K-Clique Community Detection in Social Networks Based on Formal Concept Analysis

Fei Hao, Geyong Min, Zheng Pei, Doo-Soon Park, *Member, IEEE*, and Laurence T. Yang

***Abstract*—With the advent of ubiquitous sensing and networking, future social networks turn into cyber-physical interactions, which are attached with associated social attributes. Therefore, social network analysis is advancing the interconnections among cyber, physical, and social spaces. Community detection is an important issue in social network analysis. Users in a social network usually have some social interactions with their friends in a community because of their common interests or similar profiles. In this paper, an efficient algorithm of k-clique community detection using formal concept analysis (FCA)—a typical computational intelligence technique, namely, FCA-based k-clique community detection algorithm, is proposed. First, a formal context is constructed from a given social network by a modified adjacency matrix. Second, we define a type of special concept named k-equiconcept, which has the same k-size of extent and intent in a formal concept lattice. Then, we prove that the k-clique detection problem is equivalent to finding the k-equiconcepts. Finally, the efficient algorithms for detecting the k-cliques and k-clique communities are devised by virtue of k-equiconcepts and k-intent concepts, respectively. Experimental results demonstrate that the proposed algorithm has a higher F-measure value and significantly reduces the computational cost compared with previous works. In addition, a correlation between k and the number of k-clique communities is investigated.**

***Index Terms*—k-clique, k-clique community, equiconcept, formal concept analysis (FCA), social networks.**

I. Introduction

A CYBER-PHYSICAL SYSTEM (CPS) is a system featuring a combination of computational and physical elements, all of which are capable of interacting, reflecting, and influencing each other. Furthermore, social systems are evolving with cyber systems and physical systems along with the popularity of online social networking [1]. A novel emerging computing paradigm cyber-physical-social system (CPSS), which converges the cyber, physical, and social spaces, is changing the way we see the world [2]. Online social networks, the main representation of the social spaces, are playing a critical role in shaping the behavior of users on the web. In social networks, users usually gather together and have a number of social interactions with each other in several communities due to their common interests and purposes. Therefore, community detection within social networks is a promising technique that provides an insight into the structural characteristics of the social networks and computational intelligence for social users in CPSSs. Community detection in networks aims to find groups of vertices within which connections are dense, but between which connections are sparser [3], [4]. In particular, the knowledge and computational intelligence of community structures can help us understand the behaviors and organization style of users in social networks [5], [6]. Two types of community detection methods are discussed in [7]: those that provide a partition of the network and those that provide a cover of the network. The main difference between these two techniques is that the former type does not allow communities to overlap, whereas the latter does. This paper aims at exploiting the second type of community detection methods with a focus on the k-clique community detection.

There has been some theoretical and empirical work on how the k-cliques and k-clique communities can be detected in social networks [8]–[14]. Adamcsek *et al.* [10] provided a faster CFinder to find the k-cliques. Kumpula *et al.* [11] proposed the sequential clique percolation algorithm to improve detection efficiency. However, these improved methods perform poorly on networks with the kind of pervasively overlapping community structure existing in many real-world social networks. Palla *et al.* [12] were first to define the k-clique community and extracted a set of k-clique communities with CFinder. Saito *et al.* [13] presented a new notion of a subnetwork called k-dense and proposed an efficient algorithm for extracting the k-dense communities. Duan *et al.* [14] solved the k-clique clustering in a dynamic social network. Tang *et al.* [15] aimed to reveal the similar structural and functional information of organic chemicals and proposed an approach for chemical structural retrieval based on formal concept analysis (FCA). However, there is no previous work on k-clique and k-clique community detection using FCA. In fact, FCA provides a more clear view to understand the network topology [16], [17]. Snael *et al.* [16] proposed a novel approach to overcoming some practical issues when dealing with analysis and visualization of large-scale social network data using FCA.

Manuscript received January 15, 2015; revised May 6, 2015; accepted May 11, 2015. Date of publication June 3, 2015; date of current version March 10, 2017. This work was supported by the Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Education under Grant NRF-2014R1A1A4A01007190. The work of G. Min was partly supported by the EU FP7 CLIMBER project under Grant PIRSES-GA-2012-318939. The work of Z. Pei was partly supported by the National Natural Science Foundation of China under Grant 61372187.

F. Hao and D.-S. Park are with the Department of Computer Software Engineering, Soonchunhyang University, Asan 336-745, Korea (e-mail: feehao@gmail.com; parkds@sch.ac.kr).

G. Min is with the College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter EX4 4QF, U.K. (e-mail: g.min@exeter.ac.uk).

Z. Pei is with the Center for Radio Administration and Technology Development, Xihua University, Chengdu 610054, China (e-mail: pqyz@263.net).

L. T. Yang is with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China, and also with the Department of Computer Science, St. Francis Xavier University, Antigonish, NS B2G 2W5, Canada (e-mail: ltyang@gmail.com).

Digital Object Identifier 10.1109/JSYST.2015.2433294

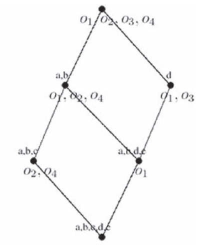
With the help of FCA's powerful analysis ability on network topology, this paper studies the FCA-based k-cliques and k-clique community detection. To the best of our knowledge, this work is the first to study the k-cliques and k-clique community detection problems using FCA. First, the transformation from a social network to a formal context, which is an input of the FCA method, is studied; then, a formal concept lattice is obtained. Then, we prove that the problem of k-clique detection is equivalent to the problem of finding the k-equiconcepts. Finally, the efficient algorithms to detect the k-cliques and k-clique communities are devised with the help of k-equiconcepts and k-intent concepts, respectively. The major contributions of this paper are as follows.

1) **Formal Context Construction**: We provide a solution for formal context construction of a social network by using a modified adjacency matrix. First, each vertex in a social network is regarded as both objects and attributes. Second, a binary relation between the objects and attributes is defined according to the social interaction. Third, a formal context is generated from the social network by the modified adjacency matrix.
2) **FCA-Based k-Clique Detection**: An FCA-based k-clique detection approach is proposed. First, we prove that the k-clique detection problem is equivalent to finding the k-equiconcepts in the concept lattice of a social network. In addition, an interesting conclusion that extra k-cliques can be derived from the detected k-equiconcepts is discovered. Then, an algorithm of detecting k-cliques with FCA is presented.
3) **FCA-Based k-Clique Community Detection**: Following k-clique detection, an FCA-based k-clique community detection approach is devised. We prove that the k-clique community detection problem is equivalent to finding the k-intent equiconcepts in the concept lattice of a social network. Then, we analyze the formation principle of k-clique communities and find that each k-clique community can be formed based on the skeleton k-cliques (k-intent concepts). Finally, an efficient algorithm of FCA-based k-clique community detection is presented.
4) **Evaluations**: The proposed approach is evaluated using four data sets. First, we evaluate various approaches on how well they can find the k-clique community structure from a social network. Second, in terms of efficiency, the proposed approach can detect the k-cliques and k-clique communities quickly compared with other existing approaches. Finally, the correlation between k and the number of k-clique communities is investigated thoroughly.

The rest of this paper is organized as follows. Section II presents the preliminaries about k-clique, k-clique community, and FCA. The problem definition of k-clique community detection is described in Section III. Section IV presents the FCA-based k-cliques and k-clique community detection from a social network, respectively. Experimental results are reported in Section V. Finally, Section VI concludes this paper.

$U \backslash A$	a	b	c	d	e
o_1	×	×		×	×
o_2	×	×	×		
o_3					×
o_4	×	×	×		

(a)



(b)

Fig. 1. Example of formal context and its corresponding concept lattice. (a) Formal context. (b) Concept lattice.

II. Preliminaries

This section presents the definitions of k-clique, k-clique community, as well as the theory of FCA theory.

A. k-Clique and k-Clique Community

Definition 1 (Clique): Let $G = (V, E)$ be an undirected graph. A clique in G is a subset $S \subset V$ such that for any two vertices $v_i, v_j \in S$, there exists an edge $(v_i, v_j) \in E$.

Definition 2 (k-Clique): Let $G = (V, E)$ be an undirected graph. A k-clique in G is a subset $S \subset V$ and $|S| = k$ such that for any two vertices $v_i, v_j \in S$, there exists an edge $(v_i, v_j) \in E$.

Definition 3 (k-Clique Community): A k-clique community [12] is defined as the union of all k-cliques (i.e., complete subgraphs of size k) that can be reached from one or other through a series of adjacent k-cliques (where adjacency means sharing k-1 vertices).

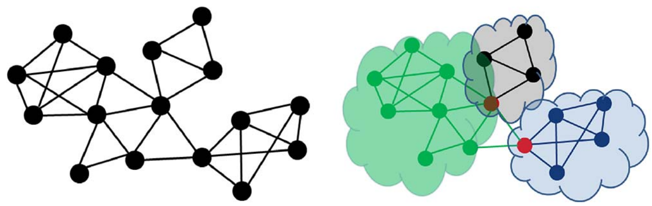
B. FCA

FCA is a typical computational intelligence technique for data analysis. FCA defines formal concept to represent the relationships between objects and attributes in a domain. The objects and attributes are grouped into concepts, and then, a conceptual hierarchy of the concepts can be constructed.

Definition 4 [21] (Formal Context): A formal context is organized as a triple $K = (U, A, I)$, where $U = \{x_1, x_2, \ldots, x_n\}$ is the set of objects, $A = \{a_1, a_2, \ldots, a_m\}$ is the set of attributes, and I is the binary relation between U and A. $I \subseteq U \otimes A$, $(x, a) \in I$ denotes that object x has the attribute a, and $(x, a) \notin I$ denotes that object x does not have the attribute a, where $x \in U, a \in A$.

Remark 1: Let "1" denote $(x, a) \in I$ and "0" denote $(x, a) \notin I$. Then, this formal context can be viewed as an information system with only "0" or "1." In many literatures, the cross table is often used for describing the formal context, i.e., if $(x, a) \in I$, the binary relation I is represented as "×"; otherwise, the blanks are given for $(x, a) \notin I$.

Example 1: Fig. 1(a) shows a formal context. The set of objects is $U = \{o_1, o_2, o_3, o_4\}$, the set of attributes is $A = \{a, b, c, d, e\}$, and in which "×" denotes that there exists the binary relation between U and A. For example, the object "o_2" has the attributes "a," "b," and "c."



(a) (b)

Fig. 2. Toy example of k-clique community detection. (a) Social network G. (b) 3-clique communities.

Definition 5 [22]: For a formal context $K = (U, A, I)$, the operators $\uparrow$ and $\downarrow$ on $X \subseteq U$ and $B \subseteq A$ are, respectively, defined as

$$X^{\uparrow} = \{a \in A| \quad \forall x \in X, (x, a) \in I\} \tag{1}$$

$$B^{\downarrow} = \{x \in U| \quad \forall a \in B, (x, a) \in I\} \tag{2}$$

$\forall x \in U$, *let* $\{x\}^{\uparrow} = x^{\uparrow}$, and $\forall a \in A$, let $\{a\}^{\downarrow} \in a^{\downarrow}$.

Definition 6 [21] (Concept): For a formal context $K = (U, A, I)$, if a pair (X, B) satisfies $X^{\uparrow} = B$ and $B^{\downarrow} = X$, then the pair (X, B) is a concept, where X is called the extent of the concept, and B is called the intent of the concept. Let $C(K)$ denote the set of all concepts with respect to formal context K.

Definition 7 [22]: Let $C(K)$ denote the set of all formal concepts of the formal context $K = (U, A, I)$. If $(X_1, B_1), (X_2, B_2) \in C(K)$, then let

$$(X_1, B_1) \leq (X_2, B_2) \Leftrightarrow X_1 \subseteq X_2 (\Leftrightarrow B_1 \supseteq B_2) \tag{3}$$

then "$\leq$" is a partial relation of $C(K)$.

Definition 8 (Concept Lattice): A concept lattice $L = (C(K), \leq)$ can be obtained by all formal concepts $C(K)$ of a context K with the partial order $\leq$. Its graphical representation is a Hasse diagram. Fig. 1(b) illustrates the concept lattice for the context of Fig. 1(a). Each circle denotes a concept. The upper labels and lower labels of the circles represent intents and extents of the concepts, respectively.

III. Problems Definition

In this paper, we mainly investigate the k-clique community detection problem using FCA theory in a social network. The formulism of k-clique community detection problem is described as follows.

Problem Statement (k-Clique Community Detection): Give a social network $G = (V, E)$, where the node set V includes the entities in the social network, and the edge set $E = \{(u, v)|u, v \in V\}$ denotes the relationship between entities. Once the parameter k is given, the k-clique community detection problem is to detect all k-clique communities from G.

To better understand the problem addressed in this paper, a toy example of 3-clique community detection is given in Fig. 2. Obviously, Fig. 2(a) is the topology of a social network G. After k-clique community detection, there are k (here, $k = 3$) separated communities that appear in G, as shown in Fig. 2(b).

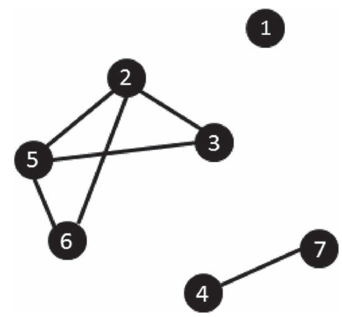


Fig. 3. Social network g.

IV. FCA-Based k-Clique Community Detection

This section provides a novel detection approach of k-clique communities using FCA theory. To elaborate our approach more clearly, we address and provide the solutions for the following issues: 1) construct a formal context from a social network G; 2) study the relation between the concept lattice and k-clique as well as k-clique detection; and 3) present an algorithm for detecting the k-clique communities.

A. Formal Context Construction

A social network G can be modeled as a set of subjects, in which some of them have some relationships with others. This can be formalized as a classical mathematical relationship visualized as an undirected graph. In this paper, we adopt the modified adjacency matrix of G as a formal context of G, namely, $FC(G) = (V, V, I)$, in which I is the binary relationship between two vertices.

A modified adjacency matrix is defined as follows.

Definition 9 (Modified Adjacency Matrix): Let G be a graph with n vertices that are assumed to be ordered from v_1 to v_n. The $n \times n$ matrix A' is called a modified adjacency matrix, in which

$$A' = \begin{cases} a_{ij} = 1, & \text{if there exists an edge from } v_i \text{ to } v_j \text{ and } i \neq j \\ a_{ij} = 1, & \text{if } i = j \\ a_{ij} = 0, & \text{otherwise.} \end{cases} \tag{4}$$

Therefore, $FC(G)$ is equivalent to the modified adjacency matrix of G, i.e., $FC(G) \equiv A'$. According to the properties of A', $FC(G)$ also has following properties.

Property 1:

1) $FC(G)$ is symmetric.
2) One difference from the adjacency matrix is that all the diagonal elements are "1".

Example 2: Fig. 3 presents a social network g with vertices indicating users and edges indicating the relationships between users, and a formal context of g is constructed in Table I according to the definition of the modified adjacency matrix.

TABLE I
FORMAL CONTEXT OF g

User	u_1	u_2	u_3	u_4	u_5	u_6	u_7
u_1	1	0	0	0	0	0	0
u_2	0	1	1	0	1	1	0
u_3	0	1	1	0	1	0	0
u_4	0	0	0	1	0	0	1
u_5	0	1	1	0	1	1	0
u_6	0	1	0	0	1	1	0
u_7	0	0	0	1	0	0	1

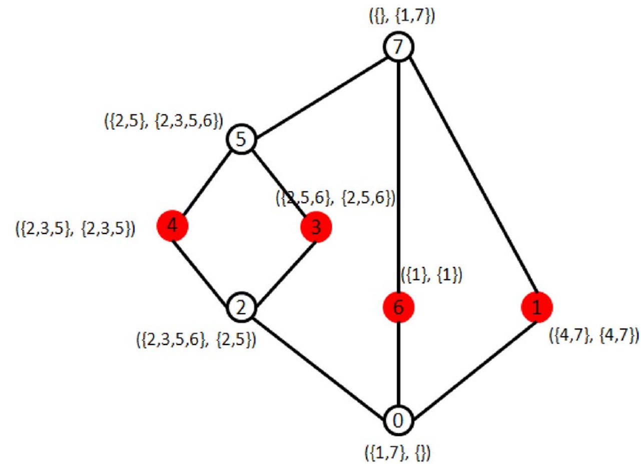


Fig. 4. Concept lattice of social network g (the "*red*" nodes denote the equiconcepts).

B. k-Clique Detection

This section introduces the concepts of equiconcept and k-equiconcept and then provides several interesting theorems and properties about k-clique detection with k-equiconcepts.

Definition 10 (Equiconcept): For a formal context $K = (U, A, I)$, if a pair (X, B) satisfies $X^\uparrow = B$, $B^\downarrow = X$ and $X = B$, then the pair (X, B) is an equiconcept, where X is called the extent of the equiconcept, and B is called the intent of the equiconcept. Moreover, let $EC(K)$ be the set of all equiconcepts with respect to the formal context K.

Definition 11 (k-Equiconcept): For a formal context $K = (U, A, I)$, if a pair (X, B) satisfies $X^\uparrow = B$, $B^\downarrow = X$, $X = B$, and $|X| = |B| = k$, then the pair (X, B) is a k-equiconcept, where X is called the extent of the k-equiconcept, and B is called the intent of the k-equiconcept. Moreover, let $KEC(K)$ be the set of all k-equiconcepts with respect to the formal context K.

Theorem 1: Given a social network G, the k-clique detection problem is equivalent to finding $KEC(FC(G))$.

Proof: Let $P_{k-clique}$ be the k-clique detection problem and $P_{KEC(FC(G))}$ be the problem of finding $KEC(FC(G))$. The above theorem is mathematically described as: $P_{k-clique} \equiv KEC(FC(G))$. Hence, we need prove it toward two directions: 1) $P_{k-clique} \Rightarrow P_{KEC(FC(G))}$ and 2) $P_{k-clique} \Leftarrow P_{KEC(FC(G))}$.

1) ($P_{k-clique} \Rightarrow P_{KEC(FC(G))}$): Given a social network G, a k-clique contains vertices $v_1, v_2, \ldots, v_k$, for any two vertices v_i, v_j, there exists an edge between them. Since a k-clique is a subgraph, we can easily construct the formal context using a modified adjacency matrix. Obviously, the formal context of a k-clique is a matrix of 1's. We can extract a special such kind of concept $(\{v_1, v_2, \ldots, v_k\}, \{v_1, v_2, \ldots, v_k\})$ from this formal context, which satisfies $X = B$, X is the extent of this special concept, and B is the intent of this special concept. This special concept is actually a k-equiconcept; hence, $P_{k-clique} \Rightarrow P_{KEC(FC(G))}$.
2) ($P_{k-clique} \Leftarrow P_{KEC(FC(G))}$): Due to Definition 11, we know that all extracted k-equiconcepts $KEC(FC(G)) = \{(X_i, B_i)|i = 1, 2, \ldots, r\}$ and r is the number of k-equiconcepts with respect to the formal context $FC(G)$. Here, (X_i, B_i) is the ith k-equiconcept, X_i is the extent of the ith k-equiconcept, and B_i is the intent of the ith k-equiconcept and $|X_i| = |B_i| = k$. In a formal context of social network G, both X_i and B_i consist of a subset of vertices, i.e., $X_i = \{v_1, v_2, \ldots, v_k\}$. Since X_i and B_i are one of k-equiconcepts, it means that the vertices in X_i are connected with the vertices in B_i. Hence, we can obtain a subgraph (k-clique) based on the association between X_i and B_i. For $i = 1, 2, \ldots, r$, we can obtain all k-cliques. Consequently, $P_{k-clique} \Leftarrow P_{KEC(FC(G))}$.

Since we have already proved that $P_{k-clique} \Rightarrow P_{KEC(FC(G))}$ and $P_{k-clique} \Leftarrow P_{KEC(FC(G))}$, $P_{k-clique} \equiv KEC(FC(G))$ holds. ■

Lemma 1: Let (X, B) be a k-equiconcept, the number of derived $(k-1)$-cliques from (X, B) is equal to C_k^{k-1}.

Proof: Since (X, B) is a k-equiconcept, all the vertices in X are connected with the vertices in B. Let $X' \subseteq X$ or $B' \subseteq B$, and $|X'| = |B'| = k - 1$. This problem is converted into a combination problem about how many combination cases for extracting X'. Hence, there are C_k^{k-1} cases for X', i.e., we can derive the $(k-1)$-cliques from (X, B). ■

Example 3: Let us continue Example 2, we can build the concept lattice of the social network g according to Definition 8, which is denoted as $L(C(FC(g)), \leq)$.

The visualization of $L(C(FC(g)), \leq)$ is shown in Fig. 4, from which we can easily find the four equiconcepts marked in red, i.e., ({1}, {1}), ({4, 7},{4, 7}), ({2, 5, 6}, {2, 5, 6}) and ({2, 3, 5}, {2, 3, 5}). In fact, in the social network g, these equiconcepts correspond to 1-clique, 2-clique, 3-clique, and 3-clique, respectively. Moreover, we can derive more 2-cliques from ({2, 5, 6}, {2, 5, 6}) and ({2, 3, 5}, {2, 3, 5}), such as ({2, 3}, {2, 3}), ({2, 5},{2, 5}), ({2, 6},{2, 6}), ({3, 5}, {3, 5}), and ({5, 6}, {5, 6}). However, they are not concepts; they do not appear in the concept lattice of social network g.

Theorem 2: Given a social network G, all k-clique detection is composed of the following parts: 1) basic cliques are generated from the k-equiconcepts; 2) remaining cliques are derived from the $(k+1)$-equiconcepts, $(k+2)$-equiconcepts,..., M-equiconcepts. $(M > k)$. M is the number of maximum extent or intent of maximum equiconcepts.

Based on the above theorem of all k-clique detection, the working process of Algorithm 1 is described as follows.

Algorithm 1 FCA-Based k-Clique Detection Algorithm

Input:
$G = (V, E)$;
Parameter k;
Output:
Set of k-cliques Γ
1: Initialize $\Gamma = \emptyset$
2: ***begin***
3: Construct a formal context $FC(G)$ by Definition 4
4: Build a concept lattice $C(FC(G))$ by invoking *CLBuilder*
5: ***end***
6: ***for*** each concept $(X, B) \in C(FC(G))$
7: ***begin***
8: ***if*** $X = B$ and $|X| = |B| = k$
9: $\Gamma \leftarrow \Gamma \cup (X, B)$
10: ***end***
11: ***if*** $X = B$ and $|X| = |B| > k$
12: ***for*** $i = k + 1$ ***to*** M ***do***
13: ***begin***
14: $\Gamma \leftarrow \Gamma \cup Derived((X^i, B^i))$
15: ***end***

Algorithm 2 CLBuilder

Input:
A formal context K;
Output:
Set of concepts *conceptset*
1: Initialize $conceptset \leftarrow \emptyset$
2: ***begin***
3: $conceptset \leftarrow \mathbf{BasicConcept}(\mathbf{C})$;
4: **AddConcept**(C);
5: Enter(queue,conceptset);
6: ***while*** queue $\neq \emptyset$ ***do***
7: ***begin***
8: $(X, X^{\uparrow}) \leftarrow$ queue.concept;
9: $SubNodes \leftarrow \mathbf{FindSubNodes}\ (X, X^{\uparrow})$
10: ***if*** SubNodes $\neq \emptyset$ ***then***
11: ***for*** $(Y, Y^{\uparrow}) \in$ SubNodes ***do***
12: $(X, X^{\uparrow})$.Edge $\leftarrow (Y, Y^{\uparrow})$
13: ***if*** SubNodes $= \emptyset$ ***then***
14: $(X, X^{\uparrow})$.Edge $\leftarrow (\emptyset, D)$
15: ***end***
16: ***end***

Algorithm 3 BasicConcept(C)

1: Initialize $conceptset \leftarrow \emptyset$
2: ***begin***
3: ***for*** i $= 1$ ***to*** $|V|$ ***do***
4: ***for*** j $= 1$ ***to*** $|V|$ ***do***
5: ***if*** $(c_{ij}^{\downarrow}, c_{ij})$ is not in *conceptset* ***then***
6: $conceptset \leftarrow conceptset \bigcup (c_{ij}^{\downarrow}, c_{ij})$
7: ***return*** *conceptset*
8: ***end***

Algorithm 4 AddConcept(C)

1: $conceptset' \leftarrow conceptset$
2: $conceptset'' \leftarrow \emptyset$
3: ***do***
4: ***begin***
5: ***for*** $(X_1, Y_1), (X_2, Y_2)$ in $conceptset'$ ***do***
6: ***begin***
7: $Y \leftarrow (Y_1 \bigcap Y_2)$
8: if $(Y^{\downarrow}, Y)$ is not in *conceptset* ***then***
9: ***begin***
10: $conceptset \leftarrow conceptset \bigcup (Y^{\downarrow}, Y)$
11: $conceptset'' \leftarrow conceptset'' \bigcup (Y^{\downarrow}, Y)$
12: ***end***
13: ***end***
14: $conceptset' \leftarrow concept''$
15: $conceptset'' \leftarrow \emptyset$
16: ***end***
17: ***until*** $conceptset' \leftarrow \emptyset$
18: ***end***

First, a social network G and parameter k are the inputs of the whole algorithm; then, we initialize a set of k-cliques with Γ (Line 1). After the initialization of algorithm, it goes into the formal context construction and concept lattice generation codes part (Lines 2–5). Lines 6–10 insert the detected k-equiconcepts (X, B) into Γ. The remaining set of k-cliques is derived from other high-order equiconcepts and is inserted into Γ (Lines 11–15). In the **CLBuilder** algorithm, we first initilize the conceptset as an empty set (Line 1). Line 3 invokes the algorithm **BasicConcept(C)** to obtain the basic concepts. Then, **AddConcepts** is to obtain the extensive concepts (Line 4). We store all the obtained concepts with a first-in–first-out queue data structure (Line 5). Lines 6–16 deal with constructing a concept lattice iteratively. Note that the algorithm of **FindSubNodes** has been already presented in [20].

C. Time Complexity Analysis

This section discusses the time complexity of building the formal concepts lattice. In the proposed formal context, the number of objects is denoted with $|V|$, and the number of attributes is also denoted with $|V|$. L is the number of all concepts. L_1 is the number of the basic concepts, and L_2 is the number of the added concepts. The time complexity analysis is given as follows.

1) **When a formal context is constructed, there exists a matrix operation**: The time complexity is $|V|^3$.
2) **The time complexity of obtaining basic concepts**: The time complexity of obtaining a basic concept is $|V|^2$, and the number of basic concepts is L_1. Therefore, the time complexity of obtaining all basic concepts is $|V|^2 \times L_1$.
3) **The time complexity of obtaining added concepts**: An added concept is regarded as resulting concept from the intersection of two basic concepts. Hence, the time complexity of obtaining an added concept is $r \times C_{|V|}^2$, where r is the number of iterations. Because of L_2 size

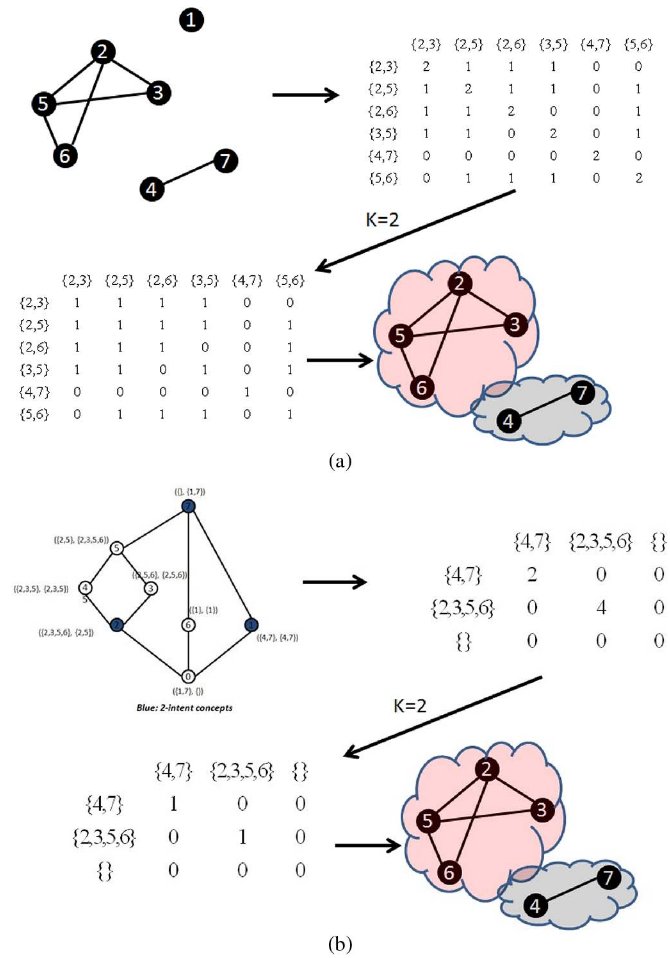


Fig. 5. Simple illustration of the extraction of the 2-clique communities. (a) The approach based on a clique–clique overlap matrix. (b) FCA-based approach.

of the added concepts, the time complexity of obtaining the added concepts is $r \times C^2_{|V|} \leq r \times |V|^2 \times |L_2|$.

In summary, the time complexity of the algorithm is $\Theta(|V|^3 + |V|^2(L_1 + L_2))$. As we know, $L = L_1 + L_2$. Therefore, the time complexity is $\Theta(|V|^3 + |V|^2 L)$.

D. FCA-Based k-Clique Community Detection

Here, we study how to detect the k-clique communities based on the detection results of k-cliques in the previous section. We first recall the existing approach for detecting k-clique communities and then propose our detection algorithm based on FCA. The k-clique communities for a given value of k are equivalent to such connected clique components in which the neighboring cliques are linked to each other by at least $k-1$ common vertices.

A simple illustration of the above analysis is shown in Fig. 5(a).

As shown in Fig. 5(a), 2-cliques are detected from the original social network G; then, a clique–clique overlap matrix is constructed. Then, a merged clique–clique matrix is generated. The elements "1" in this matrix indicate the connection relation between two separated 2-cliques. Finally, two 2-clique communities {2, 3, 5, 6} and {4, 7} are obtained.

One advantage of this method is that the clique–clique overlap matrix encodes all information necessary to obtain the communities for any value of k; therefore, once the clique–clique overlap matrix is constructed, the k-clique communities for all possible values of k can be obtained very quickly. However, the scalability of this method is very low. From the FCA point of view, we can devise an efficient algorithm to discover all of the k-clique communities.

Before we present the FCA-based k-clique community detection algorithm, an important definition is given as follows.

Definition 12 (k-Intent Concept): For a formal context $K = (U, A, I)$, if a pair (X, B) satisfies $X^{\uparrow} = B$, $B^{\downarrow} = X$, and $|B| = k$, then the pair (X, B) is a k-intent concept, where X is called the extent of the k-intent concept, and B is called the intent of k-intent concept. Moreover, let $KIC(K)$ denote the set of all k-intent concepts with respect to the formal context K.

Theorem 3: The problem of k-clique community detection is equivalent to finding the k-intent concepts and the extents of each k-intent concepts just share at least $k-1$ vertices.

Proof: As for a k-clique community, it is generated by k-cliques that are the skeletons of the k-clique community. Therefore, the skeleton of the k-clique community is regarded as the intent of a certain concept. To guarantee the separation of each k-clique community, a constraint of extents of each k-intent concepts only sharing at least $k-1$ vertices are given to divide the k-clique communities each other. ■

Let us continue to analyze Fig. 5(a); the intent of 2-clique community {2, 3, 5, 6} is {2, 5}. In other words, this community is generated based on the skeleton 2-clique {2, 5} with its extent {2, 3, 5, 6}.

Based on the proposed theorem, the FCA-based k-clique community detection algorithm works as follows.

- ***Step 1*** Given a social network G, generate a formal context $FC(G)$.
- ***Step 2*** Build a concept lattice about $FC(G)$: $C(FC(G))$.
- ***Step 3*** Extract all the k-intent concepts with respect to the formal context $FC(G)$: $KIC(FC(G))$.
- ***Step 4*** Construct the extent–extent overlap matrix (extent refers to B in the k-intent concept).
- ***Step 5*** The k-clique communities for a given value of k are equivalent to such extent components in the k-intent concepts in which the neighboring cliques are linked to each other by at least $k-1$ common vertices. These components can be found by erasing every off-diagonal entry smaller than $k-1$ and every diagonal element smaller than k in the matrix, replacing the remaining elements by one and then carrying out a component analysis of this matrix. The resulting separate components are equivalent to the different k-clique communities.

We provide the detailed detection procedure by a simple example illustration. Let us continue the same example in Fig. 5(a). As shown in Fig. 5(b), we first build a concept lattice of social network G. Then, we extract three 2-intent concepts (*blue nodes*) ({4, 7},{4, 7}), ({2, 3, 5, 6}, {2, 5}), and ({ },{1, 7}). After that, the extent–extent overlap matrix is generated. With a constraint, the extent–extent overlap matrix is modified as a “0-1” matrix. Eventually, two 2-clique communities are detected.

Comparing Fig. 5(b) with Fig. 5(a), the advantages of the proposed algorithm are concluded here: The proposed algorithm can significantly reduce the dimensions of the overlap matrix and the computational cost for overlapping elements in the matrix. Because there is no need to select all k-cliques, we just need extract the extent of k-intent concepts. Hence, the execution steps are significantly reduced.

Based on the above theorem of all k-clique detection, we present the detection algorithm, as shown in Algorithm 5.

Algorithm 5 FCA-Based k-Clique Community Detection Algorithm

Input:
$G = (V, E)$;
Parameter k;
Output:
Set of k-clique communities Ω
1: Initialize $\Omega = \emptyset$, $\Upsilon = \emptyset$
2: ***begin***
3: Construct a formal context $FC(G)$ by Definition 4
4: Build a concept lattice $C(FC(G))$ by invoking *CLBuilder*
5: ***end***
6: ***for*** each concept $(X, B) \in C(FC(G))$
7: ***begin***
8: ***if*** $|B| = k$
9: $\Upsilon \leftarrow \Upsilon \cup (X, B)$
10: ***end***
11: ***for*** each concept $(X, B) \in \Upsilon$
12: ***begin***
13: Construct the extent–extent overlapping matrix H with X
14: ***end***
15: ***if*** $(H_{ij} > k - 1)$
16: ***begin***
17: $\Omega \leftarrow \Omega \cup ((X^i) \cup X^j)$
18: ***end***

The working procedure of Algorithm 5 is described as follows: First, a social network G and parameter k are the inputs of the whole algorithm; then, we initialize a set of k-clique communities with Ω and a set of k-intent concepts with Υ (Line 1). After initializing the algorithm, it goes into the formal context construction and concept lattice generation codes part (Lines 2–5). Lines 6–10 insert the detected k-intent-concepts (X, B) into Υ. Then, we construct the extent–extent overlapping matrix H with X (Lines 11–14). While $H_{ij} > k - 1$, we just unify the extents together and store them into Ω (Lines 15–18).

V. Experiments

Here, we conducted experiments on four real-life networks to evaluate the proposed approach. The goal of the experiments was to investigate whether the proposed approach is efficient for detecting the k-cliques and k-clique communities.

TABLE II
Statistics of Four Data Sets in Experiments

Dataset	Nodes	Edges	Average Degree
Zachary's Karate Club	34	78	2.29
Dolphin Social Network	62	159	5.2
Jazz Musicians Network	198	5484	2.38
Yeast Protein Interaction	1486	4406	2.4

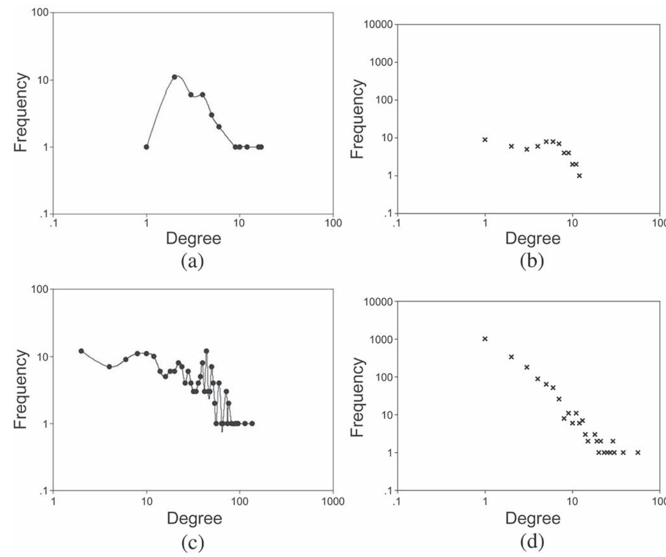


Fig. 6. Degree distributions (log–log scale) of four data sets. (a) Karate data set. (b) Dolphin data set. (c) Jazz data set. (d) Yeast data set.

A. Experiment Setup

In this paper, four data sets of social networks are adopted to evaluate the proposed approach. Some critical statistics of the data sets are shown in Table II. Data set I is a classical social network of friendships between 34 members of a karate club at a United States university in the 1970s.[1] Data set II is a small-size data set on the social network of frequent associations between 62 dolphins in a community living off Doubtful sound, New Zealand. Data set III is obtained from The *Red Hot Jazz Archive* digital database.[2] It is a network of Jazz musicians. Data set IV is a relatively large data set on yeast protein interactions between proteins, in which the 1486 nodes indicate the protein, and the 4406 edges indicate the interactions between proteins.[3]

Fig. 6 presents the degree distributions of four data sets, respectively. Obviously, they follow the power-law distribution in general.

B. Experimental Results

Our experiments were run on a 2.83-GHz quad core machine with 2-G memory. The experimental results are compared with the existing works: CPM [10], GN [18], and CDPM [19], respectively, to evaluate the effectiveness and efficiency of k-clique detection and k-clique community detection.

[1] http://www-personal.umich.edu/mejn/netdata/
[2] http://www.redhotjazz.com
[3] http://depts.washington.edu/sfields/yp_interactions/index.html

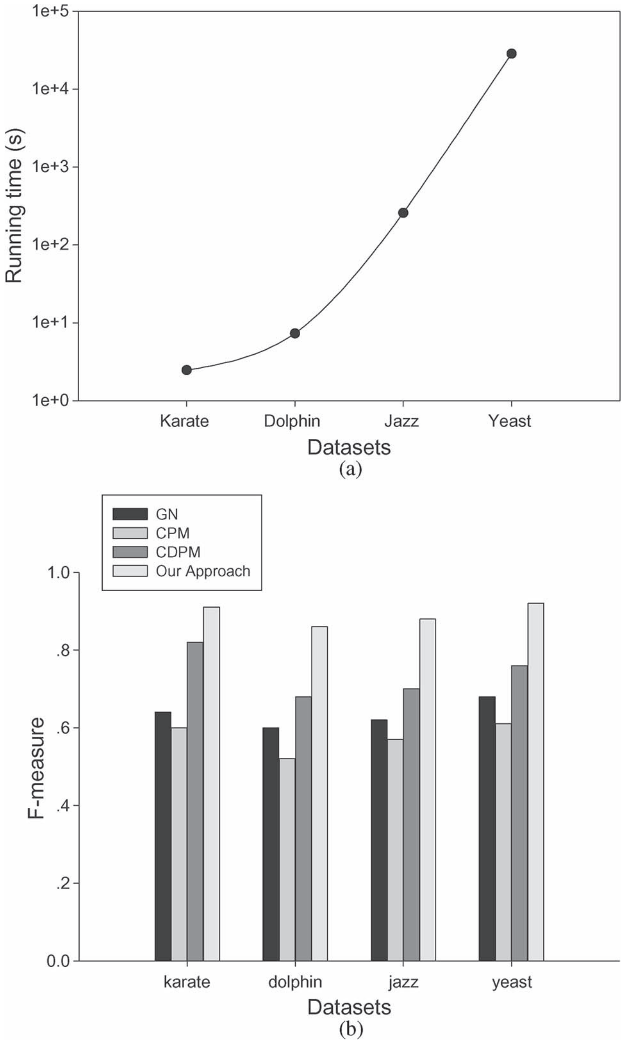


Fig. 7. Efficiency and effectiveness evaluation for four data sets. (a) The construction time of formal contexts. (b) F-*measure* results for data sets.

1) Construction Time of Formal Context: As the input data format of a social network is an undirected graph that contains information on any two vertices and its edges between them, we have to transform it into a formal context using a modified adjacency matrix. The construction time of formal contexts for four data sets is shown in Fig. 7(a). As shown in the figure, the construction time of formal context is dramatically increasing as the scale of social networks increases. Note that data set *Yeast* costs lots of time for formal context construction compared with other data sets due to its large-scale property.

2) Effectiveness Comparison Results: We run all algorithms on four data sets. The F-*measure* is used to measure how well each algorithm can find the k-clique community structure from a social network. F-*measure* is calculated as follows:

$$F - measure = \frac{2 \times recall \times precision}{recall + precision} \quad (5)$$

where *recall* denotes the fraction of vertex pairs belonging to the same k-clique community, which are also in the same cluster, and *precision* is the fraction of vertex pairs in the same cluster, which are also in the same k-clique community.

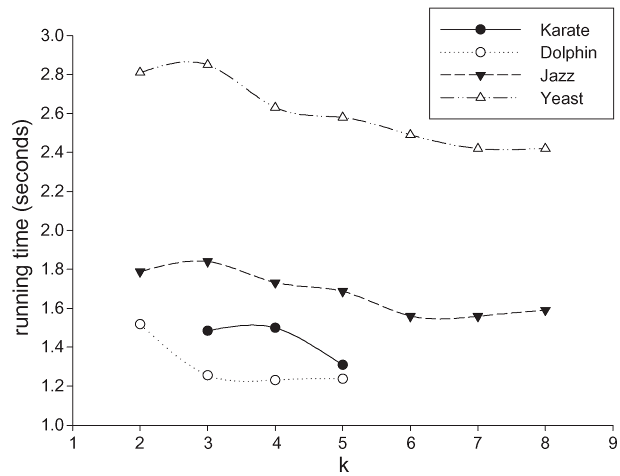


Fig. 8. Detection time of the proposed algorithm for data sets.

We calculate the F-measure for each data set with various approaches. Fig. 7(b) shows the F-measure values for various algorithms. Obviously, our approach has the largest F-measure value compared with other existing algorithms. As aforementioned, a good F-measure value can evaluate how well an algorithm can find the k-clique community structure from a social network. In other words, our approach can detect the k-clique community structure very well in a social network.

3) Detection Time: We provide the detection time of each data set using the proposed detection algorithm. Due to the impact of the system process, we simulate it five times and calculate the average detection time of each data set with our detection algorithm. Fig. 8 reveals an interesting conclusion: As parameter k increases, the detection time is changed without a special pattern. In particular, when we want to detect the bigger k-clique communities, the detection time is less than the smaller k-clique communities. In addition, as the scale of the data set increases, the time consumed increases. In particular, the average detection time of k-clique communities for the *Yeast* data set is around twice that of other data sets.

4) Correlation Between k and Number of k-Clique Communities: This section presents a correlation between k and the number of k-clique communities. In particular, we add one more big size of data set, NetHEP,[4] which is a collaboration network between authors, to observe the correlation results clearly. It contains 15 233 nodes representing the authors and 58 891 edges representing the collaboration between authors. We examine and present the correlation between k and the number of k-clique communities for all data sets in Fig. 9. In this figure, we know that the number of k-clique communities decreases with increasing k.

VI. CONCLUSION

This paper targets to detect the k-cliques and k-clique communities from a social network for providing the computational intelligence for CPSSs as well as enhancing the

[4]http://research.microsoft.com/en-us/people/weic/projects.aspx

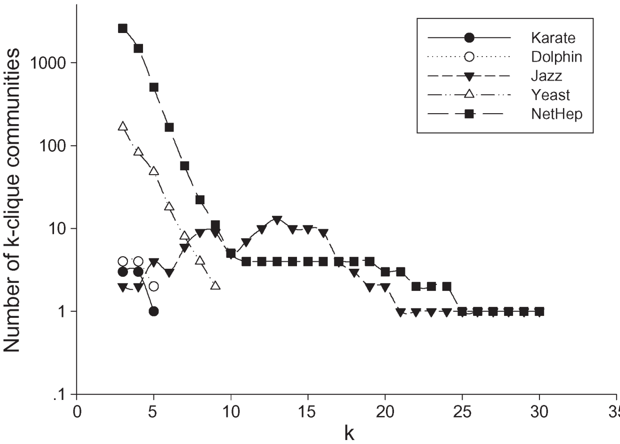


Fig. 9. Correlation between k and the number of k-clique communities.

interconnections among cyber, physical, and social spaces. We have proposed the FCA-based k-cliques and k-clique community detection algorithms. To devise the proposed detection algorithms, a solution for the formal context construction of a social network by using a modified adjacency matrix has been provided first. We have presented the new concepts k-equiconcepts and k-intent concepts and proved that the k-clique detection problem is equivalent to finding the k-equiconcepts, and the k-clique community detection problem is equivalent to finding the k-intent equiconcepts in the concept lattice of a social network. The proposed algorithm has been evaluated using four data sets. Experimental results have shown that the proposed algorithm has a higher F-measure value compared with other previous works. In addition, a correlation between k and the number of k-clique communities was investigated.

As the rapid growth of online social network sites continues, the community intelligence from social networks is widely used everywhere. From a social sustainable point of view, we plan to develop similar techniques in other urban sustainable applications, e.g., targeted marketing, and E-health field, to confirm that our approach is universally applicable in various domains.

References

[1] J. Shi, J. Wan, H. Yan, and H. Suo, "A survey of cyber-physical systems," in *Proc. Int. Conf. WCSP*, 2011, pp. 1–6.

[2] Y. Hu, F. Y. Wang, and X. Liu, "A CPSS approach for emergency evacuation in building fires," *IEEE Intell. Syst.*, vol. 29, no. 3, pp. 48–52, May/Jun. 2014.

[3] J. M. Zurada *et al.*, "Building virtual community in computational intelligence and machine learning," *IEEE Comput. Intell. Mag.*, vol. 29, no. 3, pp. 1–4, Feb. 2009.

[4] N. Du, B. Wu, and X. Pei, "Community detection in large-scale social networks," in *Proc. KDD*, 2007, pp. 16–25.

[5] S. Krajci, *Social Network and Formal Concept Analysis, Social Networks: A Framework of Computational Intelligence*. New York, NY, USA: Springer International Publishing, 2014.

[6] D. Zhang, D. Zhang, H. Xiong, C. H. Hsu, and A. V. Vasilakos, "BASA: Building mobile Ad-Hoc social networks on top of android," *IEEE Netw.*, vol. 28, no. 1, pp. 4–9, Jan./Feb. 2014.

[7] H. Shen, X. Cheng, K. Cai, and M. B. Hu, "Detect overlapping and hierarchical community structure in networks," *Phys. A, Stat. Mechan. Appl.*, vol. 388 , no. 8, pp. 1706–1712, Apr. 2008.

[8] E. Gregori, L. Lenzini, and S. Mainardi, "Parallel (k)-clique community detection on large-scale networks," *IEEE Tran. Parallel Distrib. Syst.*, vol. 24, no. 8, pp. 1651–1660, Aug. 2013.

[9] J. Yang, J. McAuley, and J. Leskovec, " Community detection in networks with node attributes," in *Proc. IEEE 13th Int. Conf. Data Mining*, 2013, pp. 1151–1156.

[10] B. Adamcsek, G. Palla, I. Farkas, I. Derenyi, and T. Vicsek, "CFinder: Locating cliques and overlapping modules in biological networks," *Bioinformatics*, vol. 22, no. 8, pp. 1021–1023, Apr. 2006.

[11] J. Kumpula, M. Kivela, K. Kaski, and J. Saramaki, "Sequential algorithm for fast clique percolation," *Phys. Rev. E, Stat. Nonlinear, Soft Matter Phys.*, vol. 78, no. 2, Aug. 2008, Art. ID. 026109.

[12] G. Palla, I. Derenyi, I. Farksa, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, Jun. 2005.

[13] K. Saito, T. Yamada, and K. Kazama, "The k-dense method to extract communities from complex networks," in *Mining Complex Data*, vol. 165, Studies in Computational Intelligence. Berlin, Germany: Springer-Verlag, 2009, pp. 243–257.

[14] D. Duan, Y. Li, R. Li, and Z. Lu, "Incremental k-clique Clustering in Dynamic Social Networks," in *Artificial Intelligence Review*. Berlin, Germany: Springer-Verlag, 2011.

[15] P. Tang, S. C. Hui, and A. C. Fong, "A lattice-based approach for chemical structural retrieval," *Eng. Appl. Artif. Intell.*, vol. 39, pp. 215–222, Mar. 2015.

[16] V. Snael, Z. Horak, and A. Abraham, "Understanding social networks using formal concept analysis," in *Proc. WI-IAT*, 2008, vol. 3, pp. 390–393.

[17] V. Snasel, Z. Horak, J. Kocibova, and A. Abraham, "Analyzing social networks using formal concept analysis: Complexity aspects," in *Proc. WI-IAT*, 2009, pp. 38–41.

[18] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E, Stat. Nonlinear Soft Matter Phys.*, vol. 69, no. 2, Feb. 2004, Art. ID. 026113.

[19] L. Wan, J. Liao, and X. Zhu, "CDPM: Finding and evaluating community structure in social networks," in *Proc. ADMA*, 2008, vol. 5139, LNCS, pp. 620–627.

[20] K. Li, Y. Du, D. Xiang, H. Chen, and Z. Liao, "A method for building concept lattice based on matrix operation," in *Proc. of ICIC*, 2007, pp. 350–359.

[21] F. Hao, S. S. Yau, G. Min, and L. T. Yang, "Detecting k-balanced trusted cliques in signed social networks," *IEEE Internet Comput.*, vol. 18, no. 2, pp. 24–31, Mar./Apr. 2014.

[22] W. X. Zhang and G. F. Qiu, *Uncertain Decision Making Based on Rough Sets*. Beijing, China: Tsinghua University Publisher, 2005, pp. 12–58.

[23] I. Derenyi, G. Palla, and T. Vicsek, "Clique percolation in random networks," *Phys. Rev. Lett.*, vol. 94, no. 16, pp. 76–85, Apr. 2005.

[24] G. Palla, A. L. Barabasi, and T. Vicsek, "Quantifying social group evolution," *Nature*, vol. 446, no. 7136, pp. 664–667, Apr. 2007.

[25] S. Gregory, "An algorithm to find overlapping community structure in network," in *Proc. PKDD*, 2007, pp. 91–102.

[26] X. Li, B. Liu, and P. S. Yu, "Discovering overlapping communities of named entities," in *Proc. PKDD*, 2006, pp. 593–600.

[27] J. Baumes, M. Goldberg, and M. Magdon-Ismail, "Efficient identification of overlapping communities," in *Intelligence and Security Informatics*, vol. 3495, LNCS, P. Kantor, G. Muresan, F. Roberts, D. D. Zeng, F.-Y. Wang, H. Chen, and R. C. Merkle, Eds. Berlin, Germany: Springer-Verlag, 2005 pp. 27–36.

Fei Hao received the B.S. and M.S. degrees from the School of Mathematics and Computer Engineering, Xihua University, Chengdu, China, in 2005 and 2008, respectively. He completed the doctoral coursework from the Korea Advanced Institute of Science and Technology, Daejeon, Korea. He is currently working toward the Ph.D. degree in the Department of Computer Software Engineering, Soonchunhyang University, Asan, Korea.

He was a Research Fellow with Huazhong University of Science and Technology, Wuhan, China. His research interests include social computing, big data analysis and processing, and mobile cloud computing.

Geyong Min received the B.Sc. degree in computer science from Huazhong University of Science and Technology, Wuhan, China, in 1995 and the Ph.D. degree in computing science from the University of Glasgow, Glasgow, U.K., in 2003.

He is a Professor of high-performance computing and networking with the Department of Mathematics and Computer Science, College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter, U.K. His research interests include future Internet, computer networks, wireless communications, multimedia systems, information security, high-performance computing, ubiquitous computing, modeling, and performance engineering.

Zheng Pei received the M.S. and Ph.D. degrees from Southwest Jiaotong University, Chengdu, China, in 1999 and 2002, respectively.

He is currently a Professor with the School of Computer and Software Engineering, Xihua University, Chengdu. His research interests are rough set theory, fuzzy set theory, logical reasoning, and linguistic information processing.

Doo-Soon Park (M'88) received the Ph.D. degree in computer science from Korea University, Seoul, Korea, in 1988.

He is a Professor with the Department of Computer Software Engineering, Soonchunhyang University, Asan, Korea. He is the President of the Korea Information Processing Society (KIPS) and the Director of the Central Library and of the Wellness Service Coaching Center with Soonchunhyang University. He was Dean of the Engineering College from 2002 to 2003 and the Director of the u-Healthcare Research Center from 2006 to 2007 with Soonchunhyang University. His research interests include data mining, big data processing, and parallel processing.

Dr. Park was Editor-in-Chief of the Journal of Information Processing Systems at KIPS from 2009 to 2012. He has served as an organizing committee member of international conferences, including CUTE 2014, CSA 2014, EMC-14, FutureTech 2014, and MUE 2014. He is a member of the Association for Computing Machinery, the KIPS, the Korean Mathematical Society, and the Korean Institute of Information Scientists and Engineers.

Laurence T. Yang received the B.E. degree in computer science and technology from Tsinghua University, Beijing, China, and the Ph.D. degree in computer science from the University of Victoria, Victoria, Canada.

He is a Professor with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China, as well as with the Department of Computer Science, St. Francis Xavier University, Antigonish, Canada. He has published more than 220 papers in various refereed journals (about 40% in IEEE/ACM Transactions/Journals and the others mostly in Elsevier, Springer, and Wiley Journals). His research interests include parallel and distributed computing, embedded and ubiquitous/pervasive computing, and big data. His research has been supported by the National Sciences and Engineering Research Council of Canada and the Canada Foundation for Innovation.