Mathematical Problems of Computer Science 51, 57-65, 2019.

UDC 004.031.42

Realization of Recommender Framework Based on Community Detection

Karen K. Mkhitaryan

Institute for Informatics and Automation Problems of NAS RA
e-mail: karenmkhitaryan@gmail.com

Abstract

Recommender systems play an important role in suggesting relevant information to users based on their available preferences about items. Utilizing a recommender system allows companies to increase revenues, customer satisfaction and enable personalization and discovery. Content-based and collaborative filtering approaches are the most popular techniques in recommender systems predicting users preferences based on "collaborative" data about users and items in the system. However, their use is not justified in certain applications, particularly when user-item collaboration data is very sparse or missing. In this paper, a recommender framework based on community detection is developed outperforming other popular recommendation methods in some applications.

Keywords: Community detection, Recommender systems, Collaborative filtering, Similarity measures.

1. Introduction

A recommender system is a type of information filtering system predicting users' preferences about items based on user-item collected data. People listen to songs, watch movies, read books, use products, etc., and express their opinions by giving ratings, liking, disliking or reviewing items online more than ever. In recent years, various recommendation techniques and algorithms were proposed to generate recommendations based on collected user-item interaction data. Effective recommender systems increase revenues, customer satisfaction, enable personalization and discovery. Several real world examples of recommender systems are Last.fm, recommending songs, Amazon, recommending products to buyers and many more recommending people, movies, books, etc.

Despite the existence of many recommendation algorithms, there are some applications, where traditional recommendation approaches are not beneficial. This is the case when users cannot explicitly rate the items, thus “collaborative” data is not available, while content-based and collaborative filtering approaches rely only on user-item rating data. In this paper, we propose a framework based on community detection that can generate recommendations in such scenarios.

Community detection is a research area from network science dealing with the investigation of complex networks aiming to detect communities of nodes that are densely connected inside the community and sparsely connected with nodes from other communities. Detection of communities has many applications in various disciplines, such as investigating protein-to-protein interactions in biology, friendship circles in social network analysis, discovering fraudulent websites on the web, etc.

Several successful attempts have been made in the literature to incorporate community detection tools and techniques into recommender systems. Abdrabbah et al. introduced a novel architecture called Dynamic Community-based Collaborative filtering (D2CF), combining both collaborative filtering and dynamic community detection techniques. Experiments on MovieLens dataset showed that the proposed technique has considerably higher recommendation accuracy outperforming the methods based on static community detection and item-based collaborative filtering [1]. Lalwani et al. presented a social recommender system based on community detection and collaborative filtering techniques (CCSRS) using a map-reduce framework [2]. The proposed approach improves scalability, coverage and cold start issue compared with item-based collaborative filtering.

There are also other papers pointing out the possible interconnections of community detection and recommender systems [3], [4], [5]. The recommender framework suggested in this paper is based only on community detection therefore allows to apply it in the situations when other methods are useless.

The paper is constructed as follows: in Section 2 we discuss traditional recommendation approaches, introduce the framework in Section 3 and show experimental results in Section 4.

2. Recommender System

The analysis of data generated by users can be used to predict preferences or ratings a user will give to an item that he/she did not interact in the past. The most popular recommendation approaches are *content-based*, *collaborative* and *hybrid* filtering techniques (Fig.1).

In **content-based filtering**, terms or keywords (e.g., music genre, article keywords) are used to describe an item and the user profile. Content-based recommendation algorithms learn the features of items the user liked in the past and recommend items that share similar features or characteristics. The advantage of this method over collaborative filtering is that it does not need other user’s ratings and avoids the cold start problem (new items or users in the system for which the information is missing) [6].

The idea behind **collaborative filtering** is to use the “collaboration” data between users and items in the system to generate recommendations. *Memory-based* and *model-based* approaches are two types of collaborative filtering technique [7].

Memory-based approach takes the user-item rating matrix M as input, where each M_{ui} element represents the rating that the user u gave to item i and predicts the missing ratings in the matrix. *User-user filtering* and *item-item filtering* are two realizations of memory-based approach. In user-user filtering, items that users similar to the given user u like, are recommended to user u, while in item-item filtering recommendation is made using the preferences of users about other items that also liked an item the user u liked.

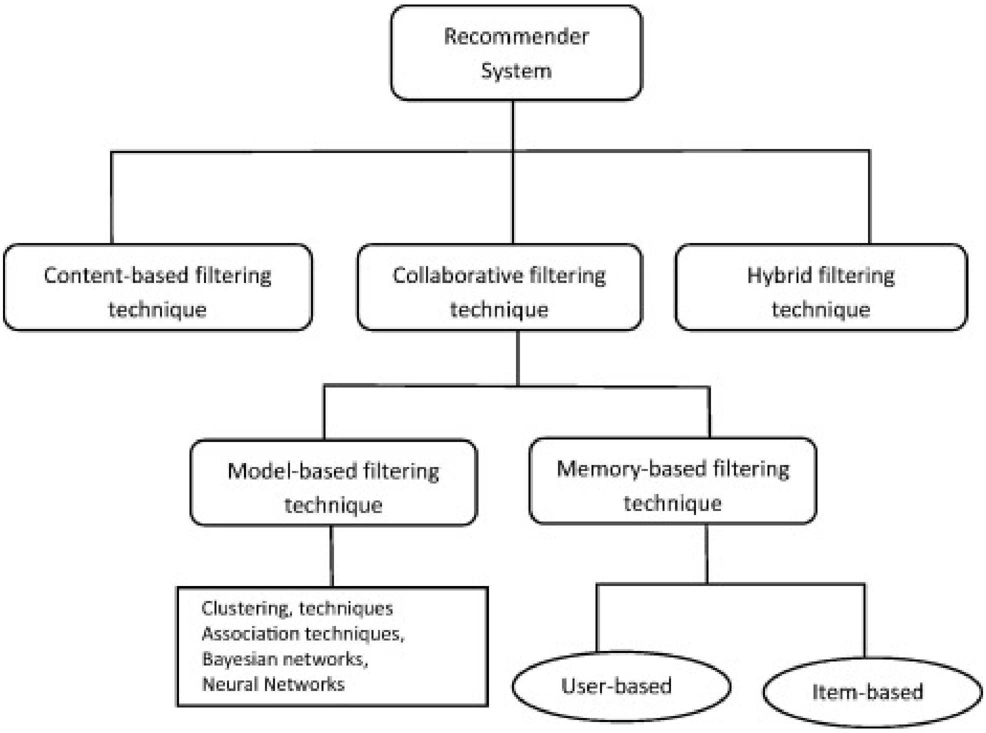
Recommender System
Content-based filtering technique
Collaborative filtering technique
Hybrid filtering technique
Model-based filtering technique
Memory-based filtering technique
Clustering, techniques
Association techniques,
Bayesian networks,
Neural Networks
User-based
Item-based

Fig. 1: Types of recommender system algorithms.

For user or item comparison, various similarity measures are used, such as Euclidean distance, cosine similarity, Minkowski distance, Manhattan distance, etc. To predict the missing rating M_{ui} in matrix M using user-user filtering we first find the weighted sum of ratings from other users

$$\sum_{q \in U} sim(u, q) M_{qi} ,$$

where U is the set of users in the system, $sim(u, q)$ is the similarity score between the users u and q, and M_{qi} is the rating the user q gave to item i. Finally, the predicted rating is obtained by normalizing the above weighted sum:

$$M_{ui} = \frac{\sum_{q \in U} sim(u, q) M_{qi}}{\sum_{q \in U} |sim(u, q)|}.$$

Item-item filtering uses the same idea but instead of user similarities, item similarities need to be calculated.

Model-based approaches use clustering, matrix factorization and deep learning techniques to make recommendations. Well-known K-means clustering technique can be used to find the top k similar users or items from user-item rating matrix to predict the unknown ratings. Neural networks from deep learning can also be used to build recommender systems [8]. There are also **hybrid** recommender systems that combine both memory-based and model-based approaches.

Although they are the most popular techniques in the literature, they have some common drawbacks in real life applications listed below:

- **Collaborative filtering**

  Data sparsity: Users do not rate too many items, therefore user-item rating data is sparse containing many unknown ratings.

  Scalability: With millions or billions of users or items, it is a computationally extensive task to run recommendation algorithms.
- **Content-based filtering**

  Over-specialization: Recommending items that are very similar to what the user liked, limits the novelty, and the recommended item is not interesting to the user.

  New user problem: When a new user enters the system or the user profile is empty, it is a difficult task to predict his/her preferences.

3. Community Detection-Based Recommender Framework

There are some applications when it is not possible to obtain the user-item rating data or to compare the items in the system. In that case, collaborative and content-based approaches will fail to make recommendations.

We develop a community detection recommender framework that is able to generate recommendations in scenarios when user-item rating data is very sparse or missing and/or item comparison cannot be accomplished using community detection techniques. The main steps of the framework to generate recommendations to the target user are as follows:

Step 1 - Feature identification. To be able to quantitatively compare users in the system and obtain the similarity scores, feature identification is needed. Features represent attributes about users that can be acquired in different ways, such as survey questions, past user records, etc.

Step 2 - Ground-truth network construction. Based on predefined features, users can be compared with each other using similarity measures designed for user or item comparison. To make recommendation to the target user, we first need a ground-truth user data, i.e., users that have already shown their preferences. Using similarity scores a weighted ground-truth network is constructed, where nodes and weighted edges represent the users in the ground-truth and similarities between them, respectively.

Step 3 - Community detection algorithm validation. In the literature of community detection, various algorithms were designed for weighted networks. In [9] we studied the subsequent state-of-the-art algorithms that will be applied in this paper: multilevel modularity optimization [10], fast greedy modularity optimization [11], Infomap [12], Walktrap [13], Label propagation [14] and Edge betweenness [15].

The algorithm takes the ground-truth network as input and partitions it into a community structure. In order to validate the results of algorithms, their detected community structures are compared with a real community structure, i.e., communities based on preferences of users. For evaluation, there are also well-designed measures for network partition comparison. A new modified χ^2-divergence measure was suggested [16], where its advantages are discussed. We use this measure to validate an algorithm for the next step by comparing its detected community structure with the real one.

Step 4 - Network updating including the target user. In this step, a new user enters the system (in the network) and based on predefined features, similarities between the new user and the users in the ground-truth are calculated using the similarity measures. Next the ground-truth network is updated including the target user.

Step 5 - Community detection and recommendation. The validated community detection algorithm is applied to an updated network, and the target user is assigned to a community with users sharing similar preferences. Recommendation to the target user is made considering the preferences of users from the same community. Items that are mostly preferred by the users in the community are recommended to the target user.

4. Experimental Results

Consider an example of recommender system using the proposed framework that recommends professions or career paths to people. Traditional recommendation techniques will require preference data of users about professions in order to generate recommendations, while in reality it is impossible for a person to deal with many professions to share preference. Developed framework considers the similarities of people rather than their preferences and generates recommendations based on the preference of other people being similar to the target person. Experiment is implemented on an artificially created dataset (ground-truth) containing 100 users and their answers to 100 survey questions. We considered that an answer to a question is an integer value from 1 (strongly disagree) to 5 (strongly agree). Let Q be the generated matrix containing 100 rows (users) and 100 columns (number of questions), where each $Q_{ij} \in \{1,2,3,4,5\}$ element represents the answer that the user i gave to the question j. We also consider that the ground-truth network has 10 communities, and each community contains 10 people. Next, the ground-truth network is constructed where nodes and edges represent the users and similarity scores between them, respectively. We used adjusted cosine similarity to measure the similarity between two users defined as:

$$sim(u,v) = \frac{\sum_i (Q_{ui} - \mu_u)(Q_{vi} - \mu_v)}{\sqrt{\sum_i (Q_{ui} - \mu_u)^2}\sqrt{\sum_i (Q_{vi} - \mu_v)^2}},$$

where Q_{ui} and Q_{vi} are the answers of users u and v to question i, and μ_u and μ_v are the average of all the answers to 100 questions of users u and v, respectively. In Fig. 2 (left) the constructed ground truth network is shown, composed of 100 user nodes and 5050 edges, where the weight of an edge between users u and v is the similarity score obtained using adjusted cosine similarity, $sim(u,v)$.

Next six community detection algorithms, discussed in the previous section, are applied to partition the network into a community structure. Let $X = (x_1, x_2, \dots, x_N)$ and $Y = (y_1, y_2, \dots, y_N)$ be two partitions of the network with N nodes into K_X and K_Y number of communities, respectively, where each $x_i \in \{1, \dots, K_X\}$ and $y_i \in \{1, \dots, K_Y\}$, $i \in \{1, \dots, N\}$ denotes the community label of node i in partitions X and Y, respectively. An information theoretic measure is called modified χ^2-divergence defined as:

$$MD_{\chi^2}(X,Y) = 1 - \frac{\sum_{x,y} \frac{p^2(x,y)}{p(x)p(y)} - 1}{\max(K_X, K_Y) - 1},$$

where $p(x,y)$ is the joint probability function of X and Y, and $p(x)$ and $p(y)$ are the marginal probability distribution functions of X and Y, respectively, and can be used to quantitatively validate the community structure detected by an algorithm, i.e., which one is most similar to real community structure.

The overview of community structures detected by all six algorithms is given in Table 1. Thus, the closest network partition to real network partition is detected by the fast greedy modularity optimization algorithm shown in Fig. 2 (right).

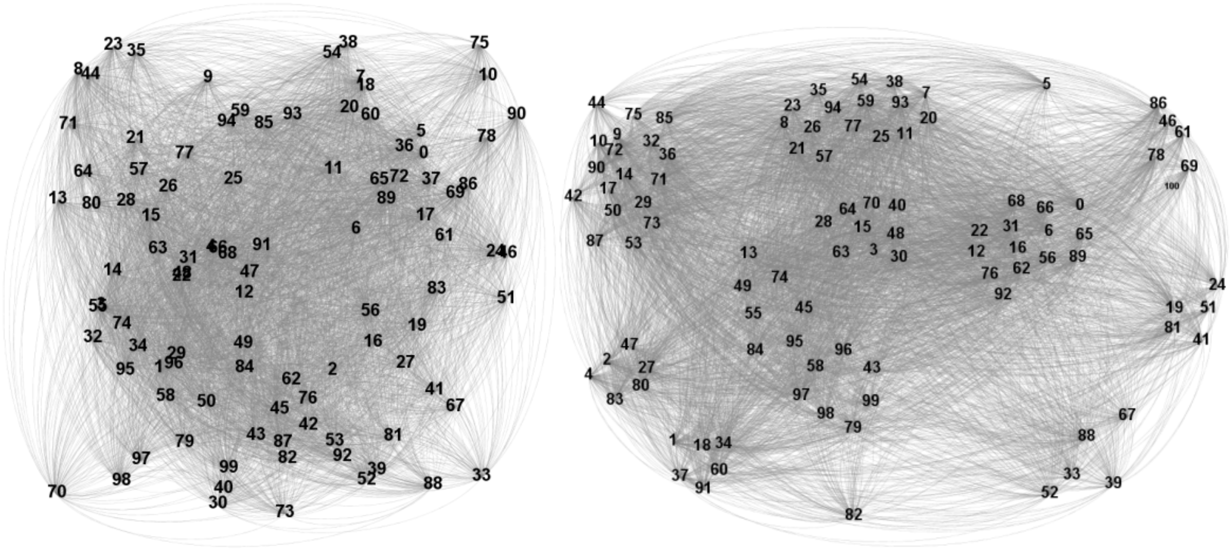

Fig.2: Weighted network composed of $\mathbf{100}$ user nodes and $\mathbf{5050}$ weighted edges (left) and community structure detected by fast greedy modularity optimization algorithm (right).

Table 1: Overview of community structures detected by six algorithms.

Algorithm	**No of communities**	**Modified χ^2-divergence**
Multilevel modularity optimization	10	0.903
Fast greedy modularity optimiation	10	0.899
Infomap	1	1
Walktrap	1	1
Label propagation	100	0.909
Edge betweenness	94	0.903

To recommend a profession to a person who has already answered 100 questions, the similarities between the target user and other users in the ground-truth network are calculated resulting in a construction of a new network composed of 101 nodes and 5151 edges. Fast greedy modularity optimization algorithm validated on ground-truth network is applied to the new network and detected 10 communities, where the 101^{th} target user lies in a community of 17 users (Fig. 3). Finally the profession selected by the majority of users among the 17 users is recommended to the target user.

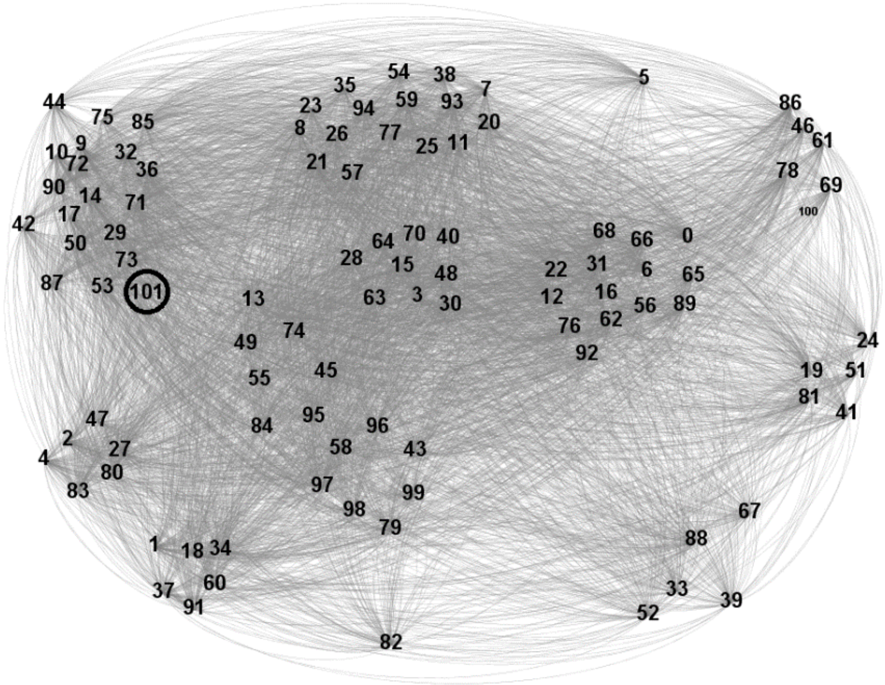

Fig. 3: Community structure of a new network including the target user (circled in white).

5. Conclusion

In this paper, we developed a community detection-based recommender framework that is able to make recommendations when user-item rating data is very sparse or missing, and/or items cannot be compared with each other. The framework was implanted into an artificially generated dataset to demonstrate how it can be used. For the work of the recommender system a real life data is needed.

References

[1] S. B. Abdrabbah, R. Ayachi and N. B. Amor, "Collaborative filtering based on dynamic community detection," in *DYNAK'14 Proceedings of the 2nd International Conference on Dynamic Networks and Knowledge Discovery*, 2014.

[2] D. Lalwani, D. V. L. N. Somayajulu and P. R. Krishna, "A community driven social recommendation system," in *2015 IEEE International Conference on Big Data (Big Data)*, DOI:10.1109/BigData.2015.7363828, 2015.

[3] W. Deng, R. Patil, L. Najjar, Y. Shi and Z. Chen, "Incorporating community detection and clustering techniques into collaborative filtering model," *Procedia Computer Science,* vol. 31, pp. 66-74, 2014.

[4] F. Gasparetti, G. Sansonetti and A. Micarelli, "Community detection and recommender systems," *Encyclopedia of Social Network Analysis and Mining,* pp. 1-14, 2017.

[5] D. Deshmuk and D. D. R. Ingle, "A community detection and recommendation system," *International Research Journal of Engineering and Technology (IRJET)* , vol. 4, no. 7, pp.

752-757, 2017.

[6] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," *The Adaptive Web,* vol. 4321, pp. 325-341, 2007.

[7] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence,* vol. 2009, no. 421425, 2009.

[8] L. Zhang, T. Luo, F. Zhang and Y. Wu, "A recommendation model based on deep neural network," *IEEE Access,* vol. 6, pp. 9454 - 9463, 2018.

[9] J. Mothe, K. Mkhitaryan and M. Haroutunian, "Community detection: comparison of state of the art algorithms," *Proceedings of International Conference Computer Science and Information Technologies,* CSIT, Yerevan, Armenia, pp. 252-258, 2017.

[10] V. D. Blondel, J.-L. Guillaume, R. Lambiotte and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment,* DOI: https://doi.org/10.1088/1742-5468/2008/10/P10008, 2008.

[11] A. Clauset, M. E. J. Newman and C. Moore, "Finding community structure in very large networks," *Phys. Rev. E,* vol. 70, no. 066111, 2004.

[12] M. Rosvall, D. Axelsson and C. T. Bergstrom, "The map equation," *The European Physical Journal Special Topics,* vol. 178, no. 1, pp. 13-23, 2009.

[13] P. Pons and M. Latapy, "Computing communities in large networks using random walks," *International Symposium on Computer and Information Sciences,* pp. 284-293, 2005.

[14] U. N. Raghavan, R. Albert and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Phys. Rev. E,* vol. 76, no. 036106, 2007.

[15] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *PNAS,* vol. 99, no. 12, pp. 7821-7826, 2002.

[16] M. Haroutunian, K. Mkhitaryan and J. Mothe, "f-Divergence measures for evaluation in community detection," in *Workshop on Collaborative Technologies and Data Science in Smart City Applications (CODASSCA 2018),* Yerevan, pp. 137--144, 2018.

Submitted 14.01.2019, accepted 16.04.2019.

Խորհրդատվական միջավայրի մշակում համայնքների հայտնաբերման միջոցով

Կարեն Կ. Մխիթարյան

ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտ
e-mail: karenmkhitaryan@gmail.com

Ամփոփում

Խորհրդատվական համակարգերը օգտատերերին ինֆորմացիա են առաջարկում՝ հիմնվելով նրանց վերաբերյալ առկա տվյալների վրա: Խորհրդատվական համակարգերի կիրառումը հնարավորություն է տալիս կազմակերպություններին ավելացնել եկամուտները, ընդլայնել ցանցը, ինչպես նաև խթանել օգտատերերի համար նոր ինֆորմացիայի բացահայտումը: Բովանդակային և համագործակցային ֆիլտրման մոտեցումները ամենահաճախ օգտագործվող մեթոդներն են խորհրդատվական համակարգերում, որոնք կանխագուշակում են օգտատերերի նախընտրությունները: Սակայն այդ մեթոդները պիտանի չեն որոշ կիրառություններում, մասնավորապես, երբ համագործակցային տվյալները քիչ են կամ բացակայում են: Այս աշխատանքում մշակվել է համայնքների հայտնաբերման վրա հիմնված խորհրդատվական միջավայր, որը որոշ կիրառություններում ունի առավելություն խորհրդատվական այլ մոտեցումների համեմատ:

Բանալի բառեր՝ համայնքների հայտնաբերում, խորհրդատվական համակարգեր, համագործակցային ֆիլտրում, նմանության չափեր:

Реализация рекомендательной среды на основе обнаружения сообществ

Карен К. Мхитарян

Институт проблем информатики и автоматизации НАН РА
e-mail: karenmkhitaryan@gmail.com

Аннотация

Рекомендательные системы предлагают информацию пользователям на основе доступных данных. Использование рекомендательных систем позволяет организациям увеличивать свои доходы, расширять сеть и способствовать открытию новой информации для пользователей. Подходы контентной и коллаборативной фильтрации являются наиболее популярными методами в рекомендательных системах, которые прогнозируют предпочтения пользователей. Однако эти методы не подходят для некоторых приложений, особенно когда коллаборативные данные отсутствуют или их очень мало. В этой работе разработана рекомендательная среда на основе обнаружения сообществ, которая в некоторых приложениях имеет преимущества перед другими подходами.

Ключевые слова: обнаружение сообществ, рекомендательная система, коллаборативная фильтрация, меры сходства.