Collaborative Filtering based on Dynamic Community Detection

Sabrine Ben Abdrabbah, Raouia Ayachi, and Nahla Ben Amor

LARODEC, Université de Tunis, ISG Tunis,
2000, Bardo, Tunisia
abidrabbah.sabrine@gmail.com,raouia.ayachi@gmail.com,nahla.benamor@gmx.fr

Abstract. *With the increase of time-stamped data, the task of recommender systems becomes not only to fulfill users interests but also to model the dynamic behavior of their tastes. This paper proposes a novel architecture, called Dynamic Community-based Collaborative filtering (D2CF), that combines both recommendation and dynamic community detection techniques in order to exploit the temporal aspect of the community structure in real-world networks and to enhance the existing community-based recommendation. The efficiency of the proposed D2CF is dealt with a comparative study with a recommendation system based on static community detection and item-based collaborative filtering. Experimental results show a considerable improvement of D2CF recommendation accuracy, whilst it addresses both of scalability and sparsity problems.*

Keywords: Recommendation systems, Collaborative filtering, Dynamic community detection, Time varying graphs

1 Introduction

Several types of recommenders have been proposed and can be categorized into three major categories [16], namely content-based filtering, collaborative filtering, and hybrid approaches.

In this work, we focus on collaborative filtering which predicts users interests/preferences from those of remaining users sharing similar tastes. Collaborative filtering is considered as one of the most used techniques due to its efficiency and its high accuracy . Typically, the recommendation process for this technique starts when users express their preferences by rating items. The system analyzes these ratings to determine the exact preferences of the user, then, matches the active user's preferences and the preferences collection to discover the category of users having similar taste with the active user. Finally, the system recommends a set of items for the active user according to the preferences of their similar users.

From another side, the community detection presents a growing interest for many researchers, especially in web applications. A panoply of community detection algorithms exist in literature and most of them focus on static community detection, but recently the dynamic aspect of networks has sparked a new line of research. Static community detection algorithms have been explored in collaborative filtering, based on the idea that community structure will enhance the performance of recommendations [1, 3–6]. Nevertheless, these ones are not able to deal with the dynamic aspect of real-world networks.

The purpose of this paper is to model the dynamic behavior of users in recommender systems. We assume that users behaviors are learned from users' ratings data and more specifically by looking firstly at the items which have been rated by each user and then finding how to link items to each other over time in order to build the dynamic network of items. Our major contribution consists in presenting a novel approach named Dynamic Community-based Collaborative Filtering (denoted D2CF for short) capturing dynamic communities of items which present the evolution of users interests and preferences over time to over recommendations more suitable for real-world networks.

This paper is organized as follows: Section 2 presents the basic concepts of recommendation and community detection. Related work is provided in Section 3. Our proposed architecture D2CF is presented in Section 4. Section 5 is dedicated to the experimental study.

2 Basic concepts for recommendation and community detection

This section gives a brief overview on both recommendation systems and community detection.

2.1 Recommendation

Recommender systems have been proposed to address the information overload problem by filtering the relevant data and suggesting items of potential interest to users. Formally, in a typical recommendation system, there is a set of users U and a set of items I. The task of recommender system is to predict user's preferences P for each item in I. The output is a list L containing items with the highest preference values. Content-based filtering, collaborative filtering and hybrid approaches are three major categories of recommendation methods:

- *Content-based approach* selects items based on their content along user's profile. Its principle is to recommend items similar to the ones that the user has preferred in the past.
- *Collaborative filtering approach* infers user's preferences from remaining users having similar tastes. Methods pertaining to this approach can be divided into user-based and item-based methods.
- *Hybrid approach* combines content-based and collaborative filtering methods, in other words it takes into account both the users and the items properties.

In this work, we will focus on the most widely used recommendation method, namely Item-based Collaborative Filtering [19]. In such a case, the user's preference is predicted on an item using the average ratings of similar items by the same user as expressed in Equation (1) where S represents the most similar items to the item i, $s(i, j)$ denotes the similarity degree between items i and j and $r_{u,j}$ corresponds to the rating of user u on item j.

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

The similarity value can be calculated in many ways. Common methods are *Cosine similarity* and *Pearson Correlation* [19].

Due to the nature of the data used in collaborative filtering, this approach suffers, as the case of all methods, from one or more weaknesses such as the cold start problem when a new user starts with an empty profile, the sparsity problem occurring when available data are insufficient for identifying similar users, and the scalability problem when there is an excessive information of users and items. To overcome these problems, several recommendation methods have been implemented using different techniques. We cite in particular, clustering techniques [13], Bayesian techniques [12] and community detection techniques [1–4].

2.2 Community detection

Community detection techniques aim to find subgroups where the amount of interactions inside the group is more than the interaction outside it, and this can help to understand the collective behavior of users.

The communities identification process depends on the nature of networks, either static or dynamic. Static networks are basically constructed by aggregating all observed interactions over a period of time and representing it as a single graph. Dynamic networks, also called time varying graphs can be either a set of independent snapshots taken at different time steps [7, 8] or a temporal network that represents sequences of structural modifications over time [9]. In what follows, we present both static and dynamic community detection.

Static community detection: A panoply of community detection algorithms exists in literature. The first idea using static networks was proposed by Girvan and Newman [15]. It is based on a modularity function representing a stopping criterion, aiming to obtain the optimum partitioning of communities. In the same context, Guillaume et al. [11] have proposed *Louvain* algorithm to detect communities using the greedy optimization principle that attempts to optimize the gain of modularity. Rosvall and Bergstrom [17] have presented *Infomap*, considered as a solution to the simplest problem of static and non-overlapping community detection. The mentioned algorithms are not able to detect overlapping communities where a node can belong to more than one community in the same time. To ensure this basic property, Palla et al. [14] have proposed the *Clique-Percolation Method (CPM)* to extract communities based on finding all possible k-cliques in the graph. This method requires the size of the cliques in input.

Dynamic community detection: Several researchers explored the dynamic aspect of networks to identify communities structure and their development over time. Hopcroft et al. [7] have proposed the first work on dynamic community detection which consists in decomposing the dynamic network into a set of snapshots where each snapshot corresponds to a single point of time. The authors applied an agglomerative hierarchical method to detect communities in each snapshot and then they matched these extracted ones in order to track their evolution over time. Palla et al. [8] have used the (CPM) method of static community detection to extract communities from different snapshots. Then, they tried to look for a matching link between them to detect their structural changes over time. Methods applying static algorithms on snapshots cannot cover the real evolution of communities structures over time because it seems harder for these methods to recognize the same community from two different time steps of network.

To overcome this problem, new studies have exploited another representation of data that takes into account all temporal changes of the network in the same graph. We cite, in particular, the intrinsic Longitudinal Community Detection *(iLCD)* algorithm proposed by Cazabet et al. [9]. The algorithm uses a longitudinal detection of communities in the whole network presented in form of a succession of structural changes. Its basic idea was inspired from multi-agent systems. In the same context, Nguyen et al. [10] have proposed *AFOCS* algorithm to detect overlapping communities in a dynamic network N composed of the input network structure N_0 and a set of network topology changes $\{N_1, N_2, ..., N_n\}$.

3 Related works

There has recently been much research on merging community detection and recommender systems in order to provide more personalized recommendations related to users belonging to the same community. In fact, community-based recommendation is a two-step approach. The first step consists in identifying groups in which users should share similar properties and the second step uses the community into which the target user pertains to recommend new items.

Using static community detection algorithms, Kamahara et al. [2] have proposed a community-based approach for recommender systems which can reveal unexpected user's interests based on a clustering model and an hybrid recommendation approach. In the same context, Qin et al. [3] have applied CPM method on the Youtube Recommendation Network of reviewers to detect communities of videos. These latters are used to provide the target user by a local recommendations which consists in recommending videos pertaining to the same community of the video watched by him. This approach aims to propose a more diverse list of items for target user. Another aim behind incorporating community detection to recommendation is to provide a solution to the cold start problem, and this idea was proposed by Sahebi et al. [4] while applying *Principal Modularity Maximisation* method to extract communities from different dimensions of social networks. Based on these latent communities of users, the recommender system is able to propose relevant recommendations for new users.

Qiang et al. [1] have defined a new method of personalized recommendation based on multi-label Propagation algorithm for static community detection. The idea consists in using the overlapping community structures to recommend items using collaborative filtering. More recently, Zhao et al [5] proposed the *Community-based Matrix Factorization (CB-MF)* method based on communities extracted using *Latent Dirichlet Allocation method (LDA)* on twitter social networks. In [6], authors focused on community-based recommendation of both individuals and groups. They used the Louvain community detection method on the social network of movies building from the Internet Movie Database (IMDb) in order to provide personalized recommendations based on the constructed communities.

These methods only deal with static networks, derived from aggregating data over all time, or taken at a particular time. The accumulation of an important mass of data in the same time and in the same graph can lead to illegible graphs, not able to deal with the dynamic aspect of real-world networks. To take into account the evolution of users behaviors over time using a kind of community-based dynamic recommendation, a first attempt was established by Lin et al. [16]. The main idea consists in providing a dynamic user modeling method to make recommendations by taking into consideration the dynamic users' patterns and the users' communities. This approach is limited since it uses a manual method to identify communities, which is not efficient especially when we deal with strongly evolving and large networks. More recently, Abrouk et al. [20] proposed to use the fuzzy k-means clustering from time to time to dynamically detect the users' interests over time. Then, they exploited these formed communities to determine user's preference for new items with regard to the updated users' ratings. In [21], author proposed an article recommender system to recommend documents for users based on the same members of communities which are identified according to their interests while browsing the web. The detection of users' interests is repeated continuously in subsequent time intervals in order to deal with the dynamic aspect of new portals. In both the previously presented methods, applying clustering techniques for time to time cannot cover the real evolution of community structure over time. In fact, several structural changes may occur and get lost without being detected. Besides, the temporal complexity of these methods increases in large networks.

Aiming to benefit from the whole advantages of the dynamic community detection process as part of recommender systems, we propose a global architecture allowing to ensure this combination as detailed below.

4 Proposed architecture

The proposed architecture, called *Dynamic Community-based Collaborative Filtering*, denoted by D2CF, is based on three main steps as shown in Figure 1.

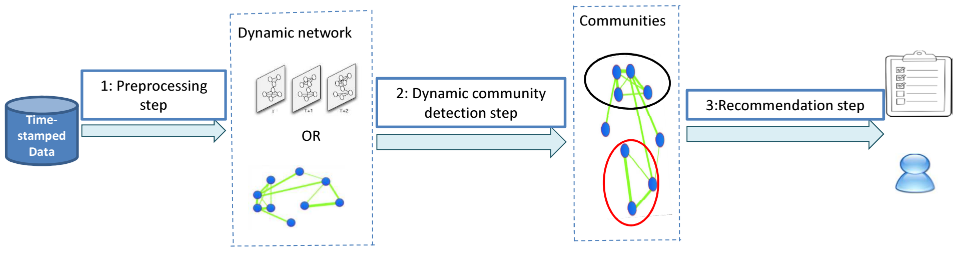

Fig. 1. D2CF architecture

4.1 The pre-processing step

This step consists in building a dynamic network in which the evolution of the users' interests (i.e. the set of items be looked at or ranked) over time is represented. We assume that if a user does not rate a given item then this latter is not yet watched by him. It is important to note that, in this work, the nodes are the items and not the users and hence the communities are groups of items. The interactions between nodes are modeled using the co-ratings relationship. Indeed, two nodes interact with each other if at least one user gives the same rating to both of them in the same time. With the aim to prepare the list of network changes over time, we have adapted the method of temporal network building [18] (i.e. An edge is established between two nodes if these ones have interacted with each others at least N times over a period of P days). The values of N and P control the set of edges adding and removal in the network and depend not only on the network topology but also on the information that we want to extract in order to create semantic links between the network nodes (e.g. messages interactions seem to be more intense than calls interactions). If over a period of P days, there are fewer than N interactions between two nodes, the edge will be automatically removed. These changes (edges removal and edges creations) can be represented either in the same graph as temporal network or in different graphs as a sequence of snapshots where each one corresponds to network changes over a specific time period (i.e. one hour, one day, ect.). The resulting dynamic network of items is considered as a generic model that represents the evolution of users interests over time.

4.2 Dynamic community detection step

Once the dynamic network (temporal network or a sequence of snapshots) is constructed in the pre-processing step, we are now able to use it as input to dynamic community detection method. Based on the network interactions, a community can be defined as a set of items that are extremely related to each other physically (strongly connected) and semantically (a learning pattern of items that tends to have the same interests of several users over a period of time). The advantage here is that a community is not restricted to item-related topics but it contains various topics (e.g. Horror, Comedy, Romance, ect.). The novelty in this work is to apply a dynamic community detection algorithm that takes into account the evolution of the network behavior in order to obtain a more appropriate community structure. To ensure this step, we choose to use the *iLCD* algorithm [9]. This choice is justified by the fact that *iLCD* detects communities in both static and dynamic networks depending on disposal data. Moreover, this algorithm takes into account the evolution of the network which enables to identify the dynamical communities more accurately. Such community detection can be more powerful because this analysis matches with the reality of networks. The algorithm deals with both large evolving networks and the overlap of communities. Extracted communities are atomic in the sense that there are not other relevant communities inside it. Finally, all operations in *iLCD* are made at a local level (i.e. a community can interact with only the nodes that are linked to and having at least one node in common), which can allow to ensure that the complexity will not grow exponentially with the size of the network. The life cycle of a community may be described via three phases:

– A new community is born when a new clique is formed in the graph.
– A modification of an internal community can lead to merge two communities having at least one node in common or to split the main community into two new communities.
– A community dies when there is no disposal nodes in its structure.

4.3 Recommendation step

In this step, the learned patterns (communities) will be exploited to help the recommender system to predict the users' future interests based on certain categories given by these communities. Firstly, a target item should be identified for each user. The target item in this case is the item in which the active user is more interested (The item with the highest rating value). The items that belong to the communities of target item and that are not rated by the active user are selected as candidate items. The list of top k recommended items for the active user contains the k candidate items that have the highest predicted preferences. Our objective is to compute the preference of the user u on the candidate item i based on the items that belong to the communities of i. By doing this the recommendation is restricted on the communities to which the candidate item pertains. To this end, we propose an adaptation of the traditional item-based Collaborative filtering method (See Equation (1)). In fact, instead of computing user's preferences taking into consideration all items present on the whole network to discover the most similar items to the candidate item, we only rely on the items that pertain to candidate item's communities extracted from the dynamic network as shown in Figure 2.

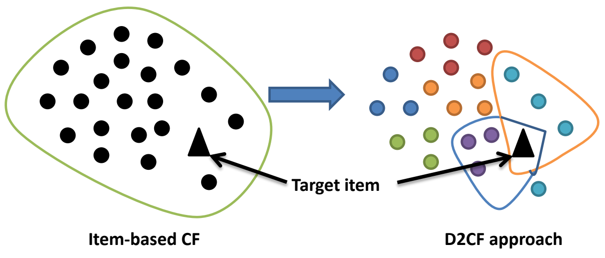

Fig. 2. The principle of user's preference computation taking into account the community structure in the network

Thus, we can formally define the preference prediction as follows:

$$P_{u,i} = \frac{\sum_{j \in C} s(i,j)\ r_{u,j}}{\sum_{j \in C} |s(i,j)|} \tag{2}$$

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

We propose to compute the similarity $s(i,j)$ using the *Pearson correlation similarity* measure [19].

Finally, the recommendation list contains the candidate items i having the highest preference values $P_{u,i}$.

In the case where the user is new (no ratings history), the target item of this user is learned by browsing item in the recommender system. We select then the candidate items that belong to the communities of the target item. The recommendation list contains the candidate items which are ranked according to their similarities relative to target item.

5 Experimental study

To evaluate the effectiveness of D2CF, we propose to use the movieLens dataset available through the movieLens website (http://movieLens.umn.edu). This dataset contains in total 100.000 ratings collected by 943 users on 1682 movies, from 19-09-1997 to 22-04-1998. The score of rating is ranged from 1 to 5. Each user has rated at least 20 movies. The ratings information are timestamped. We suppose that each movie is represented by a node and a community is defined as a set of nodes. If a user rates a movie, this means that he is interested in watching it. MoviLens data are represented as a sequence of temporal events in the following way:

- user U_1 rates movie I_1 with 5 at T_1,
- user U_2 rates movie I_1 with 3 at T_1,

– user U_2 rates movie I_5 with 5 at T_2, etc.

To experimentally determine the impact of the training size on the quality of the recommended movies, we propose to test three scenarios:

– *Set_1* : For each user, we randomly select 90% of his ratings as instances in the training set and the remaining ones will be used in the testing set.
– *Set_2* : For each user, we randomly select 40% of his ratings as instances in the training set and 10% will be used in the testing set.
– *Set_3* : For each user, we randomly select 20% of his ratings as instances in the training set and 10% will be used in the testing set.

The data selection should take into account the evolution over time, so that, instances of the testing set should be chosen after those of the training set. We run 10-fold cross-validation on data. The precision is defined to evaluate the validity of a given recommendation list and it is formulated to detect the average of the true recommendations relative to the total number of the proposed recommendations. While the recall metric is defined as the ratio of the number of recommended objects collected by users appearing in the test set to the total number of the objects actually collected by these users.

The implementation of the whole system needs several parameters. Our choices regarding the three steps of D2CF can be summarized as follows:

1. *Pre-processing step*: Our idea is to extract movies interactions in such a way that we know what a movie has been assessed with another one with the same score by the same producer in the same time. In fact, if an interaction between two movies occurred more than N times over a period of P days, an edge is established between them. We define the values of P and N such a way that we conserve more links between nodes. After performing several tests on the movieLens data, we set P and N respectively to 200 and 30 for *Set_1*, 200 and 20 for *Set_2* and 200 and 5 for *Set_3*. This means that, in *Set_1*, an edge is established between two nodes, if these ones have interacted at least 30 times over a period of 200 days and this edge is removed if less than 30 interactions have occurred between them over a period of 200 days after the edge creation date. The movies that are not very visible in the users' ratings behaviors are considered as outliers. The outliers are the nodes that are disconnected of the core of the network due to their low interactions with other movies (i.e. there are less than N users who give the same rating for both of movies).
2. *Dynamic community detection step*: In this stage, we are able to apply any state of art dynamic community detection algorithm. In this experiment we choose to use *iLCD* algorithm to extract communities from the temporal network built above. Since the quality of resulting communities depends on the threshold (i.e. is a parameter using as input in the community detection step to determine the belonging or not of new added nodes in a community), we choose the value 0.5 to obtain by the end of the process, overlapping, small and dense communities.

3. *Recommendation step*: Using detected communities, we are now able to generate the top k recommendation list of movies to the active user. This step requires both the user's ID to look for his target item and the community structure as input parameters to select the candidate items that may interest the active user. Then, the predicted preference of each candidate item is computed using *Pearson correlation-based similarity* measure. The items which have the top k preference predictions are recommended to the active user.

In order to evaluate both of the effectiveness and efficiency of our proposed D2CF approach we compare the performance of this one with the following methods:

– The Static Community-based Collaborative Filtering (denoted S2CF for short), which is a static version of our proposed architecture. We keep the same parameters used for the dynamic network without taking into account the temporal dimension. This is possible since, as mentioned before, the *iLCD* algorithm allows both static and dynamic community detection.
– The traditional item-based Collaborative Filtering with *Pearson correlation-based similarity* measure available from Apache mahout library in Java.

The obtained results are summarized in Table 1. We can notice that our D2CF method outperforms traditional recommendation methods: item-based collaborative filtering and Collaborative filtering based on static community detection. In fact in Set_3, our approach is able even with small set of users' data to provide users with a wealthy and varied recommendation list based on the communities of users' interests. The recall and precision values for both item-based and static community-based collaborative filtering decrease as we increase the training set size but our approach combining recommendation and dynamic community detection still provide better recommendation quality as shown in Figure 5. In fact, D2CF gives its best results (i.e. D2CF proposes for one user an average of 6 good items out of every 10 recommended items while S2CD offers an average of two good items out of every 10 recommended items and finally item-based collaborative filtering gives an average of 0,15 good items per user). These results show that our approach is the best in the case of scalable data, which is explained by the fact that the dynamic network learned by more users' data better performs the prediction of users' preferences for unseen items.

Table 1. Precision and Recall values for Set_1, Set_2 and Set_3

	Set_1		Set_2		Set_3	
Approach	Precision	Recall	Precision	Recall	Precision	Recall
D2CF	**0.603**	**0.687**	**0.3**	**0.49**	**0.223**	**0.34**
S2CF	0.2	0.26	0.21	0.195	0.197	0.221
Item-based CF	0.015	0.02	0.084	0.091	0.18	0.2

D2CF approach presents a significantly improvement on recommendation on both small and large sets. We can say that this approach addresses both

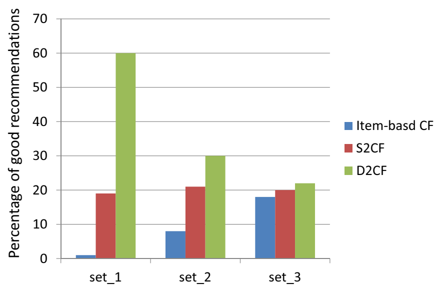

Fig. 3. Impact of the dataset size on the recommendation quality

scalability and sparse problems and it is able to handle the real-world networks by providing a dynamic recommendation based on dynamic communities.

6 Conclusion

In this paper, we propose a Dynamic Community-based Collaborative Filtering approach that combines recommendation and dynamic community detection. This approach is able to deal with real-world networks as it takes into account the evolutionary aspect of the users' interests over time. The experimental results show that our proposed D2CF outperforms both of item-based collaborative filtering and collaborative filtering based on static communities. As a future work, we will explore the similarity computation process of users pertaining to the same community in the recommendation context.

References

1. Qiang, H., Yan, G.: A method of personalized recommendation based on multi-label propagation for overlapping community detection. In: the 3rd International Conference on System Science Engineering Design and Manufacturing Informatization, 1, pp. 360–364. October (2012)
2. Kamahara, J., Asakawa, T., Shimojo, S., Miyahada, H.: A commynity-based recommendation system to reveal unexpected interests. In: Multimedia Modelling Conference, pp. 433–438. January (2005)
3. Qin, S., Menezes, R., Silaghi, M.: A recommender system for youtube based on its network of reviewers. In: the IEEE International Conference on Social Computing, (2010)
4. Sahebi, S., Cohen, W.: Community-based recommendations: a solution to the cold start problem. In RSWEB'11, (2011)
5. G. Zhao, G., Lee, M. L., Hsu, W., Chen, W., Hu, H.: Community-based user recommendation in uni-directional social networks. IN: the 22th ACM international

conference on Conference on information knowledge management, pp. 189–198. October 2013.

6. Fatemi, M., Tokarchuk, L.: A community based social recommender system for individuals groups. In: the 2013 International Conference on Social Computing (SocialCom'13), pp. 351–356. Sept (2013)
7. Hopcroft, J., Khan, O., Kulis, B., Selman, B.: Tracking evolving communities in large linked networks. In: the national academy of sciences of the United States of America, 1, pp. 5249–5253. (2004)
8. Palla, G., Barabasi, A., Vicsek, T.: Quantifying social group evolution. In: Nature, 446, pp. 664–667. April (2007)
9. Cazabet, R., Amblard, F.: Simulate to detect: a multi-agent system for community detection. In: The 2011 ACM International Conference on Web Intelligence and Intelligent Agent Technology(WI-IAT), 2, pp. 402–408. August 2011.
10. Nguyen, N., Dinh, T., Tokala, S., Thai, M.T.: Overlapping communities in dynamic networks: Their detection and mobile applications. In: the 17th annual international conference on Mobile computing and networking (MobiCom'11), pp. 85–96. Sept (2011)
11. Blondel, V., Guillaume, J., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in larges networks. Journal of Statical Mechanics : Theory and Experiment. (2008)
12. Condliff, M. K., Lewis, D. D., Madigan, D.: Bayesian mixed-effects models for recommender systems. In: ACM SIGIR'99 Workshop on Recommender Systems: Algorithms and Evaluation, (1999)
13. Sarwar, B. M., Karypis, G., Konstan, J., Riedl, J.: Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. (2002)
14. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. In: Nature, 435(7043), pp. 814–818. (2005)
15. Newman, M., Girvan, M.: Finding and evaluating community structure in networks. Phisical review E, 69(2). (2004)
16. Song, X., Lin, C., Tseng, B., Sun, M.: Modeling evolutionary behaviors for community-based dynamic recommendation. In: the 2006 SIAM International Conference on Data Mining, (2006)
17. Rosvall, M., Bergstrom, C.: Maps of random walks on complex networks reveal community structure. In: the National Academy of sciences, 105(4), pp. 1118–1123. (2008)
18. Cazabet, R.: Dynamic Community detection on temporal networks. PhD thesis, Université Toulouse 3 Paul Sabatier. (2013)
19. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: the 10th international conference on world wide web, www'01, pp. 285–295. (2010)
20. Abrouk, L., Gross-Amblard, D., Cullot, N.: Community detection in the collaborative web. International Journal of Managing Information Technology. pp. 1–9.(2010)
21. Hönsch, M.: Detecting user communities based on latent and dynamic interest on a news portals. In: the 7th student research conference in informatics and information technologies, 3, pp. 47-50. ACM(2011)