Detect Overlapping Community With Reliability Measure And Rule Based For Improving Recommender Systems

Rzgar Sirwan Raza, Sarkhel H.Taher karim

Abstract: With the affluence of information produced by users in websites, people have problems to find the best information. Recommender systems are a big ingredients of online systems such as e-stores e-commerce providers. Recommendation methods use information available from users-items interactions and their contextual data to provide a best list of items for users. These methods are constructed based on similarity between users and/or items (e.g., a user is likely to purchase the same items as his/her most similar users). In this paper, we introduce a novel community detection recommendation algorithm that is based on rules. Community detection create groups of interacting vertices (i.e., nodes) in a network depending on their structural properties .We can extract rules in users-items interaction network .use this rules can help recommendation process. We apply the proposed algorithm on a movelense dataset. Our proposed method show better precision as compared to the state-of-the-art recommenders.

Index Terms: Recommender algorithms, social networks, network science, trust, overlapping community structure, reliability.

1 INTRODUCTION

In this days websites have huge data. This data produce by user when they search keywords, shop online, watch movies, participate in social networks even when they play online games. Data mining (especially web-based data) is one of the core problems in various disciplines including computer science, information technology, mathematics and social sciences [1]. Indeed our problem is an information overload. These issues can be solved by using of recommendation systems [2-5]. When users rate items or click on items, such information can be used to provide a list of recommended items for each user. Recommendation systems use history of rating along time and contextual information (if available). They have been successfully implemented in many systems such as e-commerce and movie recommendation website. Recommendatory systems are often divided into three categories: 1) Collaborative filtering, 2) content based filtering, 3) Hybrid approaches. When only rating histories are available, we can use recommender systems based on collaborative filtering; the recommendation algorithm finds appropriate items through collaborative interaction between users and items [3]. These algorithms have been successfully implemented in different real applications. The collaborative filtering algorithms have two classes: memory-based and model-based. Memory-based algorithms are more popular and use the similarity between users and items [3, 6, 7]. When users give the same rating (or purchase the same product) for the same item or similar items is most similar. Two similar items similarly be rated by the users. Memory-based collaborative filtering have various algorithms (such as the algorithm proposed in this work). Also model-based collaborative filtering algorithms fit a model for the present the predictions based on the learnt model [5, 8]. Each approach has its own advantages and drawbacks.

- *Rzgar Sirwan Raza is currently Lecturer in Human Development University, Iraq. E-mail: rzgar.sirwan@uhd,edu.iq*
- *Sarkhel H.Taher karim is currently Lecturer in University of halabja,Iraq ,. E-mail: Sarkhel.kareem@halabjauni.org*

Most traditional recommendation algorithms exploit the data as static. This systems do not consider users' preference changes or items popularity over time. In realistic scenarios, the over time (often referred to as drift in the literature), and the recommender should consider change in preferences [9-11]. Recently, an algorithm based on community identification approach has been proposed to handle the drift problem in recommender systems [10]. Variations of users' interest can be effectively discovered using overlapping community detection methods to provide alternatives for representing the early efforts on community detection that assumed that communities are non-overlapping or disjointed. There are a number of algorithms that discover overlapping community structure in networked structures. Applying the community detection techniques on recommender algorithms can help increasing the precision of the algorithms and better handle problem of recommender systems such as the cold-star and data sparsity problem. In this paper, we propose a novel recommendation method for improved the efficacy of recommender system. We use overlapping community detection and time to complete the recommendation lists for users. The general framework of the proposed recommender algorithm including three steps, (i) a method to detect overlapping communities with using the similarity matrix between users, (ii) time based overlapping communities to model drift of users' interest, and (iii) a new recommendation model based on users' dynamic interests and multi-memberships in their overlapping communities. We apply the proposed recommendation algorithm on a benchmark dataset and show its superior performance over state-of-the-art algorithms.

2 Proposed Recommender Algorithm

Recommender algorithms are often assessed in terms of various evaluation metrics such as precision, accuracy, ranked-based measures, diversity and novelty of recommendations [12-14]. In this work, we focus on the precision and accuracy of recommendations and introduce a novel recommender algorithm to improve the precision of recommendations. The proposed algorithm has two steps: overlapping community identification, and collaborative filtering through rule mining. A pseudo-code of the algorithm will be given after mathematical descriptions. The proposed recommendation algorithm is detailed in the following.

2.1 Initial rate prediction

In many real systems, the users-items interaction matrix is very sparse. Each user only have few ratings to items (i.e., some users (and/or items) might not have enough ratings). In this conditions many recommendation algorithms do not have good recommend. Cold-start problem is another challenge in real recommenders. This problem occur for users or items when some users or items might not have enough rating history [15]. In this paper to solve this problem the initial rates of the unseen items for the active user can be predicted using the following equation. These new rates can be in user- item matrix. In first step similarity between users should be calculated. A similarity value between users a and u, which is computed using the Pearson correlation coefficient function as follows:

$$sim_{a,u} = \frac{\sum_{i\in A_{a,u}} \left(r_{a,i} - \bar{r}_a\right)\left(r_{u,i} - \bar{r}_u\right)}{\sqrt{\sum_{i\in I_{a,u}} \left(r_{a,i} - \bar{r}_a\right)^2} \sqrt{\sum_{i\in I_{a,u}} \left(r_{u,i} - \bar{r}_u\right)^2}} \tag{1}$$

Where $r_{u,i}$ denotes the rating given by user *u* on item i, $\bar{r}_u$ is the average of the ratings given by u, and $I_{a,u}$ is the set of items which are rated by both a and u. Then probability of user *u* provides positive rating items *i* calculated by:

$$P_{a,i} = \bar{r}_a + \frac{\sum_{u\in N_{a,i}} sim_{a,u}\left(r_{u,i} - \bar{r}_u\right)}{\sum_{u\in N_{a,i}} sim_{a,u}} \tag{2}$$

where, $\bar{r}_a$ refers to the average of ratings for the target user a, $N_{a,i}$ is a set of neighbours of a who that rated the item i, and $sim_{a,u}$ is a similarity value between users a and u, which is computed in Eq(1).

2.2 Calculating reliability measure

A Reliability measure is used to evaluate the quality of the predicted rate. In [16] a reliability measure is proposed based on the positive and negative factors. This reliability measure uses the user-item rating matrix to calculate the quality of predicted ratings and to provide a feedback on the quality of these rates. This measure is calculated as follows:

$$R_{a,i} = \left(f_s\left(S_{a,i}\right) . f_v\left(V_{a,i}\right)^{f_s\left(S_{a,i}\right)} \right)^{\frac{1}{1+f_s\left(S_{a,i}\right)}} \tag{3}$$

where, $f_s(S_{a,i})$ and $f_V(V_{a,i})$ are the positive and negative factors of the reliability measure, respectively. The positive factor is calculated by:

$$f_s\left(S_{a,i}\right) = 1 - \frac{\bar{S}}{\bar{S} + S_{a,i}} \tag{4}$$

where, $S_{a,i}$ $\sum_{u\in N_{a,i}} sim_{a,u}$ denotes the value of the positive factor to calculate the reliability measure, and $\bar{S}$ is the median of $S_{a,i}$ values. The negative factor is calculated as:

$$f_V\left(V_{a,i}\right) = \left(\frac{max - min - V_{a,i}}{max - min} \right)^{\gamma} \tag{5}$$

where, $V_{a,i}$ and γ are calculated using the following relations:

$$V_{a,i} = \frac{\sum_{u\in N_{a,i}} sim_{a,u} . \left(r_{u,i} - \bar{r}_u - P_{a,i} + \bar{r}_a\right)^2}{\sum_{u\in N_{a,i}} sim_{a,u}} \tag{6}$$

$$\gamma = \frac{\ln 0.5}{\ln \frac{max - min - \bar{V}}{max - min}} \tag{7}$$

where $\bar{V}$ is the median of the values of $V_{a,i}$, max and min are the maximum and minimum score the user gave to the items . If the reliability value $R_{u,i}$ for the predicted score $P_{u,i}$ (i.e. Eq.(2)) for user u and item i is higher than a threshold value (θ), then the rating $P_{u,i}$ is added to user-item matrix associated with the last time that user u rated his/her latest item. In the experiments the threshold value (θ) is set to θ 0.7 as recommended in [16].

2.3 user network construction

The first step helped to solve the sparsity problem of the matrix. Also, using community detection helps solve the problem of cold-start problem [17]. For this purpose, in the proposed method, first with using the user–item rating matrix, the user–user matrix is constructed. In this graph, each user denotes a node and a link between two users indicates that they preferred same items, and the link weight represents the number of items. Here used Pearson correlation coefficient to construct the similarity matrix between users. Suppose that user i rates r for item jth. If there is no transaction information or the rating r is 0. Then, the transactions are represented as the user-item matrix G $|sim_{u,v}|$ using with Eq(1). The in this step is to obtain the similarity values between users, which is extracted from user–item interaction data. When the time of rating is available (which is the case in this work), one can take it into account in building the similarity relations. Let's define:

$$M = G \times G^T = \left[m_{ij} \right]_{I \times I} \tag{8}$$

2.4 Overlapping community detection

In this section a graph overlapping community detection method is proposed to group the users/items into several overlapped clusters. Integrating the community structure with the recommender algorithm helps to overcome the sparsity and cold-start problems. The proposed community detection method consists of three steps including; (1) Finding initial cluster centres, (2) Identifying overlapped communities and (3) Merging clusters. In the first step, the sparsest subgraph finding algorithm -which was recently proposed in [18] (a pseudo-code is shown in Algorithm 1)- is employed on the extracted user-user graph to find the centres of the initial clusters. The nodes of this subgraph should have the maximum dissimilarities with each other, to be used as the set of initial centres. While the new centres are being identified, the nodes are assigned to their corresponding communities iteratively while they are maximizing the following fitness function [19]:

$$f_c = \frac{k_c^{in}}{\left(k_c^{in} + k_c^{out}\right)^{\alpha}} \tag{9}$$

where k_c^{in} is the total internal degree of community c and k_c^{out} is the total external degree of this community, which is computed as the sum of all of the link weights between the internal nodes and external nodes of the cth community and α is value in the range of [1,2]. This process is performed iteratively over all clusters until there is not change and a steady-state solution is gained. At the end of the iterations the set of final centres is identified and the nodes are assigned to their corresponding communities characterized by its centres.

Algorithm 1. The proposed overlapping community detection method
Input: Time-weighted user network graph.
Output: List of overlapped communities.
Algorithm:
1: *Apply* the approximate sparsest subgraph finding algorithm (i.e. Algorithm 1 in []) to obtain S as initial center set;
2: $k' = |S|$;
3: *Set* $p_i = S_j, \forall j = 1, \ldots, k'$;
4: *Let* $p_i, \forall j = 1, \ldots, k'$ be initial center corresponding to j-th cluster C_j;
5: *for* each p_i do
8: *A= Identify neighbors* of C_j,
7: C_j^{new}= Iteratively assign the nodes of *A* to C_j if maximizes the Eq.(7).
8: *If* $C_j^{new} \neq C_j$ goto line 8;
9: *end for*;
10: *for* all $C_j, j = 1, \ldots, k'$ do
11: if $|C_j| < m$ then *Merge* C_j to other clusters;
12: *end for;*
13: *for each* C_i *and* C_j *,* $i \neq j$
14: if $r_{i,j} > \tau$ then *Merge* C_i and C_j;
15: *end for;*
16: *for* all $C_j, j = 1, \ldots, k'$ and u_i, *i=1..*|*U*| do
17: *Compute* μ_j^i using Eq.(9).
18: *end for*
19: *Return* C and μ;

After identifying initial communities, some of the clusters may not have sufficient members to be considered in the Rule mining step in order to detect meaningful rules. To overcome this problem, the clusters whose associated members are less than a threshold value will be merged with others. To this end, all nodes of small clusters will be assigned to their next nearest centre. Moreover, the identified communities may contain joint members belonging to both clusters, and thus these communities carry similar patterns. In this work the following equation is used to identify similar communities:

$$r_{i,j} = \frac{|C_i \cap C_j|}{\min\left(|C_i|,|C_j|\right)} \tag{10}$$

when $r_{i,j}$ is greater than a threshold value, there is no significant difference between the two communities, and thus they are merged to form a single community. Since users can belong to more than one community, in the next step the membership value of a user to each community should be computed. This value is defined as the ratio of all of the user's link weights in this community to the total link weights of the user in all of his/her corresponding communities. The membership value of user u_i with respect to community c_k is defined as follows:

$$\mu_k^i = \frac{\sum_{j=1}^{|c_k|} sim_{ij}}{\sum_{c_j \in \{c_1 \ldots c_p\}} \sum_{m=1}^{|c_j|} sim_{im}} \tag{11}$$

where c_j are all those communities that contain the ith user.

2.5 Rule mining

In this step, a time-weighting rule mining method, originally proposed in [10], is used to set of most prominent recommendation rules. The aim of this step is to identify the rules in the form of $A \to B$, which means that if a user has purchased (or rated) an item A previously, he/she will also be interested in an unseen item B. These rules can be identified by employing the identified overlapped communities. For each community, a set of rules can be detected based on the items which have been purchased by its corresponding users. For each identified rule of community *k*, a confidence value is obtained to determine its effectiveness as follows:

$$conf(A \to B) = \frac{1}{n_k(A)} \times \sum_{i=1} n_k(A,B) \tag{12}$$

where $n_k(A)$ is the number of transactions in which item A appears in that community, and $n_k(A,B)$ is the number of transactions that contain the item-set $\{A,B\}$ with the constraint that A appears in time s and B in time t of the community. Finally the global recommendation value of item j to user i is defined as:

$$R_{ij} = \max\{\mu_k^i \times \max[conf_k(x \to j)]\} \tag{13}$$

where $x \to j$ shows any rule whose consequent part is item j in community k. The procedure of the rule mining algorithm is represented in Algorithm 2.

Algorithm 2. The rule mining method

Input: A set of overlapped communities
Output: A set of rules
Algorithm:
1: ***For*** each $\{A, B\}$ in item set that forms a rule $A \to B$
2: ***Count*** the number of item A appearances in kth community.
3: ***Count*** the number of $\{A, B\}$ appearances in the dataset
5: ***Obtain*** the confidence of $A \to B$ using Eq. (11).
6: ***End*** for.
7: ***Return*** a set of $\{A \to B\}$ rules

2.6 Top-N recommendations

After ranking the recommendation values, a recommendation list for the target user can be generated using the Top-N recommendation strategy.

3 Experimental Results and Discussion

To assess the performance of the recommendation algorithms we apply them on Movielense-1G dataset. This data set has been frequently used as a benchmark in recommendation systems. This dataset contains approximately 1 million ratings from 6040 users on 3952 movies over three years. The performance of the proposed model (*Detect Overlapping Community with Reliability measure and Rule based* Recommendation System (DOCR3S)) is compared with a number of state-of-the-art

TABLE I

PRECISION OF ALGORITHMS

Algorithm	Top5	Top10	Top15	Top20
DOCR3S	**0..07**	**0.929**	0.895	**0.907**
TWAR	0.904	0.893	0.900	0.890
IFCCF	0.906	0.916	**0.911**	0.903
TRACCF	0.871	0.884	0.874	0.871
KMCF(user-based)	0.867	0.880	0.872	0.869

Our experiments adopt the Top-N recommendation strategy and use precision as the evaluation metric. The recommendations predicted by the algorithm that is also liked by the users are defined as True Positive (TP), and those not like by the users are defined as False Positive (FP). The items that are not recommended and not liked by the users are defined as True Negative (TN), and those liked by users but not recommended by the algorithm are defined as False Negative (FN). Having these values, once can calculated the precision of the algorithm by:

$$percision = \frac{1}{n}\sum_{i=1}^{n}\frac{TP}{TP+FP} \qquad (14)$$

Table I compares the precision of the algorithms for different values of N in the top-N recommendation task. The proposed recommendation algorithm is the top-performed among the algorithms. In top-15 algorithm (IFCCF) has better performance. The proposed algorithm will be implemented in more benchmark datasets to confirm its efficiency.

4 CONCLUSIONS

Recommendation systems have many potential applications in both academia and industry. They use available information on the users-items interactions (e.g., purchase or rating history of users on items) and their contextual data to find proper recommendation list for each user. Data sparsity is one of the major challenges of recommender algorithms, meaning that often there are only a few ratings form each user, resulting in a highly sparse bipartite users-items network. In this manuscript we proposed a new recommender algorithm based on overlapping community structure of users. The proposed algorithm appropriately uses association rule to design efficient recommendations. We applied the proposed algorithm on Movilense dataset, which has been frequently used as a benchmark dataset in recommender systems recommender algorithms including Time-weighted Association Rules (TWAR) [10], user-based k-means collaborative filtering (KMCF) [20], item-based fuzzy clustering collaborative filtering (IFCCF) [20] and Trust-aware clustering collaborative filtering (TRACCF) [20]. The original ratings are on a scale of 1–5, with 5 being the highest rating and 1 the lowest rating. In this work, we are only interested in predicting Like and Dislike for the users, and thus binarize the ratings by considering the ratings above 2 as Like, and Dislike otherwise. The Like state is shown by 1, while Dislike is represented by 0 in the binary rating matrices. Such a binarization strategy has been previously applied to design recommender algorithms [6, 7, 9]**.** research. The results revealed better recommendation precision of the proposed algorithm than a number of state-of-the-art methods.

REFERENCES

[1] X. Wu, X. Zhu, G. Q. Wu, and W. Ding, "Data mining with big data," IEEE Transactions on Knowledge and Data Engineering, vol. 26, pp. 97-107, 2014.

[2] P. Resnick and H. R. Varian, " Recommender systems," Communications of ACM, vol. 40, pp. 56-58, 1997.

[3] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," presented at the 10th international conference on World Wide Web, 2001.

[4] R. Burke, "Hybrid recommender systems: survey and experiments," User Modeling and User-Adaptive Interaction, vol. 12, pp. 331-370, 2002.

[5] M. Ranjbar, P. Moradi, M. Azami, and M. Jalili, "An imputation-based matrix factorization method for improving accuracy of collaborative filtering systems," Engineering Applications of Artificial Intelligence, vol. 46, pp. 58-66, 2015.

[6] A. Javari and M. Jalili, "Accurate and novel recommendations: an algorithm based on popularity forecasting," ACM Transactions on Intelligent Systems and Technology, vol. 5, p. 56, 2014.

[7] A. Javari and M. Jalili, "A probabilistic model to resolve diversity-accuracy challenge of recommendation systems," Knowledge and Information Systems, vol. 44, pp. 609-627, 2015.

[8] X. Luo, M. Zhou, Y. Xia, and Q. Zhu, "An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems," IEEE Transactions on Industrial Informatics, vol. 10, pp. 1273-1284, 2014.

[9] S. H. Daneshmand, A. Javari, S. E. Abtahi, and M. Jalili, "A time-aware recommender system based on dependency network of items," The Computer Journal, vol. 58, pp. 1255-1266, 2015.

[10] H. Feng, J. Tian, H. J. Wang, and M. Li, "Personalized recommendations based on time-weighted overlapping community detection," Information and Management, vol. 25, pp. 789-800, 2015.

[11] P. G. Campos, F. Díez, and I. Cantador, "Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols," User Modeling and User-Adaptive Interaction, vol. 24, pp. 67-119, 2014.

[12] M. Izadi, A. Javari, and M. Jalili, "Unifying inconsistent evaluation metrics in recommender systems," presented at the ACM International Workshop on Recommender Systems Evaluation: Dimensions and Design, 2014.

[13] P. G. C. Soto, "Temporal models in recommender systems: an exploratory study on different evaluation dimensions," PhD, Universidad Autónoma de Madrid Escuela Politécnica Superior, 2011.

[14] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," Knowledge-Based System, vol. 46, pp. 109-132, 2013.

[15] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," presented at the ACM SIGIR Conference on Research and Development in Information Retrieval, 2002.

[16] P. Moradi and S. Ahmadian, "A reliability-based recommendation method to improve trust-aware recommender systems," Expert System with Applications, vol. 42, pp. 7386-7398, 2015.

[17] A. Javari and M. Jalili, "Cluster-based collaborative filtering for sign prediction in social networks with positive and negative links," ACM Transactions on Intelligent Systems and Technology, vol. 5, p. 24, 2014.

[18] P. Moradi, S. Ahmdaian, and F. Akhlaghian Tab, "An effective trust-based recommendation method using a novel graph clustering algorithm," Physica A, 2015.

[19] A. Lancichinetti, S. Fortunato, and J. Kertesz, "Detecting the overlapping and hierarchical community structure in complex networks," New Journal of Physics, vol. 11, p. 033015, 2009.

[20] C. Birtolo and D. Ronca, "Advances in clustering collaborative filtering by means of fuzzy C-means and trust," Expert System with Applications, vol. 40, pp. 6997-7009, 2013.