Neurocomputing 210 (2016) 217–226

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

# Presenting new collaborative link prediction methods for activity recommendation in Facebook

Amin Shahmohammadi [a,*], Ehsan Khadangi [b], Alireza Bagheri [b]

[a] Department of Computer Engineering, Pooyesh Institute of Higher Education, Qom, Iran
[b] Department of Computer Engineering and information technology, Amirkabir University of Technology, Tehran, Iran

ARTICLE INFO

*Article history:*
Received 5 May 2015
Received in revised form
8 June 2016
Accepted 9 June 2016
Available online 15 June 2016

*Keywords:*
Link prediction
Recommender system
Facebook
Collaborative filtering
Online social network
Activity network

ABSTRACT

One of the common methods used in recommender systems is collaborative filtering methods. In these methods, same-interest users' preferences are often recommended to each other based on examining their past interests. On the other hand, one of the recommendation methods in social networks is to measure the proximity of the two nodes in the graph. Although many researchers have dealt with friendship link prediction in different online social networks, very little notice has been spent on activity prediction based on different users' interactions. The main objective of this paper is the use of collaborative filtering methods for activity prediction and recommendation both for pairs of users without any interaction background and also for user pairs with the activity background. In this regard, a new concept is initially presented named as "collaborative path". Then based on the collaborative path, four directed proximity measures are proposed. In addition, three new algorithms, including two algorithms based on collaborative random walks, one for mixed network and one for multilayer network and the Collaborative-Association-Rule algorithm are presented. Finally, in order to evaluate our proposed methods, we perform some experiments on the dataset of different Facebook activity networks including like, comment, post, and share networks. The results show that the proposed collaborative methods deal with the activity prediction well without suffering from the cold start problem, and outperform the existing state of the art methods.



## 1. Introduction

Online social networks today have become powerful tools for people's communication with each other, sharing information and ideas as well as spreading individual thoughts and emotions. The friendship network which shows people's friendships in these networks can be studied for different purposes. Although analyzing the friendship network forms the approach of most researches on social networks, the friendship network cannot represent users' behaviors in the real world suitably. According to this, Chun et al. proposed the concept of activity network. In this network, the nodes show the users and the links model the activity between them. In Facebook as the most common social networking website, users can perform various activities on others' profile. For instance, they can like the content interesting to them and share it with friends. They can comment on others' content, or send posts to others. A user's activity on others' contents shows that user's interest in those contents.

The largeness of social networking services has caused users to spend much time to find their interesting contents. Therefore, many researchers have dealt with recommending news, friends, followee, groups and etc. These methods often use keywords interesting to users [1,2], users' profile information [3], the friendship network structure [3] and also follower–followee relationships [2]. There are also different algorithms used.

The algorithms used for recommendation of products, films, music and so on are often divided into two: collaborative filtering and content-based filtering. In content-based methods, things are recommended, similar to which the user has already liked. However, in collaborative filtering methods, the inside of the contents are not examined. Instead, based on different users' ratings for different products, their similar preferences are distinguished. In order to recommend some products to a user, we consider the ratings that similar users have made for them. The product that people similar to the user have liked more is recommended to the user. Content-based methods are not capable of assessing the quality of the product and collaborative filtering methods faces

* Corresponding author.
*E-mail addresses:* shahmohammadi.amin@gmail.com (A. Shahmohammadi), khadangi@gmail.com (E. Khadangi).
*URL:* http://www.khadangi.ir (E. Khadangi).

http://dx.doi.org/10.1016/j.neucom.2016.06.024
0925-2312/

cold start problem.

This paper aims to propose some new methods for prediction of activity links and accordingly recommending users that are likely to be interested by the user for interaction. The nodes of activity networks are the users and the directed weighted links therein are users' activities. These activities include like, comment, share, and post. In this paper, in order to solve the problem of activity recommendation, a collaborative path in the activity network is defined. According to the collaborative path, three link prediction algorithms are presented. All of the presented methods are based on the activity network structure and the activity recommendation is carried out using the link prediction in the activity network.

The rest of the paper is organized as follows. In Section 2, we review related works. Section 3 covers the problem definition. The presented methods are elaborated in Section 4. Experimental setup including dataset and evaluation measures described in Section 5. Section 6 shows the experimental results. Finally, Section 7 concludes the paper.

## 2. Related works

Many works in recent years have dealt with the analysis of the activity network. Many of these works have studied the differences between the activity and friendship networks and have reported them in terms of structural properties [4–6]. In addition, few works have studied the evolution of the activity network over time [7]. Other works have also dealt with the analysis of users' interactions for use in a special applications such as finding influential nodes [8–10] and measuring tie strength [11,12]. Most of researches approve that based on the nature of the activity network which is a better display of users' real interactions, the analysis would present us with more real results.

The problem of link prediction, on the other hand, has been studied over different networks. This problem was initially presented in [13]. They used similarity measures for predicting links in physics co-authorship network. Most of link prediction methods use nodes' proximity measures. The closer the two nodes are, the more likely they are to form links with each other in the future. For this reason, different measures have been presented and used. The common examples include number of common neighbors, Jaccard coefficient, Katz [14], and Adamic-Adar score [15]. In [13], different proximity measures for link prediction in scientific co-authorship networks were evaluated and the Adamic-Adar was the most efficient algorithm. The algorithm presented in [16] scores the nodes according to the probability of a random walker to reach from one node to another. In [17,18], random walk was used for link prediction. In [19], a method based on rooted page rank was presented. They also used a supervised method for link prediction. In [3], a biased random walk was used for the link prediction. They used the attributes of nodes and links for random walk training so as to visit nodes likely to form links to the initial node. Among the works which used the supervised method for link prediction is [3,19,20]. In these methods, proximity measures, users' profile information, and the information of users' relationships are often used as features. Then several node couples are given to the algorithm as training data. If there are links between these node couples, the label would be considered 1 and otherwise, it would be 0.

Lerman et al. [21] used proximity measures for activity recommendation in social networks. They presented and used directed version of different baseline measures. Scholz et al. also considered link prediction as prediction of new and recurring links. They performed their experiments on network of people's face-to-face contacts in conferences and compared node-based proximity measures and path-based ones [22]. Some works used collaborative and content-based filtering to link prediction and recommendation in different networks. Hannon et al. presented a twit recommendation based on collaborative and content based filtering [2]. As we mentioned before, content-based methods are not often capable of assessing the quality of texts. For instance, assessing the keywords of a text does not define the amount of its good- or badness whereas collaborative methods are capable of assessing the quality of the recommended items. For instance, if a user's texts have been liked fewer, it shows its low favor. The problem with collaborative methods is what to be called cold start. It happens because users may initially not have rated products in e-commerce websites [23].

This paper attempts to recommending activities by analyzing the activity network and using collaborative filtering methods. As proposed in other works, Facebook activity network develops faster than other networks so that it could be claimed that the structure of this graph is so rich that the problem of cold start will not hinder the presented methods. In addition, people do activities both on new people and on people with past activities. Therefore, in link prediction in the activity network, we predict new links occurrence together with rise of the weight of old links.

## 3. Problem definition

In this paper, we model Facebook activity network as a weighted and directed graph $G = (V, E, f, c)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of users, $E = \{e_1, e_2, \ldots, e_m\}$ includes the links, $f: E \rightarrow \{(x, y) | x, y \in V\}$ is the direction function, and $c: E \rightarrow N$ is the link weight and shows the number of activities. The link $f(e_i) = (x, y)$ shows the user X's activity on the user Y's profile and its weight $c(e_i)$ equal to the sum of all activities corresponding to the edge $e_i$. By this definition we can model both single activity and mixed activity networks. For single activity modeling, the edges correspond to only one kind of activity such as like, comment, share or post whereas in mixed activity networks any activity can make the edges. In this case the link weight $c(e_i)$ is equal to the sum of all activities including likes, shares, comments and posts corresponding to the edge $e_i$.

When we model the activity network as multilayer, each layer is a single activity network for one of the activity types (like, share, comment, post). The graph can be represented as $G = (V, E, f, c, l)$ where $l: E \rightarrow \{L_1, L_2, \ldots, L_k\}$ is the set of layers. Let us assume that the like activity is $L_1$, and the user $x \in V$ has liked the user $y \in V$ for $w$ times, then there will be an edge $e_1 \in E$ for which $f(e_i) = (x, y)$ and $c(e_i) = w$ and $l(e_i) = L_1$.

The goal of activity prediction problem is the recommendation of texts which if the user $s$ sees them, $s$ will do some activity over them. These texts belong to both users over whom s has already had activities and those without any activities from $s$. In this paper, the problem of activity prediction/recommendation is defined as below:

Suppose that $G^t = (V, E^t, f^t, c^t)$ is the activity graph from the time 0 till $t$. For the node $s \in V$ it is preferred to predict the node $d \in V$ so that:

$$\exists\ e \in E^{t+1} | \left( e \neq E^t \wedge f^{t+1}(e) = (s, d) \right) \vee \left( e \in E^t \wedge c^{t+1}(e) > c^t(e) \right) \quad (1)$$

The equation above shows the node $s$ has done some activity on the node $d$ in the time $t + 1$. Therefore, we can recommend $d$ to the node $s$ before time $t + 1$. Watching the profile of the node $d$, if the node $s$ likes it, $s$ will do its favorite activities over the profile of $d$.

Recommender systems in a way might be categorized into explicit and implicit. An ideal explicit recommender system for a shop will immediately tell the customer, as he/she enters the shop,

to buy some special product. Meanwhile, implicit recommender systems present their recommendation softly. The algorithms in this paper may also be used for both explicit and implicit recommendations. However, most of them recommend doing any type of activity rather than recommending a specific type of activity. In other words, they only recommend a user or their texts to another, and the user, who has received this service, reading the other's texts, can do a favorite activity. The presented algorithms, however, can also recommend a particular activity.

For link prediction, the algorithms take a source node such as $s \in V$. Then it measures how close the other nodes are to $s$. Each node in $V(G)$ will be scored based on its closeness to the source node. The higher score of a node shows that $s$ is more willing to do activities with that node. After computing scores for each node, the $top_k$ nodes will be recommended to the user $s$.

## 4. Proposed methods

In this paper, we propose and define a new specific path called collaborative path (CP) for directed graphs. Based on this concept, different proximity measures and algorithms are presented. In this section, we present our suggested methods for predicting or recommending the link in the activity network. In this section, the collaborative path and proximity measures are initially explained. Then we elaborate two link prediction algorithms.

### 4.1. Collaborative path and collaborative proximity measures

One class of methods for link prediction is based on proximity measures in the graph. In these methods, the closer the two nodes are to each other, the more likely they are to link each other in the future. Proximity measures are divided into two groups of path-based and node-based. This paper initially defines a collaborative path comprising three links. The following proximity measures and algorithms are presented based on this collaborative path. The collaborative path is defined as: suppose that in the activity graph $s, t, u, v \in V$. We say there is a collaborative path from node $s$ to $v$ including nodes $s$, $t$, $u$, and $v$ so that:

$$t \in \Gamma_{out}S, \quad u \in \Gamma_{in}t, \quad v \in \Gamma_{out}u \tag{2}$$

Where $\Gamma_{out}S$ means output neighbors of node $s$. And $\Gamma_{in}t$ means the input neighbors of node $t$. Fig. 1 shows a collaborative path. If we are talking about like activity network, node $s$ has liked node $t$. Node $u$ has also liked node $t$. If node $u$ likes node $v$, it is likely that $s$ will like $v$ in the near future. Since $u$ and $s$ have the same idea in terms of liking $t$, it is likely what $u$ likes, node $s$ will also like. This link succession makes a three-link path which may be used in different ways.

In this path, $u$ and $s$ are two nodes that are likely to have similar interests. $t$ and $v$ are also two nodes that may hold similar contents. In addition, there may be a high number of collaborative paths including links and nodes from node $s$ to another such as $v$. In each collaborative path we name the second node as $t$ and the third node as $u$. The more the number of $u$ nodes in collaborative paths from $s$ to $v$ is, it means there are many users who hold the same interest of $s$ and have liked $v$, so it is likely that $s$ like node $v$. The more the number of $t$ nodes in collaborative paths from $s$ to $v$ is, it means the contents of user $v$ are more similar to contents that $s$ has liked. Based on these explanations, several proximity measures may be presented based on collaborative paths: the number of nodes such as $u$ (Eq. (3)), $t$ (Eq. (4)), sum of $u$ and $t$ nodes (Eq. (5)) and the total number of collaborative paths (Eq. (6)). Equations of these proximity measures are presented as follows:


**Fig. 1.** Illustration of a collaborative path in a directed network.

$$tNum(s, v) = |\Gamma_{out}S \cap \Gamma_{out}(\Gamma_{in}v)| \tag{3}$$

$$uNum(s, v) = |\Gamma_{in}v \cap \Gamma_{in}(\Gamma_{out}s)| \tag{4}$$

$$utNum(s, v) = tNum + uNum \tag{5}$$

$$CNum(s, v) = |\{e \in E | e = (a, b), a \in \Gamma_{out}s, b \in \Gamma_{in}v\}| \tag{6}$$

It should be mentioned, the node $v$ may be out-neighbor of node $s$. Therefore the proposed methods, not only score nodes without any link from $s$ to it, but also they can predict recurring of the current links. Besides the proximity measures stated, we preset two other algorithms described as follows.

### 4.2. Collaborative Random Walk (C-RW)

One of the methods for predicting links is the use of random walk for finding the closeness between nodes. Random walk starts from a source node. At each stage, it selects a neighbor node with probability $1/k_v$, visits that node and continues this job. A random walk based proximity measure scores each node it visits at each stage. The nodes with the highest scores mean they are more accessible from the first node. So it is likely that the high-score nodes link to the first node in the future.

The Collaborative Random Walk (C-RW) method starts from an initial node $s$ and at each step passes through a collaborative path. In other words, at each iteration of Algorithm 4.1, one of the output neighbors of node $s$ is selected as $t$. Then one of in-neighbors (respectively out-neighbors) of node $t$ (respectively $u$) is selected as $u$ (respectively $v$). According to these three steps, random walker has passed through a collaborative path and reached to an arbitrary node $v$. In this time, it adds one score to node $v$ and goes on. At the end of the algorithm, higher scores an arbitrary node has gained, it is more likely that $s$ performs some activity on his/her profile. Selecting a random output/input neighbor of a node, can be both uniformly and by considering the weights of output/input links. So we have two weighted and unweighted collaborative random walk algorithms.

**Algorithm 4.1.** Collaborative-Random Walk algorithm (C-RW).

1: Input: $s$, graph (ID of its nodes are from 1 to N), maxIteration
2: Output: $top_k$ ($k$ nodes for recommendation)
3: Initialize: $visits[1: N] \leftarrow 0$
4: **while** $i < maxIter$ **do**
5: $sNeighbors \leftarrow$ ID of out-neighbors of node $s$ of graph
6: $t \leftarrow$ Select 1 element uniformly at random from $sNeighbors$
7: $tNeighbors \leftarrow$ ID of in-neighbors of node $t$ of graph
8: $u \leftarrow$ Select 1 element uniformly at random from $tNeighbors$
9: $uNeighbors \leftarrow$ ID of out-neighbors of node $u$ of graph
10: $v \leftarrow$ Select 1 element uniformly at random from $uNeighbors$
11: $visits[v] \leftarrow visits[v] + 1$
12: $i \leftarrow i + 1$
13: **end while**
14: $top_k \leftarrow$ Index of top $k$ values of visits (which is ID of $k$ most selected nodes as $v$ node)

The input graph of this algorithm is mixed activity network. The weight of mixed network links is sum of the weight of corresponding links in like, comment, post, and share networks. When this algorithm takes mixed network as input, then only recommends doing an activity rather than recommending the activity type, and it could be used for implicit recommender systems. Also we can use this algorithm on a specific activity network, such as like, comment, post, and share. In this case it could be used for explicit recommender systems. As stated earlier, a collaborative path in the form "$s \to t \leftarrow u \to v$" denotes $s$ and $u$, both are interested in $t$'s profile. In the case of user $t$, $s$ has done an arbitrary activity ActS; it may be like, comment, post, and share. The user $u$ has done an arbitrary activity on $t$ as ActU. It is clear that, if $u$ does the same activity ActU on an arbitrary node $v$, then $s$ may be interested to do ActS on $v$. Based on this, the multilayer collaborative random walk algorithm has been designed. This algorithm recommends a special activity for the user $s$. Let ActS be the activity to be recommended to the user $s$. In each iteration, the algorithm chooses $t$ from $s$'s out-neighbors in the ActS layer of network. It randomly selects an activity layer from which the other two edges of collaborative path will be selected. Finally the node $v$ will be scored by one. Algorithm 4.2 shows the pseudo-code of multilayer collaborative random walk algorithm. Besides node $s$ and the max iteration, the input of this algorithm is a four-layer network with the layers like, comment, post and share. Activity also is a special activity to be recommended.

**Algorithm 4.2.** Multilayer collaborative random walk algorithm (ML-C-RW).

1: Input: $s$, maxIteration, Multilayer Activity Graph (ID of vertices are from 1 to $N$), Activity
2: Output: $top_k$
3: Initialize: $visits[1{:}N] \leftarrow 0$
4: ActS ←Activity
5: **while** $i$<maxIteration **do**
6: $sNeighbors$ ←ID of out-neighbors of node $s$ in ActS layer of Multilayer Activity Graph
7: $t$ ←Select 1 element uniformly at random from $sNeighbors$
8: ActU ←Randomly select 1 of layers with probability 1/number of layers
9: $tNeighbors$ ←ID of in-neighbors of node t in ActU layer of Multilayer Activity Graph
10: $u$ ←select 1 element uniformly at random from $tNeighbors$
11: $uNeighbors$ ←ID of out-neighbors of node $u$ in ActU layer of Multilayer Activity Graph
12: $v$ ←select 1 element uniformly at random from $uNeighbors$
13: $visits[v] \leftarrow visits[v] + 1$
14: $i < i + 1$
15: **end while**
16: $top_k$← Index of top $k$ values of visits (which is ID of $k$ most selected nodes as $v$ node)

In this algorithm, we select different layers i.e. activities with equal probability. Based on the dataset under study, this probability is 0.25. Then in the selected layer, we move on a collaborative path and increase the score of the visited node in that layer. At the end, we will have the visiting number of different nodes in different layers and we can present an activity-based recommendation based on this number. It is worth to mention that both presented algorithms are applicable in both weighted and unweighted modes.

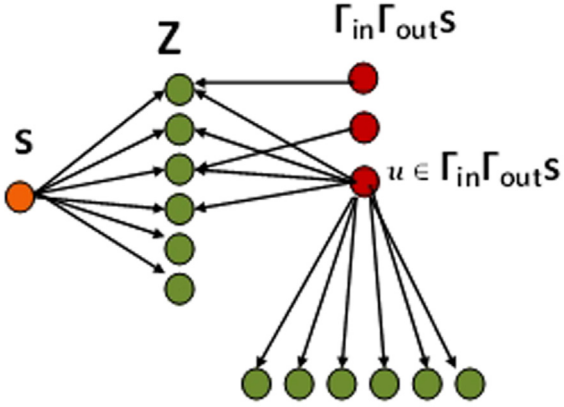


**Fig. 2.** Definition of link prediction as an associated rule problem.

### 4.3. Collaborative-association rule method

Mining association rules is one of the popular data mining techniques [24]. Based on users' past purchases, this method extracts rules based on which some recommendations are ready to present to users [25]. Recommender systems based on mining associate rules are often applicable for recommendations in markets according to market basket analysis. The problems which association rules deal with have two inputs: set of items and transactional database. The set of items is as $I = \{i_1, i_2, \ldots, i_n\}$ and the set of transactions is as $D = \{t_1, t_2, \ldots, t_m\}$, where $t_i \in D \subset I$. One rule is represented as $X \to Y$, where $x \cap Y = \varnothing$ and $X, Y \subseteq I$. This rule means if there is a transaction $t$ including the item set $X$, it is likely that $Y$ will also be there. The problem of predicting activity for a node such as $s$ can be defined as an association rule mining problem. In such a case, $I = \{x | x \in \Gamma_{out}\Gamma_{in}\Gamma_{out}s\} \subseteq V$ and $D = \{t | \forall\, u \in \Gamma_{in}\Gamma_{out}s,\ t = \Gamma_{out}u\}$ where $t \in D \subset I$. If $Z = \Gamma_{out}s$ (Fig. 2), we only need to calculate the value of the rule $Z \to x$ for each node $x$ existent in $I$. Finally, $k$ most valuable nodes are recommended to $s$. It is significant to note that the condition $Z \cap x = \varnothing$ is not included herein so that user $s$ can again preform activities on nodes which have already received activities from $s$.

For evaluating association rules, different measures can be used. The most applicable method is the confidence. Methods such as All-Confidence and lift can also be used. In this paper, new measures are presented for gaining the value of association rules, which are applicable for the problem above. Proposed rule evaluation methods measure how much times node $x$ have been present with nodes belonging to $Z$ at $t$ sets. In this regard, two evaluation measures are presented as below.

$$f_1(Z \to a) = \sum_{t \in D} \text{isin}(a, t) \times |t \cap Z| \tag{7}$$

Where

$$\text{isin}(a, t) = \begin{cases} 1; & \text{if } a \in t \\ 0; & \text{otherwise} \end{cases}$$

To calculate $f_1$, for each $t \in D$ the algorithm initially gains the number of nodes of $Z$ present at $t$. Then it adds this value to the score of all nodes at $t$. Therefore, each node gets one score corresponding to each presence together with another node belonging to $Z$ at a $t$. The more a node is located together with nodes of $Z$ at $t$ sets, the higher scores it gets. Eq. (8) shows the evaluation measure $f_2$.

$$f_2(Z \rightarrow a) = \sum_{t \in D} \text{isin}(a, t) \times \frac{|t \cap Z|}{|t|} \quad (8)$$

This measure is the normalized version of the first function. For each, this function initially measures the proportion of the number of nodes belonging to $Z$ present at $t$ to the total nodes at $t$. Then, it adds this value to the value of all nodes at $t$. If the first function is used, the gained weight for each node equals the number of collaborative paths from $s$ to that node. Algorithm 4.3 represents the collaborative-associative rule algorithm.

**Algorithm 4.3.** Collaborative associated rule algorithm (CAR).

1: Input: graph ((ID of its nodes are from 1 to $N$)), $S$ (source node), $f$ ($f$ is "$f_1$" or "$f_2$")
2: Output: $top_k$
3: Initialize: $values[1{:}\,N] \leftarrow 0$
4: $Z \leftarrow$ IDs of Out-Neighbors of node $S$ of graph
5: $S' \leftarrow EmptyVerctor()$
6: **for** each $z$ in $Z$**do**
7: $S' \leftarrow S' \cup$ In-Neighbors $(z)$
8: **end for**
9: **for** each $u$ in $S'$ **do**
10: **if** $f == f_1$ **then**
11: $D = |\text{Out} - \text{Neighbors}(u) \cap Z|$
12: **end if**
13: **if** $f == f_2$ **then**
14: $D = |\text{Out} - \text{Neighbors}(u) \cap Z| / |\text{Out-Neighbors } (u)|$
15: **end if**
16: **for** each $a$ in Out-Neighbors $(u)$ **do**
17: $values[a] = values[a] + D$
18: **end for**
19: **end for**
20: $top_k \leftarrow$ Index of top $k$ elements of values

## 5. Experimental setup

### 5.1. Dataset

The dataset used in this paper is extracted over a period of three years from the activities of some Facebook users' from January 2011 to January 2014 including likes, comments, posts, and shares. During these three years, for each month, some information about each user's activities on other users' profiles were collected. This information consists of the type of activities and the weight of activity for each type, which is the number of times that activity has been done. Because users' activities for each month are available, we can construct the activity network for different time periods. For an arbitrary time period $t$, the activity network can be created such that when node 1 does an activity (such as like) with node 2 during the time period $t$, a link is drawn from node 1 to node 2. For each activity type in the dataset, an activity network can be made for any time period. If we ignore the type of activities, the resulted network will be a mixed activity network.

The dataset of this paper consists of 20 different activity networks. They differ in activity type and time period. In terms of activity type, there are like, share, comment, post and mixed networks. In terms of time period, there are 4 time periods differing in length. They are 9-month, 15-month, 21-month and 27-month periods. The first time period is from the 19th month to the 27th month. The second time period is from the 13th month to the 27th month. The third one is from the 7th month to the 27th month. The last one is from the first month to the 27th month. Tables 1 and 2 show different statistics of dataset networks. In these tables the name of each network consists of its type and its length.

**Table 1**
different statistics of 20 activity networks in the dataset.

| Network | ♯of nodes | ♯of edges | Ave. degree | Diameter |
|---|---|---|---|---|
| Like9 | 4979 | 17,798 | 3.575 | 28 |
| Like15 | 5262 | 20,582 | 3.911 | 28 |
| Like21 | 5451 | 22,384 | 4.106 | 28 |
| Like27 | 5640 | 24,409 | 4.328 | 28 |
| comment9 | 2958 | 4573 | 1.546 | 32 |
| comment15 | 3315 | 5589 | 1.686 | 25 |
| comment21 | 3617 | 6532 | 1.806 | 28 |
| comment27 | 4002 | 8378 | 2.093 | 26 |
| share9 | 939 | 927 | 0.987 | 8 |
| share15 | 1070 | 1086 | 1.015 | 8 |
| share21 | 1129 | 1147 | 1.016 | 8 |
| share27 | 1141 | 1159 | 1.016 | 8 |
| Post9 | 1229 | 1006 | 0.819 | 6 |
| Post15 | 1677 | 1523 | 0.908 | 7 |
| Post21 | 2026 | 1994 | 0.984 | 9 |
| Post27 | 2539 | 3014 | 1.187 | 15 |
| Mixed9 | 5278 | 19,746 | 3.741 | 23 |
| Mixed15 | 5584 | 23,072 | 4.132 | 23 |
| Mixed21 | 5789 | 25,362 | 4.381 | 23 |
| Mixed27 | 6033 | 28,562 | 4.734 | 23 |

**Table 2**
different statistics of 20 activity networks in the dataset.

| Network | Size of Giant SCC | Ave. Clustering coefficient | ♯of SCCs |
|---|---|---|---|
| Like9 | 0.360 | 0.215 | 3037 |
| Like15 | 0.391 | 0.227 | 3045 |
| Like21 | 0.406 | 0.233 | 3076 |
| Like27 | 0.421 | 0.237 | 3111 |
| comment9 | 0.086 | 0.133 | 2460 |
| comment15 | 0.116 | 0.144 | 2670 |
| comment21 | 0.150 | 0.151 | 2802 |
| comment27 | 0.201 | 0.165 | 2968 |
| share9 | 0.007 | 0.069 | 897 |
| share15 | 0.007 | 0.068 | 1020 |
| share21 | 0.006 | 0.069 | 1075 |
| share27 | 0.006 | 0.068 | 1087 |
| Post9 | 0.003 | 0.020 | 1211 |
| Post15 | 0.004 | 0.028 | 1646 |
| Post21 | 0.003 | 0.035 | 1987 |
| Post27 | 0.007 | 0.050 | 2453 |
| Mixed9 | 0.384 | 0.225 | 3102 |
| Mixed15 | 0.418 | 0.235 | 3088 |
| Mixed21 | 0.444 | 0.242 | 3058 |
| Mixed27 | 0.473 | 0.253 | 3044 |

In these tables, the number of nodes column specifies the number of users involved in that activity during that time period. The table shows that increasing the time period increases the number of nodes and edges. The first column also denotes that in our dataset most of the users were interested to do the like activity, and the post and share activity networks have the minimum number of users. In directed networks, the average in-degree and the average out-degree are identical. The average degree of activity networks shows that each person normally involves in like activity more than other activities. However the users of post activity network have the least activity on average. Average clustering coefficient is undirected and unweighted. By increasing the number of nodes and edges in a graph, the size of giant connected component will be increased. So the giant components of like activity network are largest in comparison to other networks.

In each experiment, we deal with two time periods. The first time period is for analysis and the second time period is the target time period and used for prediction. The target period in all experiments is the 28th month activities. We have selected 100

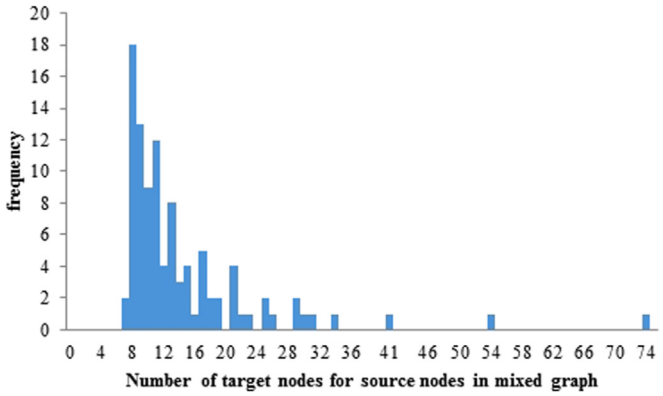


**Fig. 3.** Histogram of the number of target nodes for source nodes for implicit recommendation.

source nodes which are active in both time periods. In all experiments these 100 nodes have been used to evaluate different algorithms. For each source node some target nodes are selected. The source nodes have performed activities on target nodes in the second period.

Our experiments consist of explicit and implicit activity recommendations. For implicit recommendation, mixed activity network has been analyzed. So the target nodes have been selected from the second time period of the mixed activity network, for each source node. For explicit recommendation, like activity network has been analyzed. So the target nodes have been selected from the second time period of like activity network. Clearly different users do not have the same number of activities in the second time period (the 28th month). So the 100 source nodes do not have the same number of target nodes. Fig. 3 shows the histogram of the number of targets for source nodes used in implicit recommendation experiments. Fig. 4 shows the same histogram for explicit recommendation experiments.

### 5.2. Evaluation measures

The methods of this paper detect several candidate nodes for one source node and score them. The other nodes of the graph which do not deserve to be candidates according to the algorithm do not get scores. Both the scored and non-scored nodes may include real target nodes for the source node. The more target nodes an algorithm scores, and the higher it scores the target nodes versus other nodes, the algorithm is better. Therefore, to assess proposed algorithms and compare with different methods, the measures Precision, Recall, and $F1$-Score can be used. Precision herein means the proportion of real scored target nodes to the total scored nodes. Recall means the proportion of real scored target nodes to the total real target nodes. This paper uses the $F1$-Score measure in order to combine the other two measures. This measure equals the harmonic mean of Precision and Recall. Since our algorithms take a source node as an input and give $k$ number of nodes with high scores, the measures are calculated for $k$ high-ranked nodes. The precision at $k$-best nodes ($P@k$) means the ratio of the number of correctly predicted nodes in the $k$-best nodes to $k$. Eq. (9) shows the formula of $P@k$ for a given source nodes $s$.

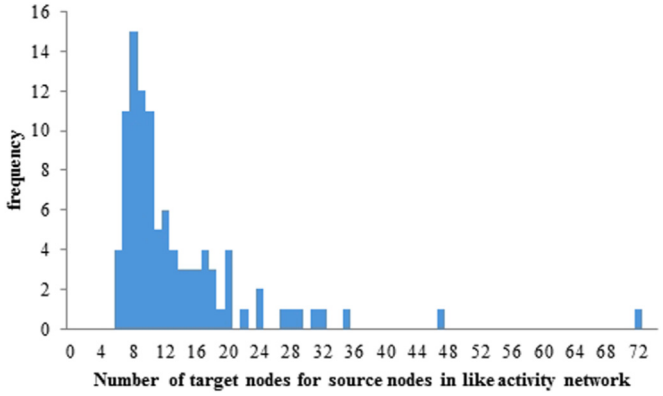


**Fig. 4.** Histogram of the number of target nodes for source nodes for explicit recommendation (like activity).

$$P@k(s) = \frac{|Targets(s) \cap top_k|}{k} \tag{9}$$

In this formula $Targets(s)$ is the set of real target nodes for source node and $topk$ is the set of $k$-best nodes predicted for source node. The recall at $k$-best nodes is called $R@k$. It is the ratio of the number of correctly predicted nodes in the $k$-best nodes to the number of real target nodes. Eq. (10) shows the $R@k$ for a given source node $s$.

$$P@k(s) = \frac{|Targets(s) \cap top_k|}{Targets(s)} \tag{10}$$

Similarly, $F1@k$ is the measure $F1$-Score for $k$-best nodes (Eq. (11)).

$$F1@k = 2\frac{P@k \times P@k}{P@k + P@k} \tag{11}$$

The other measure considered in this paper is the reciprocal rank. It is equal to $P@t$, in which $t$ is the rank of highest scored real target node among the scored nodes. The higher the score of the first real target node is, this measure will be higher. The mean of RR for all source nodes is shown by MRR. In this paper, we consider $k$ to be 10 and $d$, which is the number of real target nodes for each $s$. The measure $P@d$ equals $R@d$. The harmonic average of these two, accordingly, is equal to themselves.

## 6. Experimental results

In this section, we compare the proposed methods with other popular methods. These methods include weighted random walk starting from the source node, unweighted random walk, Directed Common Neighbors (DCN) [21], and Directed Adamic Adar (DAA) [21]. The random walk follows a randomly outgoing link of a node and at each step it returns to the initial node, $s$, with the probability 0.15. When we reach a node at each step, we score that node. In addition, if we reach a node without an outgoing link, we return again to node $s$. Upon selecting neighbors of the node, if a uniform probability is used, it will be unweighted random walk (UW-RW) and if the probability of selecting a neighbor is based on the weight of links, it will be weighted random walk (W-RW). Although methods CN and AA have been initially defined for undirected graphs, since the activity graph is directed, we use the directed version of both, which is presented in [21].

As stated before, we deal with two types of activity recommendation-Implicit and explicit. For evaluating different methods for implicit activity recommendation, mixed activity network has been used. Then, recommendation of like activity has been considered as a case study of explicit activity recommendation. For this purpose, multilayer activity network has been used in ML-C-RW algorithm, and single layer like activity network has been used in other algorithms.

### 6.1. Implicit activity recommendation

As previously mentioned, in implicit activity recommendation,

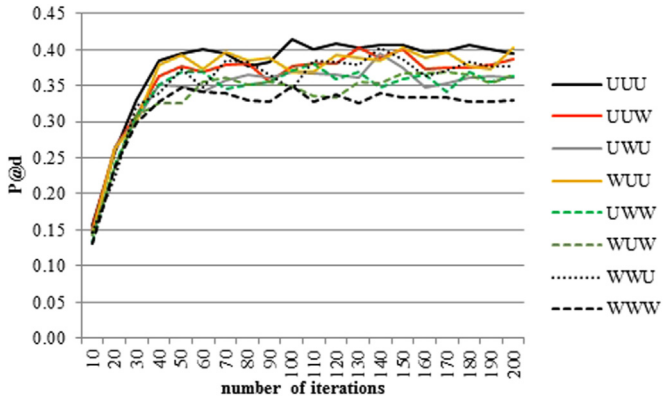


**Fig. 5.** Comparison of 8 different weighting methods for collaborative random walk.

the system recommends the user to do an arbitrary activity on the profile of another user. The user may decide the type of activity. So, during the experiments the algorithms must predict the occurrence of activity-not the type of activity-between users. In this case the link prediction in mixed activity network will be a good simulation method.

### 6.1.1. Impact of link weights on collaborative random walk

In contrast to simple random walk, the collaborative random walk moves on collaborative paths. Each collaborative path consists of three links; therefore, the collaborative random walk can become weighted in different ways. Any of the three links in the collaborative path can be either weighted or unweighted. In this way, there will 8 different modes. Fig. 5 compares these 8 modes in terms of *P@d* through 10–200 steps, where *d* is the number of real target nodes for each of the source nodes. In this figure, UUU represents the mode when all the three links of the collaborative path are considered unweighted. In UUW mode, the first two links are unweighted and the last link is weighted in the collaborative random walk. It is seen in this figure that the best mode is when all the links are unweighted.

As seen in Fig. 5, in all 8 modes of collaborative random walk method, *P@d* is improved up to a limited number of iterations. Most of the methods reach this limit at steps 40–60. After reaching this state, with increasing the number of steps, the value of *P@d* of each method rises and falls between two particular values. It can be stated that through steps 100–200, the methods do not gain considerable improvement. By this experiment we choose the best mode of collaborative random walk method to consider in comparisons of next experiments. This experiment only shows the adverse effect of considering weights of links in collaborative random walk method in the field of activity recommendation. The reason behind this negative effect has been investigated in the next experiments.

### 6.1.2. Comparison of different random walk methods in different steps

Random walk based algorithms have an input parameter, called maximum iterations. By increasing this parameter, the results of predictions will be improved. After comparing random walk based algorithms with other algorithms we need to find out the best maximum iteration. So, the effect of walking iterations on different random walk algorithms will be discussed in the experiment below.

Fig. 6 shows the *P@d* measure for random walk based algorithms in different iterations. Herein *d* is the number of real target nodes for each source node. The unweighted collaborative algorithm is the best algorithm in all modes. In addition, all random

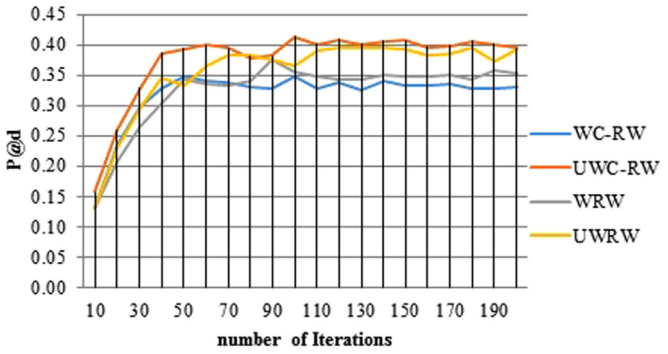


**Fig. 6.** Comparison of walking based methods from 10 to 200 iterations, in terms of *P@d*.

walk algorithms do not gain a special improvement in their results after a particular number of iterations. Furthermore, considering the weight of activities in random walks has negative impacts on the results of the algorithm. The reason has been considered at the next experiments. Fig. 6 shows collaborative and simple random-walk methods. Both methods can be weighted or unweighted. Similar to collaborative random-walk methods, simple random-walk methods after a while of random walk reach a mode when they will not have any improvement and rise and fall between two particular values. For the simple random walk also, the unweighted mode is better than the weighted.

As we mentioned previously, mean reciprocal rank is another important evaluation measure. Fig. 7 shows the value of MRR for different algorithms, in different walking steps. According to this measure, it is seen that the efficiency of the unweighted collaborative random walk method is better than the others. Also unweighted collaborative random walk has reached the best values.

### 6.1.3. Comparison of all methods

Table 3 compares the presented algorithms with the other methods of link prediction using different evaluation measures. In this experiment 9 months before than 28th month have analyzed to predict activities of 28th month. For random-walk algorithms, we have presented the average of each value from 100 to 200 steps.

It can be inferred from Table 3 that the proposed methods outperformed the baseline link prediction methods according to different measures. Based on the measure *F*1@10, tNum, Collaborative Random-Walk and Collaborative Association rule with the function $f_2$ were the best respectively. In addition, in terms of *P@d*, unweighted collaborative random walk and collaborative association rule with functions $f_2$ were respectively the best. On the other hand, the method *tNum* has often performed better than *uNum*.

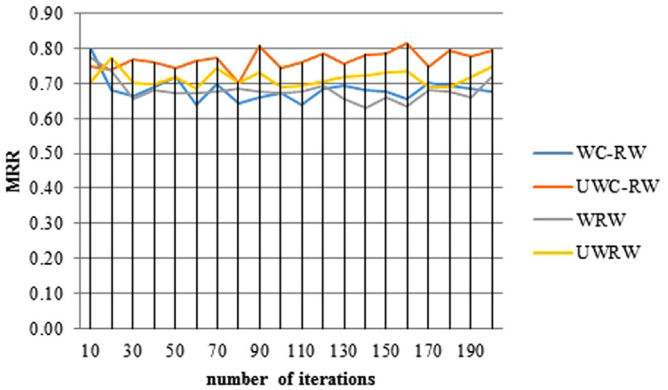


**Fig. 7.** Comparison of walking based methods from 10 to 200 iterations, in terms of MRR.

**Table 3**
Comparing proposed algorithms with other link prediction methods.

| Algorithm | *P@d* | *P*@10 | *R*@10 | *F*1@10 | MRR |
|---|---|---|---|---|---|
| UW-C-RW | 0.403 | 0.468 | 0.366 | 0.411 | 0.777 |
| W-RW | 0.349 | 0.385 | 0.315 | 0.347 | 0.668 |
| UW-RW | 0.388 | 0.450 | 0.355 | 0.397 | 0.713 |
| C-AR-f1 | 0.388 | 0.438 | 0.344 | 0.386 | 0.783 |
| C-AR-f2 | 0.391 | 0.449 | 0.359 | 0.399 | 0.750 |
| DAA | 0.272 | 0.31 | 0.246 | 0.274 | 0.533 |
| DCN | 0.332 | 0.364 | 0.288 | 0.321 | 0.586 |
| uNum | 0.280 | 0.329 | 0.257 | 0.289 | 0.566 |
| tNum | 0.360 | 0.535 | 0.343 | 0.418 | 0.692 |
| tuNum | 0.353 | 0.459 | 0.317 | 0.375 | 0.640 |
| CPNum | 0.388 | 0.438 | 0.344 | 0.386 | 0.783 |

This shows that the taste similarity between users with the similar interests with the source node is more important than the number of people with the similar interests in collaborative paths. In other words, the number of similar contents is more important than similar people in the collaborative path.

Compared to other fields of information retrieval, the values of precision and recall are not high. In [3] the best results for 2 different undirected networks were 4.25 and 7.57 in terms of non-normalized precision at 20 metric. After normalization these values will be 0.21 and 0.38. It may be because the experiments for evaluating the link prediction algorithms are all in offline mode. When an algorithm recommends $k$ nodes for the user $s$, actually we do not present the list for the user $s$. Instead we look at the user's choices in the next month and compare them with our list. There may be some nodes in the presented list, which user $s$ was not aware about in that time, but he was interested to do activities on such users. So it is clear that the results of offline experiments for all algorithms will not be high.

### 6.1.4. Negative impact of link weights

As stated in Section 6.1.1, considering link weights has negative effect on the results of prediction algorithms. However, it is intuitively plausible that weighted networks are more informative than unweighted ones. This section will discuss the reason why the weighted methods cannot perform better in activity recommendation. In previous experiments, an activity network of a 9-month period was analyzed to predict the activities of the 10th month. During this analysis, the weight of a link does not give any information about the time of each activity corresponding to that link. Suppose the case of two links $E1$ and $E2$, which $weight(E1) < weight(E2)$, while activities corresponding to link $E1$ have occurred in the 9th month and activities corresponding to link $E2$ have occurred in the first month. It is intuitively plausible that, to predict the activities of the 10th month, $E1$ is more important than $E2$.

In the previous experiments the length of the first period was 9 months. We can decrease or increase the length of this period. One way for analyzing the reason of the negative impact of link weights is to analyze the length of the first period. As previously stated in Table 1, by increasing the first time period, some old links appear. Also it is clear that some links will be weighted more than others. By increasing this period, the old activities will affect the weights of links.

Fig. 8 shows how the length of the first time period impacts on the *P@d* measure of different algorithms. For random walk based methods, mean value of 10 times each one with 500 iterations was calculated. We have analyzed 9, 15, 21 and 27 months before 28th month for predicting activities in these experiments. It is shown in Fig. 8 that these methods perform better when first time period is close to the target time. This shows the reason why unweighted random walks perform better than the weighted ones. When a link has more weight than other, its activities may be far from the target month. And also when a link has less weight, its activities may be occurred near the target month.

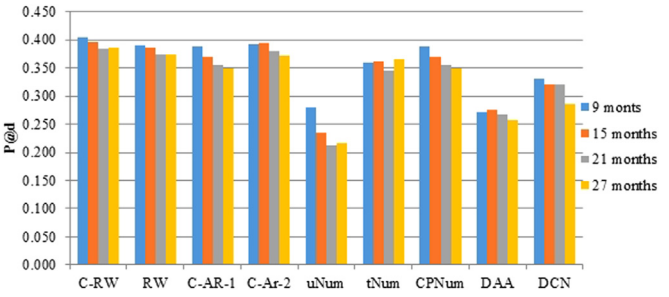


**Fig. 8.** Comparing different lengths for first time periods on algorithms.

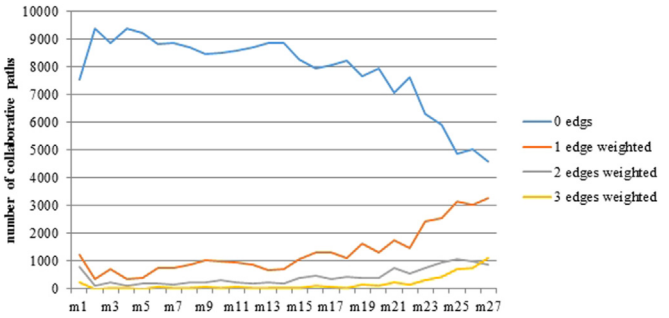


**Fig. 9.** Contribution of each month on weighting the predictive collaborative paths for 28th month.

The result obtained from this experiment can be proved by another experiment. This experiment shows that users' interests change over time. So for predictive algorithms, a user's interests in the last month will be more informative than their interests in the first months. Fig. 9 shows the impact of time passing on user's interests and behaviors. It shows how user's interests change over time from the first month to the 27th month. This experiment has analyzed the activity network of first 27 months. It has considered all collaborative paths, which truly predict the 28th month activities for 100 source nodes. These are predictive collaborative paths. Existence of a collaborative path from node $s$ to node $d$ shows that the user $s$ is interested to do activity on user $d$'s profile. Also each collaborative path is constructed from three weighted edges. Each of these three edges has been weighted in one or several arbitrary months. Each of these 27 months has weighted some of predictive collaborative paths. More predictive collaborative paths weighted in a single month shows user $s$'s interests on that month are more similar to his/her interests on the target month. Fig. 9 shows the contribution of each month on weighting the predictive collaborative paths for 28th month. It separately shows number of predictive collaborative paths which their 1, 2, 3 and 0 edges were weighted on each single month. We can infer from Fig. 9, the closer to the target month, months have more contribution on weighting 1, 2 or 3 edges of the predictive collaborative paths.

## 6.2. Explicit activity recommendation

The purpose of explicit activity recommendation is to recommend users to do a special activity on another user's profile. For each type of activity, we can construct an activity network. The link prediction on that network will be used to recommend that activity. We also presented a multilayer collaborative random walk algorithm, which uses multilayer activity network. In this paper predicting like activity has been considered as a case study for explicit activity recommendation.

**Table 4**
Comparing proposed algorithms with other link prediction methods for predicting like activity.

| Algorithm | *P@d* |
|---|---|
| Like-CRW | 0.38716 |
| ML-CRW | 0.38887 |
| Like-RW | 0.37414 |
| AR1 | 0.38928 |
| AR2 | 0.38631 |
| uNum | 0.28453 |
| tNum | 0.33852 |
| CPNum | 0.38928 |
| DAA | 0.25927 |
| DCN | 0.31356 |

**Table 5**
Pearson correlation coefficient between the out-degree of source nodes and *P@d* measure of source nodes for different algorithms.

| Algorithm | Pearson correlation coefficient |
|---|---|
| CRW | −0.08 |
| RW | −0.11 |
| AR1 | −0.10 |
| AR2 | −0.12 |
| uNum | −0.13 |
| tNum | 0.10 |
| CPNum | −0.10 |
| DAA | 0.07 |
| DCN | 0.02 |

Table 4 compares different methods for predicting like activity, in terms of *P@d* measure. Walking based methods have run 10 times with 500 iterations and the average *P@d* for each one is calculated. It should be mentioned that we considered unweighted version of walking based algorithms. ML-CRW algorithm has used multilayer network and all other methods have used like activity network. This experiment has analyzed the time period of 9 months to predict 28th month. This table shows that proposed methods have outperformed the popular link prediction methods. Multilayer collaborative random walk was a little better than other walking based methods. Also associate rule with $f_1$ measure was the best algorithm in terms of *P@d* to predict like activity.

### 6.3. Cold start problem

As discussed previously, collaborative filtering methods generally suffer from cold start problem. This happens when a user has presented just few opinions. Table 5 shows the Pearson correlation coefficient between the output degree of source nodes and *P@d* measure of source nodes for different algorithms. We can see in this table that the output degree of source nodes (number of presented opinions of them) approximately has not impact on the quality of proposed activity prediction algorithms. So our collaborative methods do not suffer from cold start problem.

## 7. Conclusion

One of the common recommending methods in online social networks is link prediction. For recommending common activities in Facebook like, comment, share and post to users, the problem of link prediction in the activity network was considered in this paper. For this purpose, we proposed a new specific path called "collaborative path". Then, based on this new concept we presented some proximity measures in directed networks. After that, we presented three new algorithms for activity prediction each of which can be used in weighted/unweighted and simple/multilayer modes. Experimental results show that the proposed activity prediction methods outperform the popular exiting link prediction methods according to *P@d*, *F1@d*, and MRR. In addition, since the presented methods do not utilize users profile information, combination of the proposed collaborative link prediction algorithms with analyzing users' attributes and their similarity by profile information and interesting keywords is recommended for the future work.

## References


[1] M. Agrawal, M. Karimzadehgan, C. Zhai, An online news recommender system for social networks, Urbana 51 (2009) 61801.
[2] J. Hannon, M. Bennett, B. Smyth, Recommending twitter users to follow using content and collaborative filtering approaches, in: Proceedings of the Fourth ACM conference on Recommender Systems, (ACM2010), 2010, pp. 199–206.
[3] L. Backstrom, J. Leskovec, Supervised random walks: predicting and recommending links in social networks, in: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, (ACM2011), 2011, pp. 635–644.
[4] Y. Yao, J. Zhou, L. Han, F. Xu, J. Lü, Comparing Linkage Graph and Activity Graph of Online Social Networks, International Conference on Social Informatics, Springer, Berlin Heidelberg, 2011, pp. 84–97.
[5] H. Chun, H. Kwak, Y.-H. Eom, Y.-Y. Ahn, S. Moon, H. Jeong, Comparison of online social relations in volume vs interaction: a case study of cyworld, in: Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement, (ACM2008), 2008, pp. 57–70.
[6] C. Wilson, A. Sala, K.P. Puttaswamy, B.Y. Zhao, Beyond social graphs: User interactions in online social networks and their implications, ACM Trans. Web 6 (2012) 17.
[7] B. Viswanath, A. Mislove, M. Cha, K.P. Gummadi, On the evolution of user interaction in facebook, in: Proceedings of the 2nd ACM Workshop on Online Social Networks, (ACM2009), 2009, pp. 37–42.
[8] J. Heidemann, M. Klier, F. Probst, Identifying Key Users in Online Social Networks: A PageRank Based Approach, 2010.
[9] A. Corbellini, S. Schiaffino, D. Godoy, Intelligent analysis of user interactions in a collaborative software engineering context, International Conference on Advances in New Technologies, Interactive Interfaces, and Communicability, Springer, Berlin Heidelberg. 2011, pp. 114–123.
[10] J. Fu, Y. Fan, Y. Wang, S. Wang, Network analysis of terrorist activities, J. Syst. Sci. Complex. 27 (2014) 1079–1094.
[11] E. Gilbert, K. Karahalios, Predicting tie strength with social media, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, (ACM2009), 2009, pp. 211–220.
[12] K. Panovich, R. Miller, D. Karger, Tie strength in question and answer on social network sites, in: Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work, (ACM2012), 2012, pp. 1057–1066.
[13] D. Liben-Nowell, J. Kleinberg, The link-prediction problem for social networks, in: Conference on Information and Knowledge Management (CIKM'03), 2003, pp. 556–559.
[14] L. Katz, A new status index derived from sociometric analysis, Psychometrika 18 (1953) 39–43.
[15] L.A. Adamic, E. Adar, Friends and neighbors on the web, Soc. Netw. 25 (2003) 211–230.
[16] Y. Koren, S.C. North, C. Volinsky, Measuring and extracting proximity graphs in networks, ACM Trans. Knowl. Discov. Data 1 (2007) 12.
[17] S. Wu, J.M. Hofman, W.A. Mason, D.J. Watts, Who says what to whom on twitter, in: Proceedings of the 20th International Conference on World Wide Web, (ACM2011), 2011, pp. 705–714.
[18] F. Wu, B.A. Huberman, Novelty and collective attention, Proc. Natl. Acad. Sci. 104 (2007) 17599–17601.
[19] R.N. Lichtenwalter, J.T. Lussier, N.V. Chawla, New perspectives and methods in link prediction, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, Washington, DC, USA, 2010, pp. 243–252.
[20] M. Mohajireen, C. Ellepola, M. Perera, I. Kahanda, U. Kanewala, Relational similarity model for suggesting friends in online social networks, in: 2011 6th IEEE International Conference on Industrial and Information Systems (ICIIS), (IEEE2011), 2011, pp. 334–339.
[21] K. Lerman, S. Intagorn, J.-H. Kang, R. Ghosh, Using proximity to predict activity in social networks, in: Proceedings of the 21st International Conference Companion on World Wide Web, (ACM2012), 2012, pp. 555–556.
[22] C. Scholz, M. Atzmueller, G. Stumme, Link Prediction and the Role of Stronger Ties in Networks of Face-to-Face Proximity, CoRR, abs/1407.2161, 2014.
[23] X.N. Lam, T. Vu, T.D. Le, A.D. Duong, Addressing cold-start problem in recommendation systems, in: Proceedings of the 2nd International Conference on Ubiquitous Information Management and Communication, (ACM2008), 2008, pp. 208–211.

[24] R. Agrawal, T. Imieliski, A. Swami, Mining association rules between sets of items in large databases, in: ACM SIGMOD Record, (ACM1993), 1993, pp. 207–216.
[25] X. He, F. Min, W. Zhu, Top-N Recommendation Based on Granular Association Rules, Rough Sets and Knowledge Technology, International Publishing, Springer, 2014, pp. 194–205.

**Amin Shahmohammdi** is a M.S. student in Software Engineering in Pooyesh Institute of Higher Education, Qom, Iran. He received his B.S. degree in Computer Engineering from Islamic Azad University, Kaleibar Branch, Iran in 2013. His current research interests include Data Mining, Machine Learning, Social Network Analysis, Distributed Systems, Optimization, and Parallel algorithms. His programming experiences are in R, and C++ languages.

**Ehsan Khadangi** received his M.Sc. in Information Technology from Amirkabir University of Technology in 2009. He is currently a 5th year Ph.D. student in Computer Engineering at Amirkabir University of Technology and simultaneously a Lecturer at Qom University. His current research interests mainly focus on Social Media Marketing, Social Network Analysis, Link Prediction, Data Mining on Social Networks, Viral Marketing, Recommender Systems, and etc.

**Alireza Bagheri** received his B.S. and M.S. degrees in Computer Engineering from Sharif University of Technology (SUT) at Tehran. He received his Ph.D. degree in Computer Science from Amirkabir University of Technology (AUT) at Tehran. Currently he is an Assistant Professor in the Computer Engineering and IT department at Amirkabir University of Technology (AUT) at Tehran. His research interests include computational geometry, graph drawing and graph algorithms.