INTELLIGENT AUTOMATION & SOFT COMPUTING, 2017
https://doi.org/10.1080/10798587.2017.1355656

Friends Classification of Ego Network Based on Combined Features

Jing Jia[a], Tinghuai Ma[b], Fan Xing[a], William Farah[a] and Donghai Guan[a,c]

[a]School of Computer & Software, Nanjing University of Information Science & Technology, Nanjing, China; [b]CICAEET, Jiangsu Engineering Centre of Network Monitoring, Nanjing University of Information Science & Technology, Nanjing, China; [c]School of Computer, Nanjing University of Aeronautics & Astronautics, Nanjing, Jiangsu, China

ABSTRACT

Ego networks consist of a user and his/her friends and depending on the number of friends a user has, makes them cumbersome to deal with. Social Networks allow users to manually categorize their "circle of friends", but in today's social networks due to the unlimited number of friends a user has, it is imperative to find a suitable method to automatically administrate these friends. Manually categorizing friends means that the user has to regularly check and update his circle of friends whenever the friends list grows. This may be time consuming for users and the results may not be accurate enough. In this paper, to solve this problem, we present a method, which combining user attributes, network structure and contact frequent three aspects. Efficiently using the profile of users, we first identify the relationship between them and then we attempt to solve the problem of community identification when a user's profile is missing or inaccessible by use of ego network structural features. Lastly, to obtain more accurate results and realize updates automatically, we attempt to find those friends who have frequent contacts with the user. We compare the performance of the proposed algorithm with other methods, and the results show that our method has significant advantages to them.

KEYWORDS

Ego networks; Circle of friends; Combined feature; Automatically

1. Introduction

Social networks have gained much interest recently and the reason being mainly, because of its relation with a person's social activities (Kossinets & Duncan, 2006; Liu, Yang, Wang, et al., 2015). For this reason, people tend to spend more time on social network sites, browsing contents and enjoying the various streams of services offered by Social Network Services (SNS) (Cheng & Yan, 2014). Most social networks allow users to interact with each other in different kinds of ways from business, education and entertainment. In recent years, the size of a social network has been measured primarily by the number of users on the network as well as the number of friends and acquaintances a user has. The growth in volume of a user's friend has changed from about hundreds of people to around thousands of people and that is just for a single user. With that being said, how will social network sites enable users efficiently and effectively administrate their circle of friends should the need arise. In practice, several big social network sites have made available such functions. For example, Google+uses "circles" as a means for a user to organise friends and then comes "lists", which is adopted by Facebook and Twitter. Social circles are used as mechanisms that enable users of the mentioned social networks to organize their friends and the contents they receive and deliver to them. Social circles can also be used to filter contents as well as protecting a user's privacy, information sharing and a host of others. When users create their personal social circle for their friends, they are able to give different permissions to different kinds of friends within their social circle. Only then will they realize the true purpose of managing friends within the social circle effectively.

Presently, Google+, Twitter and Facebook users categorize their friends either manually or by recommendation from SNS. However, neither approach is peculiarly satisfactory in that the former is time consuming and needs to manually divide whenever a user adds more friends, whereas the latter just depends on the profile similarity between a user and friends to identify the relation, which leads to inaccurate results and also cannot categorize accurately or complete the categorization when certain attributes are missing. This is the reason why several studies on social networks concentrate on how to effectively divide the whole network into several sub communities (e.g. community detection). Two main types of methods are developing rapidly, thus one based on nodes profile and the other is based on the network structure. The node profile method focuses on fully analysing a users' information to find common features among them and then assigning similar users into the same community. Several different algorithms have been proposed in previous social networking literature (McAuley & Leskovec, 2014; Gao & Bettina, 2013; Ma, Rong, Ying, et al., 2016), and some in complex network theory (Lv et al., 2016; Wadhwa & Bhatia, 2014; Coscia, Giannotti, & Pedreschi, 2011) or elsewhere. However, none has yet to solve the question of a user with less or even no personal profile on a social network. The other predominant approach takes advantage of the structural features of a network to find the connection relation among users meanwhile, each community consist of users who are linked together with others. Some literatures such as (Fan, Yeung, & Fan, 2015; Miao et al., 2014; Yuan et al., 2010; Zhang, 2014) have done lots of work on this however, the personal information is important for identifying the relation among people, hence the result of this type of method may be not accurate enough.

In this paper, we propose a method, which will automatically update a user's circle of friends on any given social network; in addition to that our method improves the accuracy of results.

CONTACT Tinghuai Ma thma@nuist.edu.cn

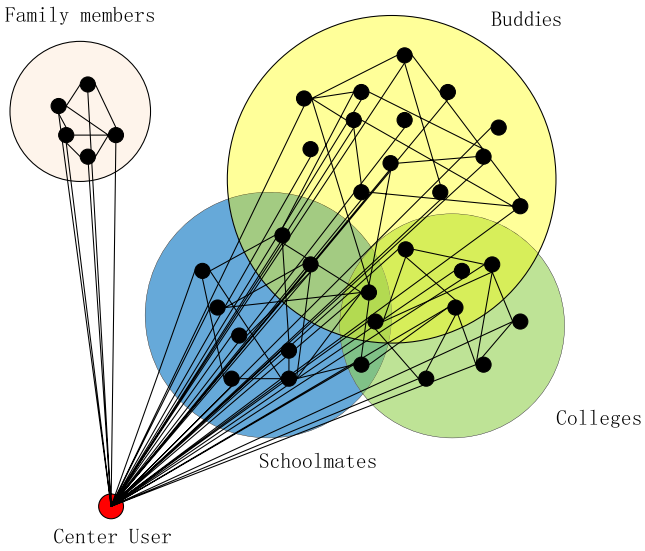


Figure 1. An Ego Network.

For a given social network user referred to as "centre user", we detect social circles in his/her ego network where the nodes and edges in the ego network represent users and their connections respectively, bearing in mind that there may be some personal information of users. Social circles are similar to communities within networks and are defined as a subset of friends with common features. For example as shown in Fig. 1, an ego network, can be divided into several other circles, which contain some friends of the centre user and other people in different circles who have different relationships with the centre user. The size of each circle is not the same, because the number of different categories in the centre user's friends is uneven. It is also worthwhile to mention that some friends may not be in only one circle, which means that they may have multiple relationships with the centre user.

We focus on these three aspects in our research; the similarity of user attributes, the features of network structure and the contact frequency between the centre user and friends. We first divide all friends by the similarity of properties between each friend and the centre user and then we utilize the characteristics of an ego network's structure to categorize friends whose attributes are incomplete or missing. Lastly, we analyse the frequency of contact between the centre user and friends to reflect interactions between them and then we find out people who constantly keep in touch with the centre user and place them in a special circle.

2. Related Work

Many researchers have commented on the community detection problem and even though our work involves a user's circle of friends, it still can be categorized under community detection. Newman (2004) summarized some common approaches and presented the Fast-Newman algorithm on offline social network based on the GN method. With the progress of online social networks, increasing studies have diverted focus on how to detect communities in online social networks. Du et al. (2007) introduced several traditional community detection methods applied on the online social network. Ferrara (2012) discussed the optimization approach based on modularity applied on Facebooks network data-set, which needed high computation complexity. Being of high volume of data, the large data-set is becoming important, so Du et al. (2007) proposed a detection algorithm, which could make use of overlapping communities on a large scale network. Later, Ma, Wang, et al. (2016) presented new fast overlapping communities detection algorithm based on structure, which is verified on C-DBLP datasets.

Moving on, ego networks are important types of networks in social networks where "ego" is an individual centre node. McAuley and Leskovec (2014) first studied the problem of social circles determination and formulated it as a social circles identification problem. Most of the work in this field was done based on clustering of members of an ego network relying on either the network structure or the user profile.

2.1. Identifying Circles based on a Network Structure

Arnaboldi et al. (2012) analysed the structural characteristics of ego networks and made the conclusion that online ego networks have similarities with offline social networks. Both networks mainly consisted of four hierarchies of a friends' circle. Hu and Yang (2015) proposed a new method for social circles identification on ego networks, which integrates node features and network structure by constructing an edge profile for each edge. The utilization of both node features and the network structures information makes the proposed method more effective. It uses edge similarity instead of node similarity to distribute nodes into different circles. Miao et al. (2014) proposed a novel, DC-S, which is a circle detection algorithm. Also, some research focuses on how to modify the structure for privacy preserving. Ma, Zhang, Cao, et al. (2015) proposed a structure changed methods using Vertex and Edge modification. Furthermore, Rong et al (2017) use graph similarity detection to implement sub graph anonymity. Unlike in community detection the detected circles on the ego network largely based on the structure information from the view of an ego node.

2.2. Identifying Circles based on a User Profile

Mislove et al. (2010) gathered data from two social networks and tried to infer user profile attributes. From their research, they found that users with common attributes were most likely friends and often form dense communities so they proposed a method of inferring user attributes that was inspired by previous approaches to detecting communities in social networks. Kim et al. (2010) crawled Twitter list data that includes about ten percent of the Twitter user population, the lists they belong to, and the tweets of all the members of the lists. They used a standard feature selection algorithm and a supervised classification algorithm to verify the semantic coherence of the lists. They then conducted a user survey, which confirmed that lists served as good groupings for Twitter users with respect to the perceived characteristics of the users. Yoshida (2013) proposed the utilization of a graph structure (called a profile graph), which is constructed via profile data and then suggested a simple model to utilize both the observable connectivity relation and the profile graph. Furthermore, instead of a hierarchical approach, which is based on the modularity matrix of a network structure, they proposed an embedding approach, which utilizes the regularization via the profile graph. McAuley and Leskovec (2014) presented an unsupervised learning method to find latent circles of the ego network based on the additional user information called DC-M. In their method, personal information is extracted as input features and the circles represented as different social meanings in the detecting model.

In the above methods, researchers tend to use either the network structure or the user profile to identify circles. The former's accuracy is a bit unsatisfactory and the latter performed poor under different circumstances. Find the best friend circle is an optimization problem; Xue et al. (2017) proposed a self-adaptive artificial bee colony algorithm based on global best for global optimization. The typical methods among them are DC-S and DC-M and we will compare them in later chapters.

3. Hybrid Method Algorithm for Identifying Circle of Friends

3.1. Basic Problem Definition

We abstract the ego network into an undirected graph $G(V, E)$, where V represents the nodes set, the social networks users set is represented as $|V| = n$, where n is the number of nodes; $E \in V \times V$ represents a collection of edges between nodes in V, which is the relationship between any users, $|E| = m$ is the number of edges.

Definition 1: (User attributes set) For each node $v_i \in V$, we define the set of all attributes of v_i as $Attr_{v_i} = \{A_1, A_2, A_3, \ldots, A_p\}$, where A_j $(1 \le j \le p)$ represents an attribute.

An exception is made for some attributes that are divisible into multiple values, e.g. $A_j = \{a_1, a_2, a_3, \cdots, a_{jp}\}$, where jp represents the number of sub-attributes of A_j.

Definition 2: (Centre user) For the owner of an ego network, we define it as the centre user, denoted by $Cu \in V$.

Definition 3: (Friends set) For each node except Cu in V, we define a friends set $F = \{f_1, f_2, f_3, \cdots, f_{n-1}\}$, where f_k $(1 \le k \le n - 1)$ represents a friend of a centre user Cu.

Definition 4: (Centre user's attribute vector) For all attributes in $Attr_{Cu}$, we merge them to form the row vector of a centre user's attribute, called $Vector_{Cu} = [Value_{A_1}, Value_{A_2}, Value_{A_3}, \cdots, Value_{A_p}]$.

Definition 5: (Friends' attribute matrix) For all attributes in $Attr_F$, which means friends' attributes set, we merge them to form the matrix of a friends' attribute, called $Matrix_F$, as shown in Fig. 2.

3.2. Proposed Algorithm

In this section, we describe in detail our proposed circle of friends' detection method. First, we compare the similarity in attributes between a centre user and friends. We divide all friends by the similarity of properties according to the comparison of single attributes and multiple values of attributes. Secondly, for a friend whose attributes are incomplete or missing, we utilize the characteristic of an ego network's structure to categorise them further. To determine if a node, which represents a single friend and should be added into a circle is decided by two aspects; the probability of one node being added into a circle and then how the F value changes if the node is added into the circle. Lastly, we analyse the frequency of the contact between a centre user and friends, which will represent the interactions among them in order for us to find out the people who are constantly in touch with the centre user and place them in a specialized circle.

$$\begin{array}{c} \\ f_1 \\ \\ \\ \\ f_{n-1} \end{array} \begin{bmatrix} Value_{A_1} & Value_{A_2} & Value_{A_3} & \cdots & \cdots & Value_{A_p} \\ Value_{A_1} & Value_{A_2} & Value_{A_3} & \cdots & \cdots & Value_{A_p} \\ Value_{A_1} & Value_{A_2} & Value_{A_3} & \cdots & \cdots & Value_{A_p} \\ \vdots & & & & & \\ \vdots & & & & & \\ Value_{A_1} & Value_{A_2} & Value_{A_3} & \cdots & \cdots & Value_{A_p} \end{bmatrix}$$

Figure 2. Friends' Attribute Matrix.

The process of circles detected are step by step and the subsequent one is on the basis of the previous. So, we can achieve a result, which is complete and reliable.

3.2.1. Profile Based Similarity

Every user in on a social network has various attributes ranging from name, company, education, hobby, and so on. The relationship between a centre user and each friend depends on the similarity between different properties. For example, the relationship among colleagues needs the same profession between them, but the schoolmate relationship requires the same school in the education experience. In order to distinguish all kinds of different relationships between a centre user and friends, we partition each circle according to different attributes. e.g., workmates in the same circle is due to the same profession, but the same education experience for another circle. In this way, some of the friends are assigned to the circle, because they have the same attribute as the centre user. Also, it is worth mentioning that a single friend may be assigned to more than one circle since he/she may have at least one same property as the centre user.

The number of attributes for a given data is large especially when the number of user data is much less than their attributes and this may lead to inconsistencies. On one hand, excessive dimensionality will take a longer time when there are only a few users. On the other hand, the significance of each user attribute cannot certainly be the same so therefore, we need to select those attributes, which can provide important information first and denote them as primary attributes. During this process, we can also reduce the dimensionality of certain attributes should the need arise.

After selecting attributes, we will compare them to identify the relationship between a centre user and friends.

3.2.1.1. Attribute Selection

One of the methods of attribute selection is information gain. Information gain is simply defined as the number of entropy reduction from one attribute. Information gain can be calculated as below:

$$IG(A) = H(S) - \sum_{t \in T} p(t)H(t) \tag{1}$$

Where $H(S)$ represents entropy, A is an attribute and $p(t)$ is the subset created from splitting S by attribute A. Entropy can be computed as follows:

$$H(S) = -\sum_{x \in X} p(x) log_2 p(x) \tag{2}$$

In which $p(x)$ is the proportion of number of elements in class x to the number of elements in set S.

So, the primary attributes will be picked out with high value of information gain.

3.2.1.2. Attribute Comparison

For whether two same attributes of a centre user and a friend are similar, we give a definition as follows:

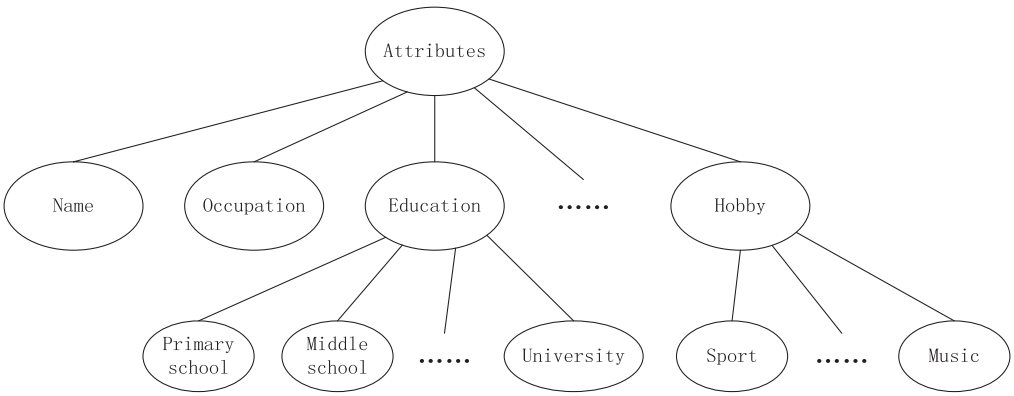


Figure 3. Users' Attributes

If the attribute $A_z (1 \le z \le p)$ is a single value, then when the two values of A_z are similar, then we say these two attributes are similar.

For example, there are four values in attribute "political" to be selected for users and everyone can only choose one. So, if two users have the same option for "political", we say their "political" attributes are similar.

If the attribute A_y $(1 \le y \le p)$ is made up of multiple values, we calculate the ratio of the similar values of A_y. By the result of this ratio, we can assess the different similarities between users' attributes.

Another example is there are six values in attribute "hobby" (as shown in Fig. 3) to be selected by users and everyone can choose more than one. Let's say two users have the same three options of "hobby", so we say the ratio of the similar values of their attributes "hobby" is 0.5.

For some special attribute, such as age, we divide the value into several intervals; childhood, teenage, youth, midlife, old age, includes 0 to 10, 11 to 17, 18 to 30, 31 to 50 and greater than 50 respectively. If two users belong to same intervals, we say they are similar of this attribute.

According to preliminary works and definition, we are beginning to find potential circles in the network and add friends into them.

3.2.1.3. Circle Discovery

We define an operation for the matrix of friends' attributes set and the matrix of a centre user's attribute to obtain a matrix of the attributes' result called $Matrix_{R,}$ of m rows and p columns, where $Matrix_R[b][d] (1 \le b \le m; 1 \le d \le p)$ can be computed as follows:

$$Matrix_R[b][d] = \begin{cases} e & Matrix_F[b][d] \cap Vector_{Cu}[d] \neq \emptyset \\ 0 & Matrix_F[b][d] \cap Vector_{Cu}[d] = \emptyset \end{cases} \quad (3)$$

Where $Matrix_F[b][d]$ is the value of $b-$ th friend's and $d-$ th attribute, also $Vector_{Cu}[d]$ is the value of the centre user's $b-$ th attribute. The "e" is the ratio of similarity representing the similarity between the $b-$ th friends and the centre user of $d-$ th attribute. The "0" means they are dissimilar.

For each primary attribute, as long as its value of some friend is similar to that of a centre user, this friend is added to a circle, which is created according to this primary attribute. For other attributes (not primary attributes), we define a threshold value α as minimum number of similar attributes to decide whether to add into a circle namely buddy, or not.

$$\begin{array}{c} f1 \\ f2 \\ f3 \\ f4 \\ f5 \\ f6 \end{array} \begin{bmatrix} 0 & 1 & 0 & 0 & \frac{3}{5} & 0 \\ \frac{1}{3} & 0 & 0 & 1 & \frac{2}{5} & 0 \\ 0 & 0 & 0 & 1 & \frac{2}{5} & \frac{1}{4} \\ 0 & 1 & \frac{1}{2} & 0 & \frac{1}{5} & 0 \\ \frac{2}{3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{3}{4} & 1 & 0 & \frac{1}{2} \end{bmatrix}$$

Figure 4. Attribute Result Matrix.

The following is the process of discovering a circle:

(1) Read the matrix of friends' attribute set. Read the matrix of a centre user's attribute.

Performing operations using formula (3) on the two input matrixes to obtain the matrix of attribute result. We compare each friend to center user, e.g., comparing each row of $Matrix_F$ to $Vector_{Cu}$ respectively. Then the result is added into attribute result matrix corresponding one by one.

(2) Iterate the columns of $Matrix_{R,}$ which represents primary attributes to find every nonzero number, which means the friend has the similar primary attribute as centre user. So, result, this friend will be classified into center user's friend circle. This will be terminated when all primary attributes columns are checked.
(3) Iterate all the rows of non-primary columns in $Matrix_R$ until all friends' rows are checked. We calculate the sum of values for each row and remember the result as the friends' np-value.

If one friend's np-value is not less than α, we assign him into buddy circle.

As Fig. 4 shows, the (*1st–3rd*) columns are primary attributes, the (*4th–6th*) columns are non-primary attributes.We set

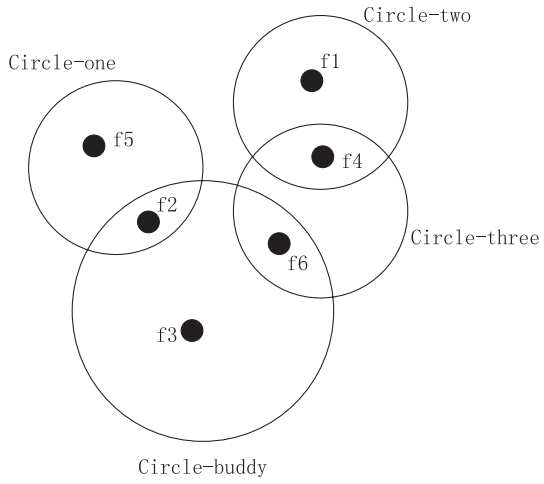


Figure 5. Example Result.

the value of α to 1, and the values of elements in the attribute result matrix is shown in the Figure. The f_2 and f_5 have the similar first primary attribute as center user, so they are added into circle-one. In the same way, the f_1 and f_4 are added into circle-two, and the f_4 and f_6 are added into circle-three. The np-value (the last three columns of $Matrix_R$ in Fig. 4) of six friends from f_1 to f_6 are 3/5, 7/5, 33/20, 1/5, 0, 3/2 respectively. Hence, the f_2,f_3 and f_6 are added into the circle-buddy, because their sum of np-value is not less than α. The result is shown in Fig. 5.

3.2.2. Network Structure based Similarity

For friends whose attributes are missing or incomplete and as a result, do not get assigned to any circle, we propose a method, which decides to assign them to any circle according to the network structure. By utilizing the structure features, we define two measurements to judge whether each friend should be added to any circle or not. In order to aid understanding, we use a node to represent a friend; the first metric is considered to evaluate the probability of one node being added into a circle. We introduce similarity to solve this problem and the similarity between one node and a circle is decided by the ratio of the number of friends of this node in this circle to that of all friends. So therefore we define the probability of one node to be added into a circle as follows:

$$P(f, circle) = \frac{neighbor(f) \cap circle}{neighbor(f)} \tag{4}$$

Where f denotes a friend who does not belong to any circle of a centre user, $neighbor(f)$ and $circle$ represents the neighbor nodes of f and all nodes in the circle respectively. The higher the P value, the more likely it is of adding f into a circle. We define a threshold value β as the minimum value meeting the condition for adding.

In order to enhance the rationality of this process and the accuracy of the results, another metric F value is introduced and it can be calculated as shown below:

$$F(circle) = \frac{neighbor(circle) \cap (f \notin circle)}{neighbor(circle)} \tag{5}$$

Where $neighbor(circle)$ are those nodes, which are connected with a circles inner nodes. The F value reflects the independence of a circle, which means the rationality of it being divided. In that sense, in order to determine whether the new node will be added, we give a condition as follows:

$$F(circle') \leq F(circle), \quad circle' = circle \cap node \tag{6}$$

If the F value of the new circle is not greater than the original one, then the independence of circle is not affected by the newly added node hence making $_{\text{this}}$ addition acceptable.

```
Algorithm 1 Circle detection based on structure
Input: All already existing circles; all candidate nodes;
Output: All new circles
1. for circle in all circles:
2.    compute F
3.    for node in all nodes:
4.       if ∃node(neighbor)∈circle:
5.          compute P(node, circle)
6.          if P>β:
7.             compute F(circleUnode)
8.             if F(circleUnode)<F(circle):
9.                circle=circleUnode
10.                  compute F
11.               end if
12.            end if
13.         else:
14.            continue
15.         end if
16.      end for
17. end for
18. return circles
```

As shown in algorithm 1, in the pseudocode, all the circles created in the previous method the nodes, which do not belong to any circles nodes–denoted as candidate nodes are considered as input data. In the middle of our algorithm, every circle tends to merge with the nodes that meet the requirements.

The algorithm first computes the F value of the initial circle and traverses all of the candidate nodes to find the node that has neighbors in this circle. Secondly, the probability P of a candidate node will be calculated and if the result meets the requirement of the threshold, then it will calculate the F value of a new circle, which merges the initial circle with the node. This is only possible if and only if the new F value is not greater than the previous value, then the node could be added into the circle.

3.2.3. Identifying Circle of Friends based on Contact Frequency

In reality, people change with time and so does their relationships with other people. It cannot be guaranteed that two friends in a close relationship may remain close friends forever within a specific time period so therefore, we introduce a parameter called contact frequency, which takes this feature into account in order to make generated results more reasonable and feasible.

In SNS, users contact can make contact with others via messaging, commenting on a post, reposts, mentions and likes. The contact frequency is higher if more connections are made during this period. We should bear in mind that the relationship between friends must be close to achieve this.

In this paper, we assume different kinds of connection that are not the same and it is of no relevant importance to our work. For example, users can use text message to discuss private matters among themselves or things considered personal. However, users may like a post from any of their friends no matter the relationship between them. On this note, we summarize the characters of every contact and assign them different weights in Table 1.

Table 1. The Characters and Weight of Contacts.

Contact Way	Private or Public	To User or Content	Written Communication	Weight
message	private	user	yes	ω_1
comment	public	content	yes	ω_2
repost	public	content	yes	ω_3
mention(@)	public	user	yes	ω_4
Favorited (Like)	public	content	no	ω_5

From Table 1, for a given period, the value of contact for each friend is the total of each contact times multiplied by its weight, thus;

$$C_V = \omega_1 \times \text{message}(f, Cu) + \omega_2 \times \text{message}(f, Cu) + \omega_3 \times \text{report}(f, Cu) + \omega_4 \times \text{mention}(f, Cu) + \omega_5 \times \text{likes}(f, Cu) \quad (7)$$

Where the value of each contact way (e.g., message (f, Cu), comment (f, Cu), etc.) corresponds with the number of times and the default value of their weights are depending on importance.

We set the value of ω_1 to 1, and it has features private to user and written communication. For others, if they are public, the value reduces 0.2; if they are to content, the value reduces 0.3; if they don't have written communication, the value reduces 0.3. So, the values of ω_2 to ω_5 are 0.5, 0.5, 0.8 and 0.2 respectively.

In other applications, the value of weights defined by alike method: Firstly, a basic value is set and then compare others with basic one to find differences and compute the weight for them respectively.

The demand for contact can be persistent so we need to consider its mutability. For example, for a given period T, it contains t weeks and the C_V between a friend and a centre user is $C_V f = \left\{ C_{V_1}, C_{V_2}, C_{V_3}, \cdots, C_{V_t} \right\}$. Let's also assume the T to be fifty-six days, and $t = 8$. Meanwhile, the C_V between f_1 and a centre user is $C_V f_1 = \{6, 7, 5, 4, 6, 8, 5, 6\}$, and the C_V between f_2 and a centre user is $C_V f_2 = \{0, 0, 0, 47, 0, 0, 0, 0\}$. Although the total of both are 47, the connection between f_1 and a centre user is continuous, whereas that between f_2 and a centre user may just be to discuss heated arguments or be involved in a discussion. So it is more likely that a centre user has a close relationship with f_1 instead of f_2. By virtue of this fact, we introduce a fluctuation value to further identify the contact frequency between friends and a centre user and with standard deviation (σ). It is computed as below:

$$\sigma = \sqrt{\frac{1}{t} \sum_{i=1}^{1} (C_{ti} - \bar{C}_V)} \quad (8)$$

Theoretically, the smaller the fluctuation of data the closer the relation between users. In other words, when we measure using standard deviation, there is an inverse relationship between its value and the strength of the tie and the derived value may not be absolute. For example, when there are two connections $C_V f_1 = \{0, 0, 0, 1, 2, 0, 0, 0\}$ and $C_V f_2 = \{0, 0, 0, 25, 0, 0, 0, 0\}$, it is obvious that the standard deviation of $C_V f_1$ is smaller but the tie strength of $C_V f_2$ is stronger.

From the above, we research the contact frequency between friends and a centre user not only considering the value of contact but also introducing the standard deviation of it. To achieve that, we define the formula of computing contact frequency as follows:

$$C_F = \frac{1}{\sigma} \cdot C_V \quad (9)$$

Using formula (9), we can find several friends of the top contact frequency and add them into a new circle denoted frequent contacts.

To sum up, the process of our algorithm is in the following three parts:

(1) We select in advance the primary attributes using information gain. Then by comparing the attributes of a centre user and friends and also calculating the np-value of each friend to assign friends into circles.
(2) For friends whose attributes are missing or withheld, we calculate the probability of them and the change with F value of circles to decide whether to assign them into circles.
(3) We calculate the value of C_F, which reflects contact frequency between each friend and a centre user to find the top friends and add them into a new circle denoted frequent contacts.

4. Experiment

We use the data-set of ego networks extracted from three major social network sites; Facebook, Google+and Twitter[4]. All of these ego networks provide not only personal information, but also structural information.

For Facebook, there are 10 ego networks, containing 193 circles and 4039 users. The data has attributes in 26 categories, including birthdays, hometown, colleagues, political affiliations, etc.

For Google+, there are 133 ego networks, containing 479 circles and 106,674 users. The data has attributes in 6 categories, including last name, gender, job titles, universities, institutions, and places lived.

For Twitter, there are 1,000 ego networks, containing 4,869 circles and 81,362 users. The data is collected from the set of hash tags and mentions used by each user.

In order to solve the problem of a missing users profile, we randomly hid the profile of users with 30, 60 and 100% to test.

Contact frequency is not included in these data-sets, so we collected the data by making some questionnaires on campus, named CFD. A total of 20 people participated in our research of selecting friends. Ten out of the twenty respondents were male and the other ten were female. Seventeen of the respondents were students and remaining 3 teachers. All respondents willingly volunteered to participate in our survey. Respondents had an average of 113 friends. The maximum number of friends was 302 and the minimum number of friends was 41. During the survey phase, apart from dividing friends into primary circles and buddy circle, the respondents were also asked to manually pick out 10 friends whom they considered to be frequent contacts in their circle. In total, there are 20 ego networks, containing 147 circles and 1154 users.

To evaluate our approach, we conducted some experiments here. Existing methods for community detection only use one of the users profile or network structure to identify a user's circle. The former does not address the problem of less or even no user profile data and the latter performs poorly when the network is tested with real world data. In this section, we compare ASF with the earlier mentioned algorithms DC-S and DC-M.

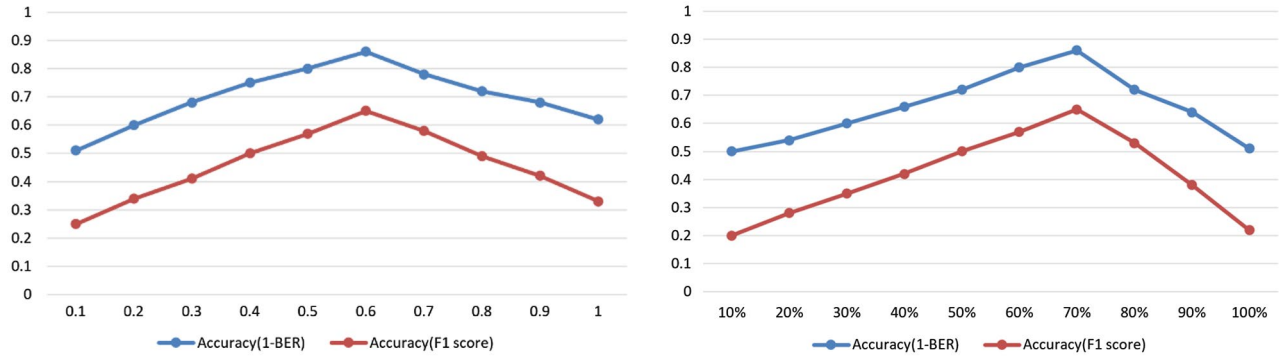


Figure 6. Performance of Changing α and β.

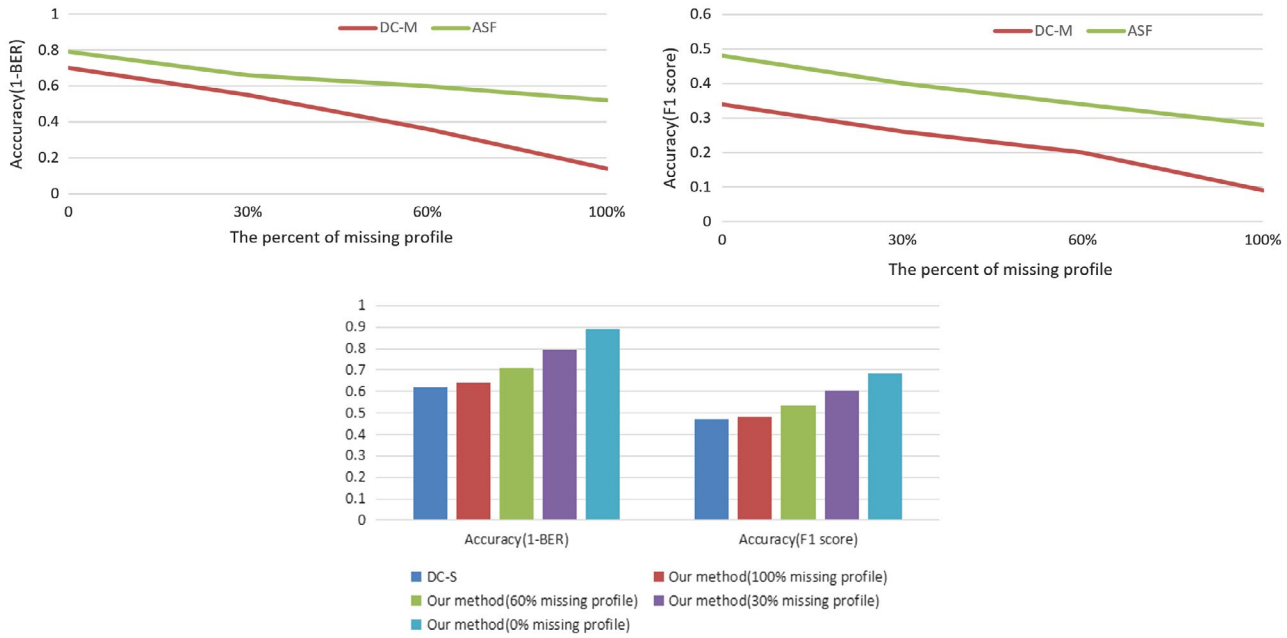


Figure 7. Performance of Facebook.

We consider two measurements to evaluate the methods; balance Error Rate (BER) [28] and F_1 score. The BER can be computed as below:

$$BER(C, C_\sigma) = \frac{1}{2}\left(\frac{|C - C_\sigma|}{|C|} + \frac{|C_\sigma - C|}{|C_\sigma|}\right) \quad (10)$$

Where C is the sum of all the predicted sets and C_δ is the sum of real data. This method assumes false positives and false negatives to have the same level of importance hence the average error rate of their trivial or random prediction is 0.5.

The F_1 score can be calculate as follows:

$$F_1(C, C_\sigma) = 2 \cdot \frac{precision(C, C_\sigma) \cdot recall(C, C_\sigma)}{precision(C, C_\sigma) + recall(C, C_\sigma)} \quad (11)$$

Where precision and recall are defined as:

$$precision(C, C_\sigma) = \frac{C \cap C_\sigma}{C}, \quad recall(C, C_\sigma) = \frac{C \cap C_\sigma}{C_\sigma} \quad (12)$$

The experiments were conducted on a desktop with 2.66 GHz CPU, 4 GB memory, 1 TB disk space and a Windows Operating System. All experimental results are the average values of more than five times of program running.

First, we tested repeatedly for deciding α and β. We tested on four datasets and observed the changing performance of different α and β. We discuss one fixed and another changed, and find the best value of α is rounding after the number of not primary attributes multiplied by 0.6, β is 70%. The result of general trends is shown in Fig 6, where we just give the best situation (fixed α is 0.6, fixed β is 70%).

The results of experiments on Facebook, Google+and Twitter are shown in Fig. 7, Fig. 8 and Fig. 9, respectively. The result of ASF is much better than DC-M. Moreover, as the percentage of missing profile increases, the accuracy of DC-M descends rapidly. Nevertheless, the rate of descent with ASF is slower. For DC-M, the reason for that is, because ASF is not totally dependent on the users' profile. And although the accuracy of DC-S is stable, we are better than it, especially when profile is increasing greatly. The reason for that is, because ASF considers not only the structure of ego network, but also the users' profile, so we can provide more accurate results.

However, these data-sets do not have the attribute of contact frequency, which cannot reflect the advantage of ASF. So, we also do experiments using collected data-sets, which have the attribute of contact frequency. The results of experiments on CFD are shown in Fig. 10.

We compare the results of ASF with DC-S and DC-M, where ASF is distinguished according to whether using contact

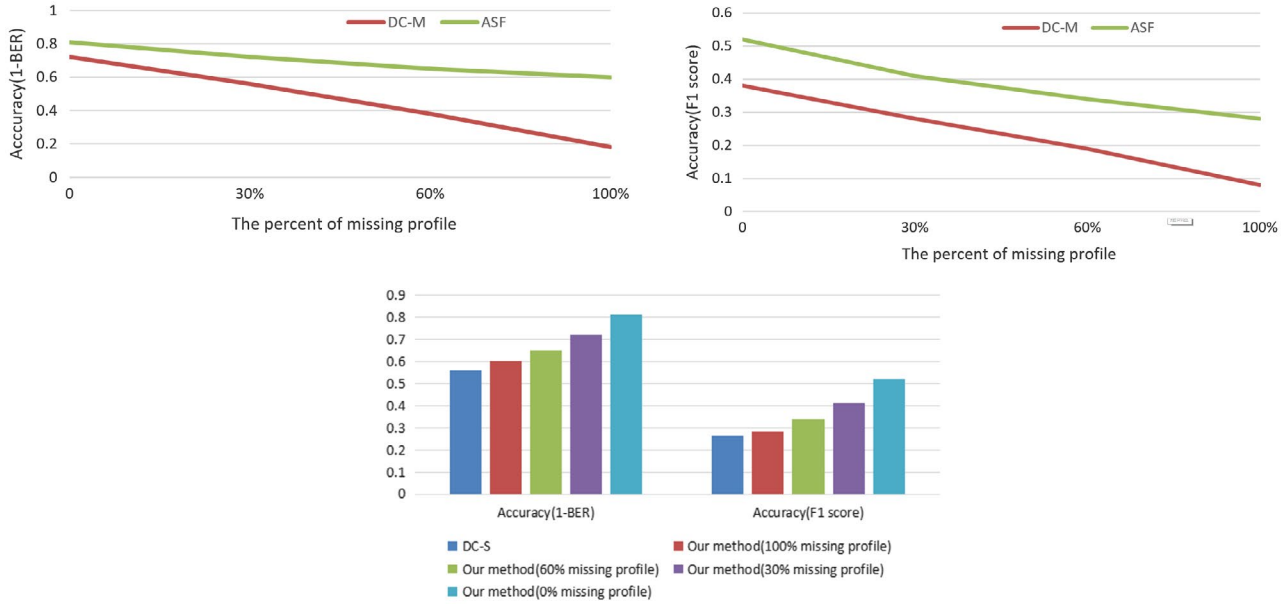


Figure 8. Performance of Google+.

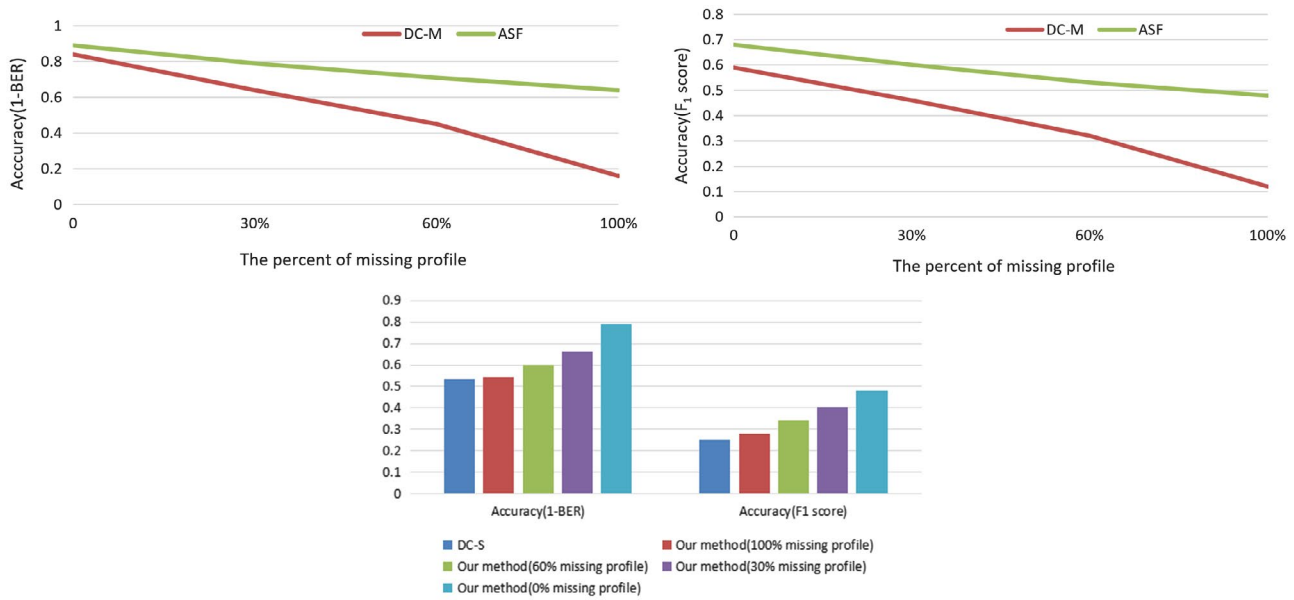


Figure 9. Performance of Twitter.

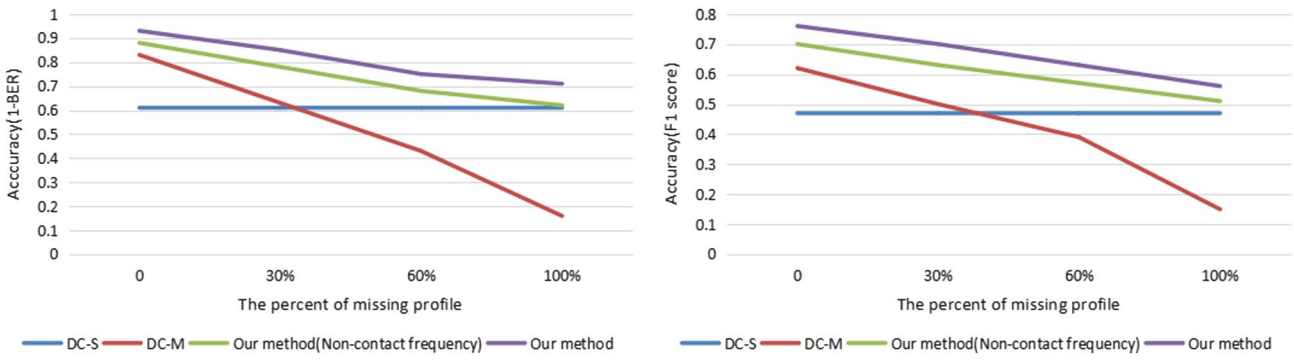


Figure 10. Performance of Collected Data.

frequency. When we do not use contact frequency, the results of each method is just like Fig. 7–9, but for some special friends, they have less similarities with a center user, but contact with a center user continually. Of course, they are close friends with the center user. However, they will not be added into circle by using profile and structure. ASF can find these friends and add

them into the circle of frequent contacts. As we can see, ASF is much better than DC-S and DC-M.

People change with time and so does their relationships with other people. In order to indicate advantage of ASF further, we observe results of different periods. In ASF, as changing of contacts among users, the result of frequent contacts changes as well. So, the accuracy of ASF is stable. However, DC-S and DC-M cannot reflect this change.

5. Conclusion

In this paper, we propose a method to detect circles in ego networks. The proposed method can identify the relation between a centre user and friends effectively. Our approach consists of three parts; the similarity of user attributes the features of a network structure and the contact frequency between a centre user and friends.

We first compare the similarity among attributes by means of a single attribute compare and multiple values of attributes compared. We then divide all friends by the similarity of the properties between each friend and a centre user. For friends with missing or incomplete attributes, we utilize the characteristics of ego network's structure to categorize them further thus by calculating the probability of whether one node should be added into a circle and the F value be changed if the node is added into a circle to decide if a node will be added into a circle or not.

Lastly, we analyse the frequency of contact for a centre user and friends, which also reflects interactions between them and also to find out which users continuously keep in touch with a centre user and then we place them in a specialized circle.

Disclosure Statement

No potential conflict of interest was reported by the authors.

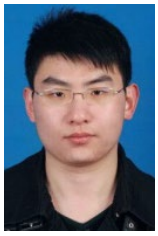
Funding

This work was supported in part by Special Public Sector Research Program of China (Grant no: GYHY201506080) and was also supported by PAPD.

Notes on Contributors

Jing Jia received a Bachelor's degree in Computer Science & Technology from Nanjing University of Information Science & Technology, China in 2014. Currently, she is a candidate for the degree of Software Engineering in Nanjing University of Information Science & Technology. Her research interest is social network and privacy preserving.

Tinghuai Ma is a professor in Computer Sciences at Nanjing University of Information Science & Technology, China. He received a Bachelor (HUST, China, 1997), Master's (HUST, China, 2000), Ph.D. (Chinese Academy of Science, 2003) and was Postdoctoral associate (AJOU University, 2004) from Nov. 2007 to Jul. 2008. He visited Chinese Meteorology Administration from Feb. 2009 to Aug. 2009. He was a visiting professor in Ubiquitous computing Lab, Kyung Hee University. His research interests are data mining, cloud computing, ubiquitous computing, privacy preserving, etc. He has published more than 100 journal/conference papers.



Fan Xing received a Bachelor's degree in Computer Science & Technology from Nanjing University of Information Science & Technology, China in 2014. Currently, he is a candidate for the degree of Software Engineering in Nanjing University of Information Science & Technology. His research interest is social network and privacy preserving.

William Farah graduated with honors in Computer Science from Koforidua Technical University in Ghana, a diploma in Information Technology from the University of Western Australia in Australia and a Master's degree in Computer Science and Technology from Nanjing University of Information Science and Technology in 2016 in China. He's currently teaching at Kunshan International School in Kunshan. His research interests include social networks, artificial intelligence, and computer vision.

Donghai Guan received a B.S. in College of Automation from Harbin Engineering University (HEU), Harbin, China in 2002. He received an M.S. degree in Computer Science from Kumoh National Institute of Technology (KIT), Gumi, South Korea in 2004 and received Ph.D. degree in Computer Science from Kyung Hee University, South Korea in 2009. From 2009, he was a Post-Doctoral Fellow at Computer Science Department, Kyung Hee University. His research interests are machine learning, pattern recognition, data mining, activity recognition, and trust modeling.

ORCID

Tinghuai Ma http://orcid.org/0000-0003-2320-1692

References

Arnaboldi V., Conti, M., and Passarella, A. (2012, September). Analysis of ego network structure in online social networks. International conference on social computing (SocialCom). IEEE, Amsterdam.

Cheng L., & Yan F. (2014, October). Research and implementation of social network service model. Intelligent Computation Technology and Automation (ICICTA), 7th International Conference on. IEEE, Changsha, China.

Coscia, M., Giannotti, G., & Pedreschi, P. (2011). A classification for community discovery methods in complex networks. *Statistical Analysis and Data Mining, 4*, 512–546.

Du N., Wu, B., Pei, X., Wang, B., & Xu, L. (2007, August). Community detection in large-scale social networks. Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis. ACM, San Jose, USA.

Fan W., Yeung K.H., & Fan W.J. (2015, February). Overlapping community structure detection in multi-online social networks. Intelligence in Next Generation Networks (ICIN), 18th International Conference on. IEEE, Paris, France.

Ferrara, E. (2012). Community structure discovery in facebook. *International Journal of Social Network Mining, 1*, 67–90.

Gao B., & Bettina B. (2013, August). Circles, posts and privacy in egocentric social networks: An exploratory visualization approach. IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Niagara Falls, Canada.

Hu, Y.M., & Yang, B. (2015). Enhanced link clustering with observations on ground truth to discover social circles. *Knowledge-Based Systems, 73*, 227–235.

Kim D., Jo, Y., Moon, I.-C., & Oh, A. (2010, April). Analysis of twitter lists as a potential source for discovering latent characteristics of users. ACM CHI workshop on microblogging, Atlanta.

Kossinets, G., & Duncan, J. (2006). Empirical analysis of an evolving social network. *Science, 311*, 88–90.

Liu, G., Yang, Q., Wang, H., Wu, S., & Wittie, M.P. (2015, September). *Uncovering the mystery of trust in an online social network*. Florence, Italy: IEEE CNS.

Lv, Y.H., & Ma, T.H., et al. (2016). An efficient and scalable density-based clustering algorithm for datasets with complex structures. *Neurocomputing, 171*, 9–22.

Ma, T.H., Zhang, Y.L., Cao, J., Shen, J., Tang, M., Tian, Y., Al-Dhelaan, A...Al-Rodhaan, M. (2015). KDVEM: A k-degree anonymity with vertex and edge modification algorithm. *Computing, 97*, 1165–1184.

Ma, T.H., & Wang, Y., Tang, M., & Al-Rodhaan (2016). LED: A fast overlapping communities detection algorithm based on structural clustering. *Neurocomputing, 207*, 488–500.

Ma T.H., Rong H., Ying C., Tian, Y., Al-Dhelaan, A., & Al-Rodhaan, M. (2016) Detect structural-connected communities based on BSCHEF in C-DBLP. *Concurrency and Computation: Practice and Experience,* 28, 311–330.

McAuley, J., & Leskovec, L. (2014). Discovering social circles in ego networks. *ACM Transactions on Knowledge Discovery from Data, 8*(1), 1–28.

Miao Q., Xing, T., Quan, Y., & Deng, K. (2014, December), Detecting circles on ego network based on structure. Computational Intelligence and Security (CIS), Tenth International Conference on. IEEE, Kunming, China

Mislove A., Viswanath, B., Gummadi, K.P., & Druschel, P. (2010, February). You are who you know: Inferring user profiles in online social networks. Proceedings of the third ACM international conference on Web search and data mining. ACM, New York City.

Newman M. (2004). Detecting community structure in networks.*The European Physical Journal B - Condensed Matter,* 38, 321–330.

Rong, H., Ma, T.H., Tang, M., & Cao, J. (2017). A novel subgraph K+ -isomorphism method in social network based on graph similarity detection. *Soft Computing,*. DOI:10.1007/s00500-017-2513-y

Wadhwa, P., & Bhatia, M.P.S. (2014). Community detection approaches in real world networks: A survey and classification. *International Journal of Virtual Communities and Social Networking, 6*, 35–51.

Xue, Y., Jiang, J., Zhao, B., & Ma, T. (2017). A self-adaptive artificial bee colony algorithm based on global best for global optimization. *Soft Computing,* 1–18.

Yoshida, T. (2013). Toward finding hidden communities based on user profile. *Journal of Intelligent Information Systems, 40*, 189–209.

Yuan, W.W., Guan, D., Lee, Y.-K., Lee, S., & Hur, S.-J. (2010). Improved trust-aware recommender system using small-wordless of trust networks. *Knowledge-Based Systems, 23*, 232–238.

Zhang, S.B. (2014, September). Influence of relationship strengths to network structures in social network. Communications and Information Technologies (ISCIT), 14th International Symposium on. IEEE, Incheon, South Korea.