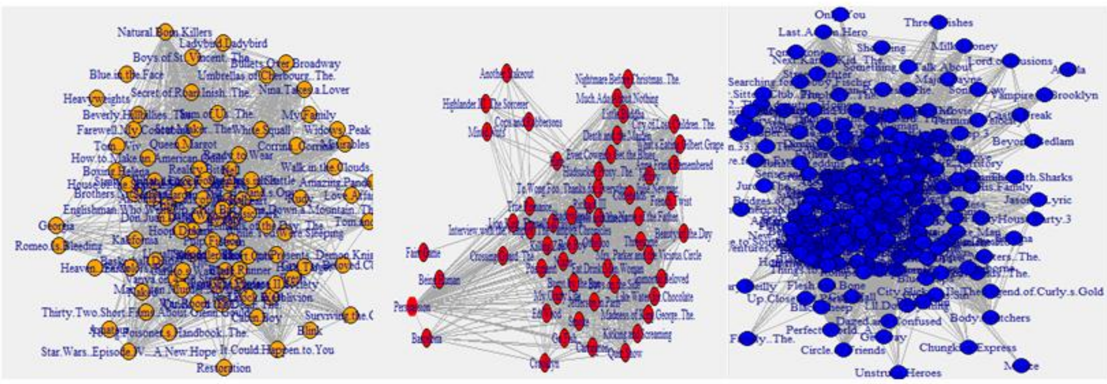


Figure 9: Major communities identified in the dataset, where a node is a movie and link represents the co-relation between movies.

In this framework each dataset is divided into a training set and a test set to determine the impact of the training size on the quality of the recommended movies. We propose to use the cross validation technique, where the original sample is partitioned randomly into k equal sized subsamples. Out of subsamples, one subsample is retained as the validation data for testing the model, and the remaining subsamples are used as the training set. We propose two scenarios, in which the first scenario selects 90% of user ratings and the second scenario select 40% as instances and the remaining ones will be used in the testing set. There are many well-known metrics in the literature, the Mean Absolute Error (MAE) and the Root Mean Square Error $(RMSE)$, precision, coverage and F- measure to measure the prediction accuracy of our proposed framework are defined as follows:

$$(MAE) = \frac{1}{T}\sum_{i,j}\left|R_{ij} - \overline{R}_{ij}\right| \quad (3)$$

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

$$precision = 1 - \frac{RMSE}{4} \quad (5)$$

$$coverage = \frac{S}{N} \quad (6)$$

$$F - measure = \frac{2 * precision * coverage}{precision + coverage} \quad (7)$$

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 ratings.

Table 5: Accuracy comparisons

Algorithms	RMSE	MAE	precision	Recall	F-measure
Item-based CF	1.271	0.942	0.844	0.7264	0.756
SNCF	1.254	0.887	0.896	0.742	0.796
CBCR	1.243	0.872	0.912	0.889	0.861

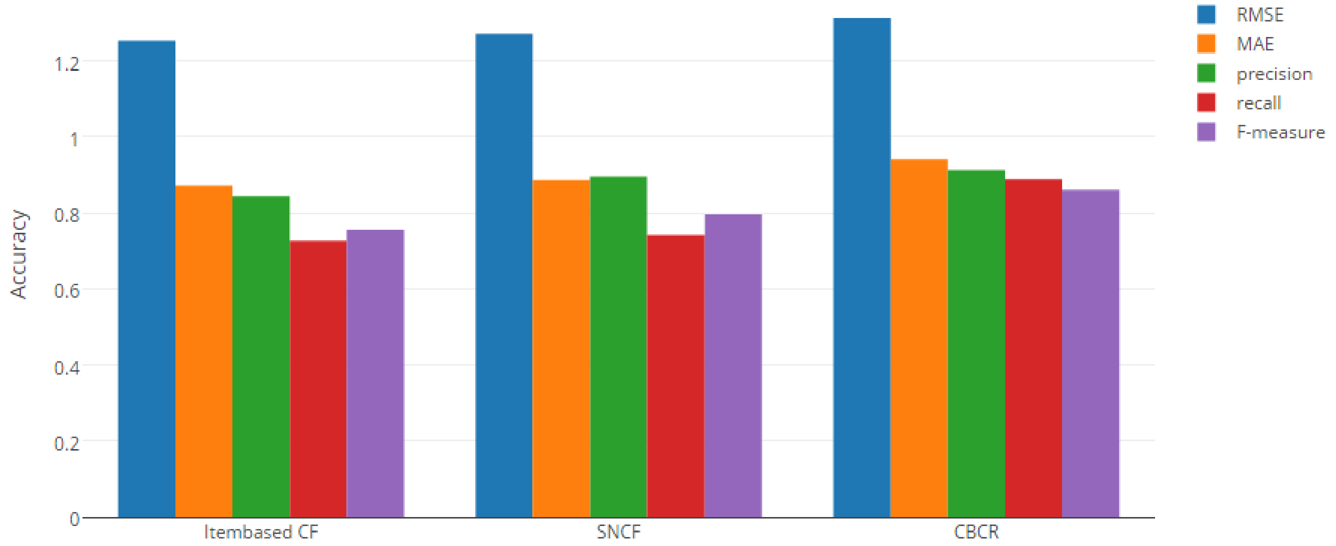


Figure 10: Accuracy comparision

CONCLUSION

In this paper, we proposed a Community-based Collaborative Filtering approach that combines community detection and recommender system to alleviate from cold-start and sparsity problems. This approach explores the users' preferences from the same group of users (i.e., community) for recommending an item, since a group or community views as similar tastes or preferences. The experiment results show that our approach CBCR outperformed than traditional approach and static community algorithms based on top-k neighborhood of the entire dataset. As a forthcoming work, we will explore the similarity computation of trusted users pertaining to the same community.

REFERENCES

[1] Fatemi, Maryam, and Laurissa Tokarchuk. "A Community Based Social Recommender System for Individuals & Groups." *Social Computing (SocialCom), 2013 International Conference on*. IEEE, 2013.

[2] Zhang, Xi, et al. "TopRec: domain-specific recommendation through community topic mining in social network." *Proceedings of the 22nd international conference on World Wide Web*. ACM, 2013.

[3] G. Zhao, M. L. Lee, W. Hsu, W. Chen, and H. Hu. Community-based user recommendation in uni-directional social networks. In Proceedings of the 22nd ACM international conference on Conference on information & knowledge management, pages 189-198.ACM, 2013.

[4] Dong, Jiali, Xiaoyong Li, and Binxing Fang. "Community-Based Recommendations: A Solution to the Vulnerability Problem." *Semantics, Knowledge and Grids (SKG), 2016 12th International Conference on*. IEEE, 2016.

[5] He, Chaobo, et al. "A topic community-based method for friend recommendation in online social networks via joint nonnegative matrix factorization." *Advanced Cloud and Big Data, 2015 Third International Conference on*. IEEE, 2015.

[6] Lalwani, Deepika, Durvasula VLN Somayajulu, and P. Radha Krishna. "A community driven social recommendation system." *Big Data (Big Data), 2015 IEEE International Conference on*. IEEE, 2015.

[7] Golbeck J (2006a) Generating predictive movie recommendations from trust in social networks. Trust Manag, pp 93–104.

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

[9] Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng 17(6):734–749.

[10] Angadi, Anupama, and P. Suresh Varma. "Overlapping Community Detection in Temporal Networks." *Indian Journal of Science and Technology* 8.31 (2015).

[11] Xie J, Szymanski BK. Labelrank: A stabilized label propagation algorithm for community detection in networks.IEEE Network Science Workshop; West Point, NY. 2013. p.138–43.

[12] Girvan M, Newman MEJ. Community structure in social and biological networks. Proceedings of National national Academy of Science; USA. 2002; p. 7821-6.

[13] Sahebi, Shaghayegh, and William W. Cohen. "Community-based recommendations: a solution to the cold start problem." *Workshop on recommender systems and the social web, RSWEB*. 2011.

[14] Zhao, Gang, et al. "Community-based user recommendation in uni- Directional social networks." *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. ACM, 2013.

[15] Wu HC, Luk RWP, Wong KF, Kwok KL (2008) Interpreting tf-idf term weights as making relevance decisions. ACM Trans Inf Syst (TOIS) 26(3):13

[16] Girvan M, Newman MEJ. Community structure in social and biological networks. Proceedings of National national Academy of Science; USA. 2002; p. 7821-6.

[17] Palla G, Derenyi I, Farkas I, Vicsek T. Uncovering the overlapping community structure of complex networks in nature and society. Nature. 2005; 435:814–8.

[18] Blondel, Vincent D., et al. "Fast unfolding of communities in large networks." *Journal of statistical mechanics: theory and experiment* 2008.10 (2008): P10008.

[19] Gregory, Steve. "Finding overlapping communities in networks by label propagation." *New Journal of Physics* 12.10 (2010): 103018.

[20] Abdrabbah, Sabrine Ben, Raouia Ayachi, and Nahla Ben Amor. "Collaborative Filtering based on Dynamic Community Detection." *Dynamic Networks and Knowledge Discovery* (2014): 85.