Neurocomputing 256 (2017) 101–110

Contents lists available at ScienceDirect

Neurocomputing

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

Iceberg Clique queries in large graphs

Fei Hao[a], Zheng Pei[b], Doo-Soon Park[a,*], Laurence T. Yang[c], Young-Sik Jeong[d], Jong-Hyuk Park[e]

[a] Department of Computer Software Engineering, Soonchunhyang University, Asan, Republic of Korea
[b] Center for Radio Administration and Technology Development, Xihua University, Chengdu, China
[c] Department of Computer Science, St. Francis Xavier University, Antigonish, NS, Canada
[d] Department of Multimedia Engineering, Dongguk University, Seoul, Republic of Korea
[e] Department of Computer Science and Engineering, Seoul National University of Science and Technology, Seoul, Republic of Korea

ARTICLE INFO

Article history:
Received 27 July 2015
Revised 2 December 2015
Accepted 24 December 2015
Available online 4 March 2017

Keywords:
Iceberg concept lattice
Iceberg Clique
Formal context

ABSTRACT

This paper investigates the Iceberg Clique (*IC*) queries in a large graph, specially, given a user-specified threshold θ, an *IC* query reports the cliques where the number of vertices exceeds $\lfloor\theta|V|\rfloor$. Toward this end, a practical *IC* query theorem is formally proposed and proved. With this proposed query theorem, a formal context and its corresponding iceberg concept lattice are first constructed from an input graph topology by Modified Adjacency Matrix; then, we prove that the *IC* queries problem is equivalent to finding the iceberg equiconcepts whose number of elements exceeds $\lfloor\theta|V|\rfloor$. Theoretical analysis and experimental results demonstrate that the proposed query algorithm is feasible and efficient for finding the iceberg cliques from large graphs.

1. Introduction

1.1. Motivations

As a fundamental problem, clique query in graphs has many significant applications, such as social recommendation [1–3], network routing [4,5] and community detection [6–8]. However, a practical issue of clique queries in real-world graphs is just to extract the cliques with the required number of vertices from a given large graph. For instance, in an academic collaboration network, we may find the collaborative teams with the required number of scholars for completing some projects. Hence, it is essential to extract only the important information from a graph.

Given a large graph, how can we find interesting vertices or sub-graph structures? With this question, this paper introduces a generic concept of *iceberg clique* that refers to cliques where the number of vertices satisfies that $\{v|v \in V, |v| \geq \lfloor\theta|V|\rfloor\}$, where $\theta \in (0, 1]$ is a user-specific threshold and $|V|$ is the total number of vertices in the given large graph. The terminology, "*iceberg*", is borrowed from the concept of iceberg queries in [10]. The iceberg query finds data whose aggregate values exceed a pre-specified threshold. Iceberg cliques are obtained as those whose aggregate score is above a given threshold. Note that, the score of each vertex is set as "1", hence, the problem of "*Iceberg clique query*" is equivalent to finding the cliques where the number of vertices exceeds $\lfloor\theta|V|\rfloor$.

Iceberg clique queries are applicable to target marketing, recommendation systems, routing, spammers detection and so forth. First, let us consider a concrete scenario on target marketing (community marketing) as a motivating example. A company published a very cool online game for an online social network and wants to market it through the same network. To this end, there are two main schemes: 1) Influence maximization based social marketing [11]; 2) emerging target/community marketing based on community structures. We focus on the later one, the basic idea of target marketing is to promote their products within the detected communities that are formed by the cliques. Due to the strong connections between users within these cliques, thus this product can be influenced and accepted rapidly. Second, *IC queries* can be used to detect the criminal groups for urban safety and security. For example, governments have limited human resources and budgets, they wants to efficiently control the main bigger criminal groups, which can be modeled as the *IC queries* problem.

1.2. Contributions

The contributions of this paper are summarized as follows:

- We define a novel concept–iceberg clique that refers the portion of the number of vertices in the cliques structure and

* Corresponding author.
E-mail addresses: fhao@sch.ac.kr (F. Hao), pqyz@263.net (Z. Pei), parkds@sch.ac.kr (D.-S. Park), ltyang@gmail.com (L.T. Yang), ysjeong@dongguk.edu (Y.-S. Jeong), jhpark1@seoultech.ac.kr (J.-H. Park).

http://dx.doi.org/10.1016/j.neucom.2015.12.142
0925-2312/

the number of vertices of the input graph exceeds the a given threshold. Then, the problem of *Iceberg Clique queries* (*IC queries*) is formally defined.

- To solve the proposed *IC queries* problem, an efficient approach based iceberg concept lattice for discovering the iceberg cliques is provided. Specifically, a formal context of a given graph is initially constructed according to Modified Adjacency Matrix. Then, the corresponding iceberg concept lattice is generated based on the constructed formal context. Finally, we propose a theorem of *IC queries* in large graphs.
- We conduct the experiments and performance evaluation using three graph datasets for validating the effectiveness of the proposed approach. The experimental results demonstrate the proposed algorithm can achieve the high precision and recall of *IC queries* for the large scale graph.

1.3. Paper organization

The remainder of this paper is structured as follows. Section 2 presents the related work. The problem description and solution framework for iceberg query are proposed in Section 3. Section 4 presents the detailed approach and algorithm for querying the iceberg cliques from large graphs. The experimental results and performance evaluation are provided in Section 5. Section 6 concludes this paper.

2. Related work

This section discusses the related study focusing on the clique detection and iceberg query.

2.1. Clique detection

The researches on clique detection are widely investigated. And, these work are categorized two types: *k*-clique detection and *k*-clique communities detection [12–15]. The first work on *k*-clique community detection method is the Clique Percolation Method (CPM) which defines a model of rolling a *k*-clique template [12]. But CPM did not propose object function to quantitatively qualify the clustering results. Kumpula et al. [16] proposed an efficient approach Sequential Clique Percolation (SCP) for *k*-clique community detection. Although SCP can detect communities on weighted networks, it cannot generate *k*-clique communities for each possible *k* in a single execution. Additionally, some extended cliques detection are also carried out [17,26]. Traag and Bruggeman [17] adopted the concept of modularity to detect communities in complex social networks with both positive and negative links. Hao et al. [26] first addressed the *k*-balanced trusted cliques detection in signed social networks.

2.2. Iceberg query

Recently, the researches on iceberg queries in database community have attracted researchers significantly. Iceberg queries are used in many practical applications such as data warehousing and market-basket analysis [18]. This section reviews the recent studies available in the literature for evaluation of iceberg queries. He et al. [19] proposed an approach of executing the iceberg queries efficiently using the compressed bitmap index. Their approach eliminates the time of scanning and processing the data set in order to speed up the query processing. Bae *et al* [20] presented a partitioning algorithm for computation of Average Iceberg Queries. The basic idea of this algorithm is to partition a relation logically and to postpone partitioning to use memory efficiently until all buckets are occupied with candidates. Zhao et al. [21] implemented the global iceberg query processing by using summable sketches. The above work were presented on the pure data query fields. Actually, some existing researches attempted to use iceberg query to solve some issues about graph queries. In Refs. [22,23], an iceberg cube containing cells whose measure satisfies a threshold, paved a way for iceberg analysis on graphs. Li et al. [24] defined a novel concept, graph iceberg, which extends iceberg queries to vertex-attributed graphs and identifies interesting vertices that are close to an attribute using an aggregate score. Further, they proposed a framework *gIceberg*, scores vertices by their different levels of interestingness and finds important vertices that meet a user-specified threshold.

Fig. 1. Cliques with 1,2,3,4,5 and 6 vertices.

After the detailed survey of above related work, it is obvious to know that there exists a research gap between clique detection and iceberg query. As mentioned above, most of work on iceberg query only take the property of nodes into account in the graph. Thus, the information granularity of nodes are too small to understand the global structure of graphs. Cliques as a relative big information granularity, can lead to much challenges for querying some hidden special cliques structures (so called iceberg cliques in the paper) from large graphs. In order to fill out this gap and bridge the connection between them, this paper pioneers the problem on *IC queries* in large graphs.

3. Problem definition and solution framework

In this section, the preliminaries on clique and $k-$clique are first reviewed. Then, the problem of *Iceberg Clique Queries* is formally defined. Additionally, a solution framework for addressing the *IC queries* is presented.

3.1. Problem definition

As the preliminaries, the definitions of clique and *k*-clique are provided as follows [26,27],

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 $(v_i, v_j) \in E$.

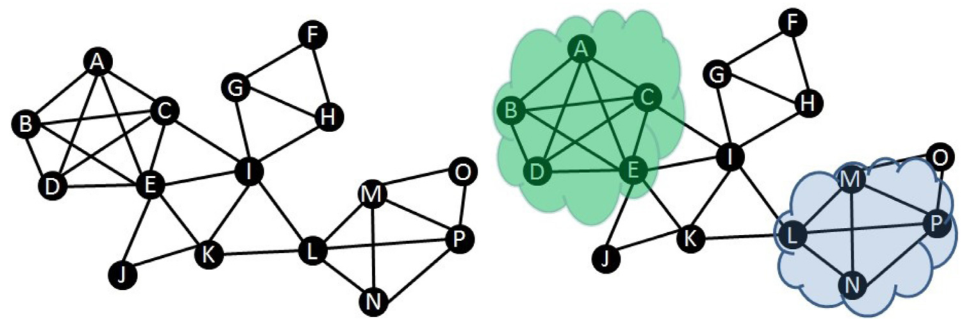
Fig. 1 presents an example of *k*-clique, $(k = 1, 2, \cdots, 6)$. The clique structure, where there must be an edge for each pair of vertices, shows many restrictions in real life modeling.

Problem 1 (Iceberg Clique Query)**.** For an undirected graph $G = (V, E)$ and a user-specified threshold $\theta \in (0, 1]$, the Iceberg clique (IC) query problem is to find out *k*-cliques, such that $k >= \lfloor \theta |V| \rfloor$, denoted as $\mathbb{Q} = (G(V, E), \theta)$.

For convenience, Table 1 lists important variables used throughout the paper.

To clearly understand the problem addressed in this paper, an example of iceberg clique queries in a graph g with a threshold $\theta = 0.25$, i.e., $\mathbb{Q} = (g, 0.25)$, is presented in Fig. 2.

Obviously, Fig. 2(a) is the structure of a graph g, then, after the *IC queries*, several iceberg cliques including 4 and 5 vertices, i.e., {*A*, *B*, *C*, *D*}, {*A*, *B*, *C*, *E*}, {*B*, *C*, *D*, *E*}, {*A*, *C*, *D*, *E*}, {*A*, *B*, *D*, *E*}, {*L*, *M*, *N*, *P*}, and {*A*, *B*, *C*, *D*, *E*} are detected in g as shown in Fig. 2(b).



(a) Graph g (b) Iceberg Cliques

Fig. 2. A toy example of Iceberg Clique queries.

Table 1
Important variables used in the paper.

Variables	Descriptions
$G = (V, E)$	An undirected graph
θ	User-specified threshold
k	number of nodes in k-cliques
$\mathbb{Q} = (g, \theta)$	An iceberg clique queries in g
T	A formal context
$FC(G)$	Formal context of G
$\widehat{C}(T)$	The set of all frequent concepts with respect to T
$L = (\widehat{C}(T), \leq)$	Iceberg concept lattice of T
$IC(T)$	Set of all iceberg equiconcepts of T
$KIC(T)$	Set of all k-iceberg equiconcepts of T

3.2. Solution framework

The main solution of this paper is to propose a novel theorem of iceberg clique queries that iceberg clique queries is equivalent to detecting the iceberg equiconcepts from the concept lattice. Fig. 3 presents a solution framework for iceberg clique queries. Our proposed solution framework contains four key technical steps: 1) constructing the formal context from an input graph; 2) mining the iceberg concept lattice; 3) detecting the explicit iceberg cliques generated from the iceberg equiconcepts and the implicit iceberg cliques derived from high-order iceberg equiconcepts; 4) combining the explicit and implicit iceberg cliques into the query output.

4. Iceberg concept lattice based IC queries in large graphs

This section provides an novel query approach of iceberg cliques using Iceberg Concept Lattice theory [25]. We address the following key issues of the above solution framework: 1) construct a formal context from a social network G; 2) extract the frequent concepts for forming an iceberg concept lattice; 3) present a generic algorithm for querying iceberg cliques.

4.1. Formal context construction on vertices set

In formal concept analysis, a formal context is represented as a triple $T = (O, A, R)$, where $O = \{x_1, x_2, \cdots, x_n\}$ is the set of objects, $A = \{a_1, a_2, \cdots, a_m\}$ is the set of attributes, R is the binary relation between O and A, $(x, a) \in R$ denotes object x has the attribute a, and $(x, a) \notin R$ denotes object x does not have the attribute a, where $x \in U$, $a \in A$. In this paper, we construct the formal context of a graph by using the approach proposed in our previous work [26]. The basic idea of this construction approach is that an Modified Adjacency Matrix of a given graph $G = (V, E)$ is represented as a formal context $FC(G) = (V, V, I)$, in which I indicates the binary relationship between two vertices, i.e., for any v_i, $v_j \in V$, $I(v_i, v_j) = 1$ if $(v_i, v_j) \in E$, otherwise, $I(v_i, v_j) = 0$. As a special case, we limit $I(v_i, v_i) = 1$ for any $i \in \{1, 2, \cdots, n\}$, obviously, $FC(G) = (V, V, I)$ is a special formal context because its sets of objects and attributes are the same.

Definition 3 (*Modified Adjacency Matrix*)**.** Let $G = (V, E)$ be a graph with n vertices that are assumed to be ordered from v_1 to v_n. The matrix $M = (m_{ij})_{n\times n}$ is called an Modified Adjacency Matrix, in which

$$m_{ij} = \begin{cases} 1, & \text{if } (v_i, v_j) \in E \text{ and } i \neq j; \\ 1, & \text{if } v_i = v_j (i = j); \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Formally, the Modified Adjacency Matrix M of $G = (V, E)$ is equivalent to the $FC(G)$, i.e., $FC(G) \equiv M$, and M is equivalent to

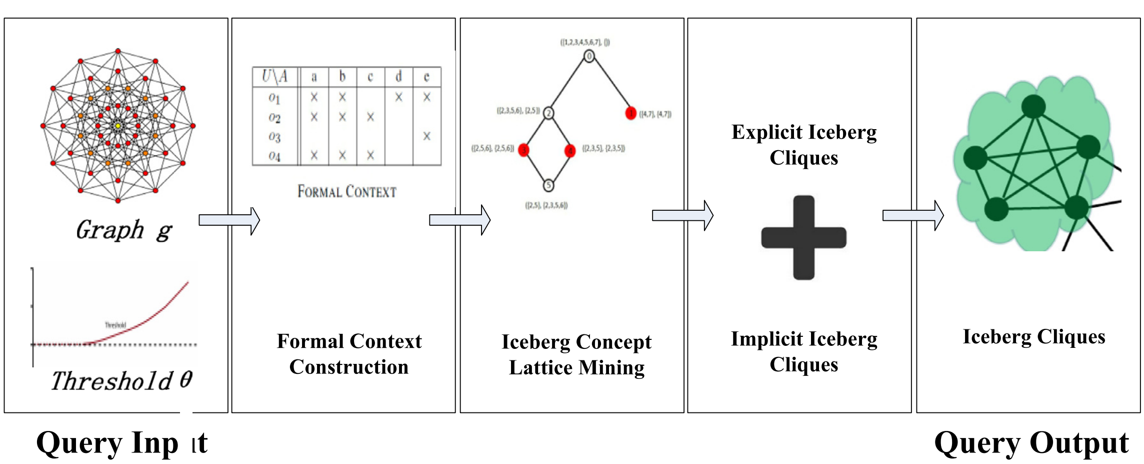


Fig. 3. Solution framework for Iceberg Clique queries.

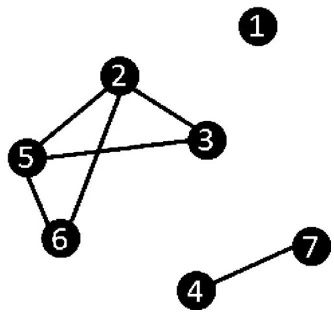


Fig. 4. An undirected graph $G = (V, E)$.

Table 2
The formal context of $G = (V, E)$.

Vertex	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	1	0	0	0	0	0	0
v_2	0	1	1	0	1	1	0
v_3	0	1	1	0	1	0	0
v_4	0	0	0	1	0	0	1
v_5	0	1	1	0	1	1	0
v_6	0	1	0	0	1	1	0
v_7	0	0	0	1	0	0	1

$G = (V, E)$ if we add $(v_i, v_i) \in E$ for any $i \in \{1, 2, \cdots, n\}$. More importance is that the formal context $FC(G)$ (or M) satisfy following notable properties.

1. **Reflexivity**: $\forall v_i \in V, I(v_i, v_i) = 1$ (or $m_{ii} = 1$);
2. **Symmetry**: $\forall v_i, v_j \in V, I(v_i, v_j) = I(v_j, v_i)$ (or $m_{ij} = m_{ji}$).

Example 1. Fig. 4 presents an undirected graph $G = (V, E)$, the formal context of $G = (V, E)$ is constructed in Table 2 based on the definition of Modified Adjacency Matrix.

4.2. Formal concept analysis on $FC(G) = (V, V, I)$

In formal concept analysis, for any formal context $T = (O, A, R)$, there exist two mappings induced from T to obtain all formal concepts [29], i.e.,

$$\uparrow : P(O) \longrightarrow P(A),$$
$$X \mapsto \uparrow (X) \triangleq X^{\uparrow} = \{a \in A | \forall x \in X, R(x, a) = 1\};$$
$$\downarrow : P(A) \longrightarrow P(O),$$
$$Y \mapsto \downarrow (Y) \triangleq Y^{\downarrow} = \{x \in O | \forall a \in Y, R(x, a) = 1\},$$

where $P(A)$ and $P(O)$ are power sets of O and A, respectively. Based on the two mappings, a formal concept of $T = (O, A, R)$ is formalized as the pair $(X, B) \in O \times A$ such that $X^{\uparrow} = B$ and $B^{\downarrow} = X$, X is called the extent of the concept, B is called the intent of the concept.

In the formal context $FC(G) = (V, V, I)$ of an undirected graph $G = (V, E)$, the above mentioned mapping can be rewritten by

$$\uparrow : P(V) \longrightarrow P(V),$$
$$X \mapsto \uparrow (X) \triangleq X^{\uparrow} = \{v_j \in V | \forall v_i \in X, I(v_i, v_j) = 1\};$$
$$\downarrow : P(V) \longrightarrow P(V),$$
$$Y \mapsto \downarrow (Y) \triangleq Y^{\downarrow} = \{v_i \in V | \forall v_j \in Y, I(v_i, v_j) = 1\}.$$

Because I of $FC(G) = (V, V, I)$ satisfies symmetry, the following property can be obtained.

Property 1. *In the formal context $FC(G) = (V, V, I)$ of an undirected graph $G = (V, E)$, for any $X \in V$, $X^{\uparrow} = X^{\downarrow}$.*

Proof. In $FC(G) = (V, V, I)$, I satisfies symmetry, this means $(\{v_i\})^{\uparrow} = \{v_j \in V | I(v_i, v_j) = 1\} = \{v_j \in V | I(v_j, v_i) = 1\} = (\{v_i\})^{\downarrow}$ for any $v_i \in V$, accordingly, for any $X \subseteq V$,

$$\begin{aligned} X^{\uparrow} &= \{v_j \in V | \forall v_i \in X, I(v_i, v_j) = 1\} \\ &= \bigcap_{v_i \in X} \{v_j \in V | I(v_i, v_j) = 1\} \\ &= \bigcap_{v_i \in X} (\{v_i\})^{\uparrow} \\ &= \bigcap_{v_i \in X} (\{v_i\})^{\downarrow} \\ &= \bigcap_{v_j \in X} (\{v_j\})^{\downarrow} \\ &= \{v_i \in V | \forall v_j \in X, I(v_i, v_j) = 1\} \\ &= X^{\downarrow}. \end{aligned}$$

This is the complete proof. □

Based on Property 1 and properties of ↑ and ↓ in formal concept analysis [9], the following property can be obtained.

Property 2. *In the formal context $FC(G) = (V, V, I)$ of an undirected graph $G = (V, E)$, for any $X_1, X_2 \in V$,*

1. $X_1 \subseteq X_1^{\uparrow\uparrow} = X_1^{\downarrow\downarrow} = X_1^{\uparrow\downarrow} = X_1^{\downarrow\uparrow}$;
2. $X_1^{\uparrow} = X_1^{\downarrow} = X_1^{\uparrow\uparrow\uparrow} = X_1^{\downarrow\uparrow\uparrow} = X_1^{\downarrow\downarrow\downarrow} = X_1^{\uparrow\downarrow\downarrow}$;
3. *if* $X_1 \subseteq X_2$, *then* $X_2^{\uparrow} \subseteq X_1^{\uparrow}$ *and* $X_2^{\downarrow} \subseteq X_1^{\downarrow}$;
4. $X_1 \subseteq X_2^{\uparrow}$ *if and only if* $X_2 \subseteq X_1^{\downarrow}$;
5. $(X_1 \cup X_2)^{\uparrow} = (X_1 \cup X_2)^{\downarrow} = X_1^{\uparrow} \cap X_2^{\uparrow} = X_1^{\downarrow} \cap X_2^{\downarrow}$;
6. $X_1^{\uparrow} \cup X_2^{\uparrow} = X_1^{\downarrow} \cup X_2^{\downarrow} \subseteq (X_1 \cap X_2)^{\uparrow} = (X_1 \cap X_2)^{\downarrow}$.

In $FC(G) = (V, V, I)$, Properties 1 and 2 induced the following corollary about its formal concepts.

Corollary 1. *For any $V_1, V_2 \in V$, if (V_1, V_2) is a formal concept of $FC(G) = (V, V, I)$, then (V_2, V_1) is a formal concept of $FC(G) = (V, V, I)$.*

Proof. Because (V_1, V_2) is a formal concept of $FC(G) = (V, V, I)$, we have $V_1^{\uparrow} = V_2$ and $V_2^{\downarrow} = V_1$. According to Property 1, we obtain $V_1^{\downarrow} = V_2$ and $V_2^{\uparrow} = V_1$, this means that (V_2, V_1) is a formal concept of $FC(G) = (V, V, I)$. □

According to Corollary 1, all formal concepts of $FC(G) = (V, V, I)$ can be roughly divided as two classes:

1. The extent of the concept is equal to the intent of the concept, i.e., the formal concept (V_1, V_1) such that $(V_1)^{\uparrow} = (V_1)^{\downarrow} = V_1$; This kind of concept is called as *Equiconcept* proposed in our previous work [26].
2. The extent of the concept is not equal to the intent of the concept, i.e., formal concepts (V_1, V_2) and (V_2, V_1) such that $(V_1)^{\uparrow} = V_2$ and $(V_2)^{\downarrow} = V_1$.

These can be seen from all formal concepts of Table 2 (shown in Fig. 5), in which, formal concepts ①, ③, ④, and ⑥ $((v_1, v_1))$ such that item 1), formal concepts ② and ⑤ such that item 2).

4.3. A binary relation on vertices set of an undirected graph

To show more detail structure properties of $FC(G) = (V, V, I)$ (or the undirected graph $G = (V, E)$), in this subsection, we define a mapping from vertices set V to its power set $P(V)$, then we induce an reflexive and transitive relation on V based on the mapping, and analyze several interesting properties of the relation on V.

For any formal context $FC(G) = (V, V, I)$ (or undirected graph $G = (V, E)$), the following mapping from V to $P(V)$ can be induced by the mapping ↑ from $P(V)$ to $P(V)$, i.e.,

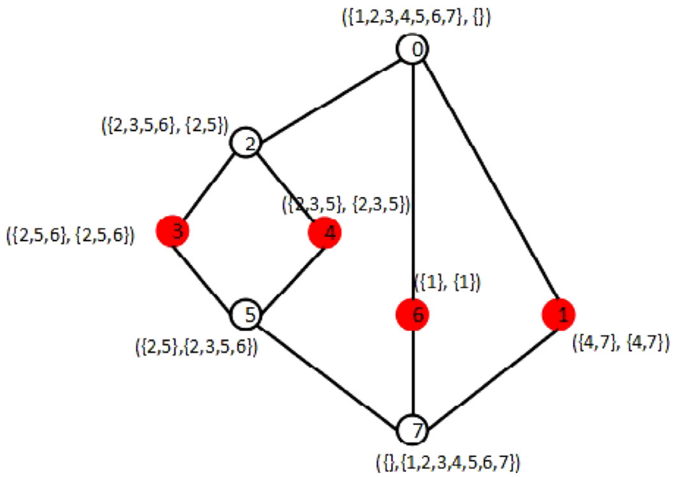


Fig. 5. All formal concepts of Table 2.

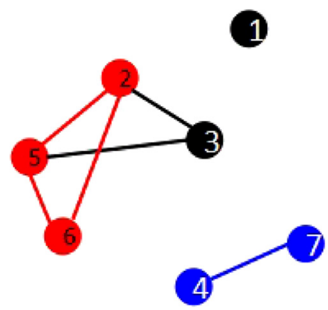


Fig. 6. $S_\uparrow(v_i)$ of $G=(V,E)$),e.g., $S_\uparrow(v_6)=\{v_2,v_5,v_6\}$, $S_\uparrow(v_4)=S_\uparrow(v_7)=\{v_4,v_7\}$.

$$S_\uparrow : V \longrightarrow P(V),$$

$$v_i \mapsto S_\uparrow(v_i) = \{v_j | v_j \in V \wedge v_j \in (\{v_i\})^{\uparrow\uparrow}\}. \quad (2)$$

Example 2. According to Table 2 and Eq. (2), we have $S_\uparrow(v_1)=\{v_1\}$, $S_\uparrow(v_2)=\{v_2,v_5\}$, $S_\uparrow(v_3)=\{v_2,v_3,v_5\}$, $S_\uparrow(v_4)=\{v_4,v_7\}$, $S_\uparrow(v_5)=\{v_2,v_5\}$, $S_\uparrow(v_6)=\{v_2,v_5,v_6\}$ and $S_\uparrow(v_7)=\{v_4,v_7\}$.

Intuitively, for any $v_i \in V$, $S_\uparrow(v_i)$ means that there must be an edge for each pair of vertices in $S_\uparrow(v_i)$ induced by the vertex v_i of $G=(V,E)$), such as in Fig. 6, $S_\uparrow(v_6)=\{v_2,v_5,v_6\}$ means that there exist edges for (v_2, v_5), (v_2, v_6) and (v_5, v_6). Similarly, $S_\uparrow(v_4)=S_\uparrow(v_7)=\{v_4,v_7\}$ means that the both vertices v_4 and v_7 can induce a 2-clique, i.e., $(\{v_4, v_7\})$. Based on the mapping $S_\uparrow$, we can define a binary relation R on V as follows: for any v_i, $v_j \in V$,

$$R(v_i,v_j) = \begin{cases} 1, & \text{if } v_j \in S_\uparrow(v_i), \\ 0, & \text{if } v_j \notin S_\uparrow(v_i). \end{cases} \quad (3)$$

Formally, the binary relation R decided by Eq. (3) shows relative relation among vertices of $G=(V,E)$, i.e., all vertices in which there exist edges each other own relation R, more important is that R satisfies the following property.

Property 3 [28]. *In the formal context $FC(G)=(V,V,I)$ of an undirected graph $G=(V,E)$, R is decided by Eq. (3), then*

1. *For any v_i, $v_j \in V$, if $v_j \in S_\uparrow(v_i)$, then $S_\uparrow(v_j) \subseteq S_\uparrow(v_i)$;*
2. *The binary relation R on V is reflexive and transitive.*

Example 3. In Example 2, due to $v_2 \in S_\uparrow(v_3)$, we have $S_\uparrow(v_2)=\{v_2,v_5\} \subset S_\uparrow(v_3)=\{v_2,v_3,v_5\}$. Due to $v_2 \in S_\uparrow(v_6)$ and $v_5 \in S_\uparrow(v_2)$, we have $R(v_6,v_2)=1$ and $R(v_2,v_5)=1$, respectively, furthermore, we also have $R(v_6,v_5)=1$ due to $v_5 \in S_\uparrow(v_6)$, i.e., $R(v_6,v_2)=1$ and $R(v_2,v_5)=1$ imply $R(v_6,v_5)=1$. Due to $v_6 \notin S_\uparrow(v_2)$, we have $R(v_2,v_6)=0$, this means that R is not symmetrical. R of Fig. 4 is shown in Table 3.

Table 3
R of $G=(V,E)$.

Vertex	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	1	0	0	0	0	0	0
v_2	0	1	0	0	1	0	0
v_3	0	1	1	0	1	0	0
v_4	0	0	0	1	0	0	1
v_5	0	1	0	0	1	0	0
v_6	0	1	0	0	1	1	0
v_7	0	0	0	1	0	0	1

According to Table 3, in $\{v_2, v_5\}$, $\{v_2, v_3, v_5\}$, $\{v_2, v_5, v_6\}$ or $\{v_4, v_7\}$, there exist edges each other, respectively, these can also be seen from Fig. 6.

Property 4. *In the formal context $FC(G)=(V,V,I)$ of an undirected graph $G=(V,E)$, $S_\uparrow$ is decided by Eq. (2), for any $v_i \in V$, $S_\uparrow(v_i)$ is a $|S_\uparrow(v_i)|$-clique in $G=(V,E)$.*

Proof. According to the mapping $\uparrow$ and Eq. (2), we have

$$S_\uparrow(v_i) = (\{v_i\})^{\uparrow\uparrow} = ((\{v_i\})^\uparrow)^\uparrow = (\{v_l \in V | I(v_i,v_l)=1\})^\uparrow$$
$$= \{v_j \in V | \forall v_{l'} \in \{v_l \in V | I(v_i,v_l)=1\}, I(v_{l'},v_j)=1\}.$$

In which, on the one hand, $v_i \in \{v_l \in V | I(v_i,v_l)=1\}$ and $v_i \in S_\uparrow(v_i)$ is obvious. On the other hand, for any $v_{j'}, v_{j''} \in S_\uparrow(v_i)$, $I(v_i,v_{j'})=1$ due to $v_i \in \{v_l \in V | I(v_i,v_l)=1\}$, hence, we have $v_{j'} \in \{v_l \in V | I(v_i,v_l)=1\}$, this means $I(v_{j'},v_{j''})=1$ due to $v_{j'} \in \{v_l \in V | I(v_i,v_l)=1\}$ and $v_{j''} \in S_\uparrow(v_i)$, i.e., for any vertices $v_{j'}, v_{j''} \in S_\uparrow(v_i) \subset V$, there exists an edge $(v_{j'}, v_{j''}) \in E$, hence, $S_\uparrow(v_i)$ is a $|S_\uparrow(v_i)|$-clique in $G=(V,E)$. □

Based on Property 4 and Eq. (3), the following corollary is obvious.

Corollary 2. *In the formal context $FC(G)=(V,V,I)$ of an undirected graph $G=(V,E)$, for any $v_i \in V$, we denote $R(v_i,*)=\{v_j \in V | R(v_i,v_j)=1\}$, then $R(v_i,*)$ is a $|R(v_i,*)|$-clique in $G=(V,E)$.*

Example 4. In Fig. 6, $\{v_1\}$ is 1-clique in $G=(V,E)$, $\{v_2, v_5\}$ and $\{v_4, v_7\}$ are 2-clique in $G=(V,E)$, $\{v_2, v_3, v_5\}$ and $\{v_2, v_5, v_6\}$ are 3-clique in $G=(V,E)$.

4.4. The topological structure of an undirected graph

Based on the reflexive and transitive relation R on V of $G=(V,E)$, we will construct a topological space of vertices set V in this subsection. Formally, a reflexive and transitive relation on a set can be used to induce covering approximation space [30–32]. In our previous work [28], the reflexive and transitive relation on a set has been used to construct an approximation space and a topology for attributes of a formal context, respectively, and a base for the topology can be adopted to generate intensions of all formal concepts of the formal context and construct the formal concept lattice. Inspired by the method, here we will construct a topological space of vertices set V by using the reflexive and transitive relation R to represent relationships and the hierarchical structures of an undirected Graph $G=(V,E)$, and provide several interesting results to show topological analysis of undirected graph.

Definition 4. In the formal context $FC(G)=(V,V,I)$ of an undirected graph $G=(V,E)$, for any subgraph $V_1 \subseteq V$, lower vertices approximations of V_1 is

$$\underline{R}(V_1) = \{v_i \in V | R(v_i,*) \subseteq V_1\}. \quad (4)$$

According to $R(v_i,*)=\{v_j \in V | R(v_i,v_j)=1\}=S_\uparrow(v_i)$ and Property 3.(1), $\underline{R}(V_1)$ can also be rewritten by

$$\begin{aligned}\underline{R}(V_1) &= \{v_i \in V | R(v_i, *) \subseteq V_1\} \\ &= \{v_i \in V | S_\uparrow(v_i) \subseteq V_1\} \\ &= \bigcup_{S_\uparrow(v_i) \subseteq V_1} S_\uparrow(v_i). \end{aligned} \quad (5)$$

Example 5. Let us take the Fig. 2 as an example for illustrating the lower vertices approximations of a subgraph $V_1 \subseteq V$. First, the formal context of this social graph is constructed with modified adjacency matrix as shown in Table 4.

Further, the corresponding formal concept lattice can be easily obtained as shown in Fig. 7. Note that the formal concepts with the annotations are equiconcepts.

Suppose two subgraphs V_1, V_2 including the vertices *A, B, C, D, E* and the vertices *L, M, N, O, P*, i.e., $V_1 = \{A, B, C, D, E\}, V_2 = \{L, M, N, O, P\}$, the lower vertices approximation can be obtained $\underline{R}(V_1) = \{A, B, C, D, E\}$ and $\underline{R}(V_2) = \{L, M, N, O, P\}$ according to Eq. (5).

Theorem 1 [28]**.** *For any formal context $FC(G) = (V, V, I)$ of undirected graph $G = (V, E)$,*

1. *$T_R = \{\underline{R}(V_1) | \forall V_1 \subseteq V\}$ is a topology for V, and (V, T_R) is a topological space for V;*
2. *$B_R = \{S_\uparrow(v_i) | v_i \in V\}$ is a base for the topology T_R.*

Table 4
Formal context of Fig. 2.

Vertex	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
A	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
B	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
C	1	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0
D	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
E	1	1	1	1	1	0	0	0	1	1	1	0	0	0	0	0
F	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
G	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
H	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0
I	0	0	1	0	1	0	1	1	1	0	1	1	0	0	0	0
J	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0
K	0	0	0	0	1	0	0	0	1	1	1	1	0	0	0	0
L	0	0	0	0	0	0	0	0	1	0	1	1	1	1	0	1
M	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
N	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1
O	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1
P	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

Theorem 1 means that any undirected graph $G = (V, E)$ can be represented by its topology $T_R = \{\underline{R}(V_1) | \forall V_1 \subseteq V\}$, which is induced by the binary relation *R* decided by Eq. (3), it is more important that the topology $T_R = \{\underline{R}(V_1) | \forall V_1 \subseteq V\}$ for *V* can be generated by the base $B_R = \{S_\uparrow(v_i) | v_i \in V\}$, which is obtained from every vertex of *V* according to Eq. (2).

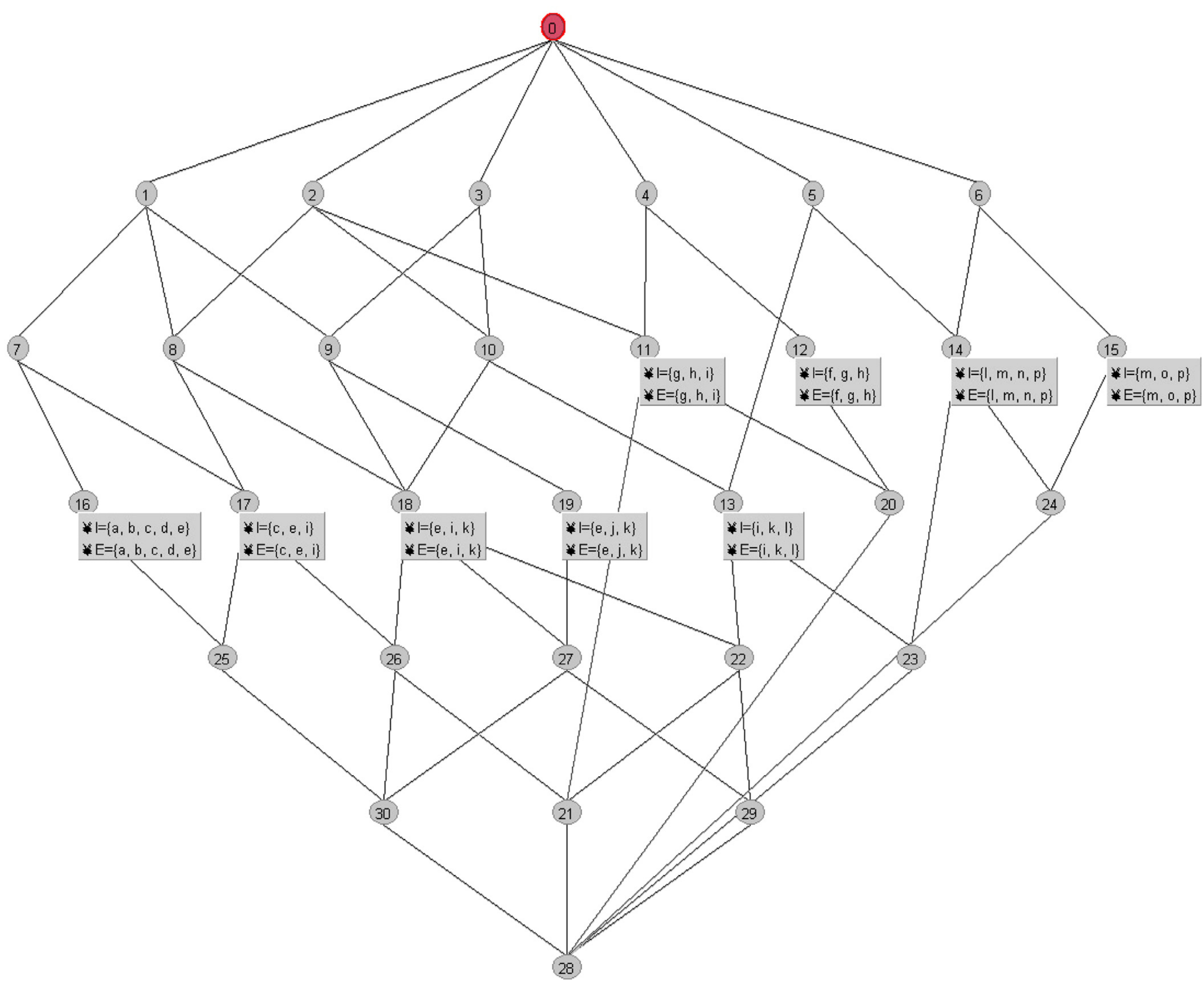


Fig. 7. The formal concept lattice of Fig. 2 (*Remark: The concepts with the annotations are the equiconcepts*).

According to Property 3. (1), we have the following corollary.

Corollary 3. *For any* v_i, $v_j \in V$, *if* $v_j \in S_{\uparrow}(v_i)$ *and* $v_i \in S_{\uparrow}(v_j)$, *then* $S_{\uparrow}(v_i) = S_{\uparrow}(v_j)$.

Due to $S_{\uparrow}(v_j)$ is a $|S_{\uparrow}(v_i)|$-clique in $G = (V, E)$ by Property 4, Corollary 3 means that v_i and v_j generate the same k-clique in $G = (V, E)$, i.e.,

Corollary 4. *If* v_i *and* v_j *are in a k-clique in* $G = (V, E)$, *then* $S_{\uparrow}(v_i) = S_{\uparrow}(v_j)$ *and they are the k-clique in* $G = (V, E)$.

According to Theorem 1 2) and Corollary 4, we have the following corollary.

Corollary 5. *Any undirected graph* $G = (V, E)$ *can be generated by its all k-clique in* $G = (V, E)$, *where* $1 \leq k \leq |V|$.

Example 6. Regarding to the above formal context, 31 formal concepts including 9 equiconcepts are extracted by using the formal concept lattice mining algorithm [33]. Without the loss of generality, for two vertices C and I 3-clique ({C, E, I}, {C, E, I}), the mappings $S_{\uparrow}(C)$ and $S_{\uparrow}(I)$ can be easily induced from C and I, i.e., $S_{\uparrow}(C) = \{A, B, C, D, E\}$ and $S_{\uparrow}(I) = \{I\}$.

4.5. Iceberg concept lattice mining

Iceberg concept lattice is a partially-ordered lattice formed by frequent concepts under a certain support degree. It clearly describes the correlations between frequent concepts. Therefore, this section focus on the study of iceberg concept lattice mining and the correlations between the concepts in this lattice. By revealing the correlations between concepts, we can easily find the iceberg cliques in a given large graph.

This section first recalls the related definitions of concept, frequent concept. Further, the basic definition of iceberg concept lattice is formally provided.

Definition 5 (Frequent Concept)**.** For a concept C, let $supp(C) = |X|/|O|$ be the support degree of concept C, if $supp(C) \geq minsupp$ ($minsupp$ is a minimal support degree (0,1] given by user), then C is called a frequent concept with the support degree of intent, namely $supp(B)$. And let $\widehat{C}(T)$ denotes the set of all frequent concepts with respect to formal context T.

Definition 6 (Iceberg Concept Lattice)**.** An iceberg concept lattice $L = (\widehat{C}(T), \leq)$ can be obtained by all frequent concepts $\widehat{C}(T)$ of a context T with the partial order $\leq$. Its graphical representation is a Hasse diagram.

Example 7. Let us continue with Example 1, we can build the iceberg concept lattice of the graph g, denoted as $L = (\widehat{C}(FC(g)), \leq)$. Fig. 8 shows an iceberg concept lattice (minsupp=25%) extracted from Fig. 4.

Obviously, the above iceberg concept lattice contains the topmost concepts of the concept lattice. And, these concepts will provide the most global structure of the domain since all the concepts in the lattices are the frequent concepts.

4.6. Iceberg Clique queries and algorithm

According to the problem statement of *IC queries*, the goal of *IC queries* is to find out the special cliques which satisfy that 1) the query results should be all cliques; 2) the number of vertices in these cliques is greater than $\lfloor\theta|V|\rfloor$.

This section introduces the concepts of Iceberg Equiconcept and k-Iceberg Equiconcept, then provides several interesting theorems and properties about *IC queries* with k-Iceberg Equiconcepts as the query index.

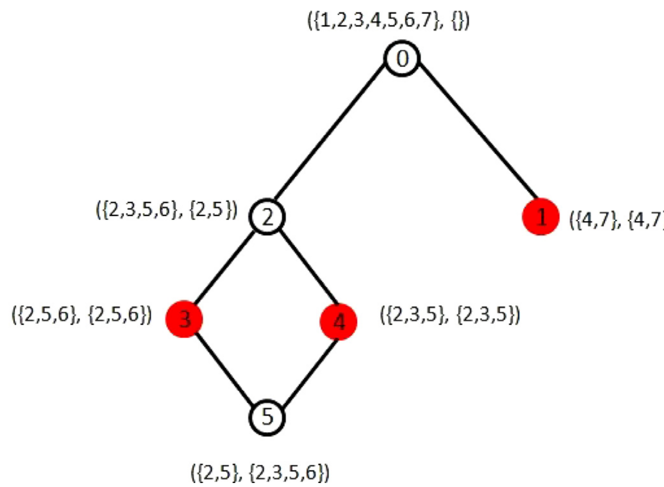


Fig. 8. Iceberg concept lattice ("*red*" nodes denote the cliques). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Definition 7 (Iceberg equiconcept)**.** For a formal context $T = (O, A, R)$, if a pair (X, B) satisfies $X^{\uparrow} = B$, $B^{\downarrow} = X$, $X = B$, and $\frac{|X|}{|O|} \geq \theta$ then the pair (X, B) is an Iceberg Equiconcept, where X is called the extent of the iceberg equiconcept, B is called the intent of the iceberg equiconcept. And let $IC(T)$ denotes the set of all iceberg equiconcepts with respect to formal context T.

Definition 8 (k-Iceberg equiconcept)**.** For a formal context $T = (O, A, R)$, if a pair (X, B) satisfies $X^{\uparrow} = B$, $B^{\downarrow} = X$, $X = B$, and $|X| = |B| = k \geq \lfloor\theta|O|\rfloor$, then the pair (X, B) is a k-iceberg equiconcept, where X is called the extent of the k-iceberg equiconcept, B is called the intent of the k-iceberg equiconcept. And let $KIC(T)$ denotes the set of all k-iceberg equiconcepts with respect to formal context T.

Clearly, the process of IC queries is similar to the identification of k-Iceberg Equiconcepts. Therefore, a theorem of IC queries in a large graph is provided as follows.

Theorem 2. *Given a graph G, iceberg clique queries with a threshold* θ *is equivalent to mining two types of iceberg equiconcepts: a) explicit iceberg cliques are generated from k-iceberg equiconcepts; b) implicit iceberg cliques are derived from* $k+1$*-iceberg equiconcepts,* $k+2$*-iceberg equiconcepts,* $\cdots$*, M-iceberg equiconcepts (*$M > k$*). M is the number of maximum extent or intent of maximum iceberg equiconcepts.*

Proof. For an *IC query* with a threshold $\theta \in (0, 1]$ in a graph G, we expect all the obtained cliques satisfy one key condition: the number of vertices is equal or greater than $\lfloor\theta|O|\rfloor$. In other words, all the iceberg equiconcepts in the iceberg concept lattice satisfy this condition, and these special concepts are regarded as the explicit iceberg cliques. Additionally, some hidden implicit iceberg cliques can be derived from its corresponding high-order iceberg concepts, $k+1$-iceberg equiconcepts, $k+2$-iceberg equiconcepts, $\cdots$, M-iceberg equiconcepts ($M > k$). For example, the explicit iceberg cliques queries($\mathbb{Q} = (g, 0.25)$) in Fig. 4 are easily detected according to its iceberg concept lattice, i.e., ({4, 7}, {4, 7}), ({2, 5, 6}, {2, 5, 6}), ({2, 3, 5}, {2, 3, 5}) are the explicit iceberg cliques. However, we can derive more iceberg 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}). But, they are not concepts, they do not appear in the iceberg concept lattice of the graph g. Hence, for an IC query ($\mathbb{Q} = (g, 0.25)$), the iceberg cliques are the union of explicit and implicit iceberg cliques, i.e., ({4, 7}, {4, 7}), ({2, 5, 6}, {2, 5, 6}), ({2, 3, 5}, {2, 3, 5}), ({2, 3}, {2, 3}), ({2, 5}, {2, 5}), ({2, 6}, {2, 6}), ({3, 5}, {3, 5}), and ({5, 6}, {5, 6}). □

Based on the above theorem of *IC queries* in a large graph, we devised Algorithm 1 as follows.

Algorithm 1 Iceberg Clique queries algorithm.

Require:
$G = (V, E)$;
Parameter $\theta \in (0, 1]$;
Ensure:
Set of Iceberg Cliques Q
1: Initialize $Q = \emptyset$
2: ***begin***
3: Construct a formal context $FC(G)$ by Definition 3
4: Build a concept lattice $L = (\widehat{C}(FC(G)), \leq)$
5: ***end***
6: ***for*** each concept $(X, B) \in \widehat{C}(FC(G))$
7: ***begin***
8: ***if*** $|X| = |B| \geq \lfloor \theta |V| \rfloor$
9: $Q \leftarrow Q \cup (X, B)$
10: $k \leftarrow \lfloor \theta |V| \rfloor$
11: ***for*** i=k+1 ***to*** M ***do***
12: ***begin***
13: $Q \leftarrow Q \cup Derived((X^i, B^i))$
14: ***end***
15: ***end***

Algorithm 1 works as follows: Initially, a graph G and a parameter θ are the inputs of whole algorithm; Then, we initialize a set of iceberg cliques with Q (Line 1). After the initialization of algorithm, it goes into the formal context construction and iceberg concept lattice generation codes part (Line 2–5). Line 6–9 find the explicit iceberg cliques by inserting the k-iceberg equiconcepts (X, B) into Q. The implicit iceberg cliques are derived from other high order k-iceberg equiconcepts and are inserted into Q (Line 10–14).

5. Experimental results and evaluation

The proposed approach deals with the queries of iceberg cliques based on iceberg concept lattice in large graphs. Our method constructs an efficient query index–iceberg concept lattice for getting better query efficiency. The performance of the proposed algorithm is evaluated in this section under different evaluation criteria. All algorithms are implemented in JAVA language and executed on an Intel core i5-3740 processor, 3.6 GHZ, 8 GB RAM computer.

5.1. Data sets

Karate Club Members Networks (Karate). This is a classical social network of friendships between 34 members of a karate club at a US university in the 1970s. The vertices are club members and the edges are their social interactions. This graph has 34 vertices and 78 edges.

Dolphin Living Networks (Dolphin). This is a small-size dataset on the social network of frequent associations between 62 dolphins in a community living off Doubtful sound, New Zealand. This graph contains 62 vertices and 159 edges.

Jazz Musicians Social Networks (Jazz). This dataset is obtained from The Red Hot Jazz Archive digital database. It is a network of Jazz musicians including 198 vertices and 5484 edges.

We execute a series of queries on each graph according to different setting of threshold θ. Without loss of generality, we choose the θ ranged from 0.02 to 0.2 with 0.02 step length, i.e., $\theta = \{0.02, 0.04, 0.06, 0.08, 0.1, 0.12, 0.14, 0.16, 0.18, 0.2\}$.

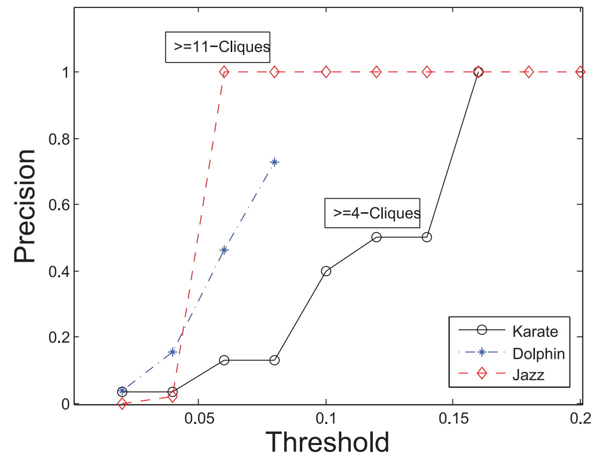


Fig. 9. IC query precision evaluation for datasets.

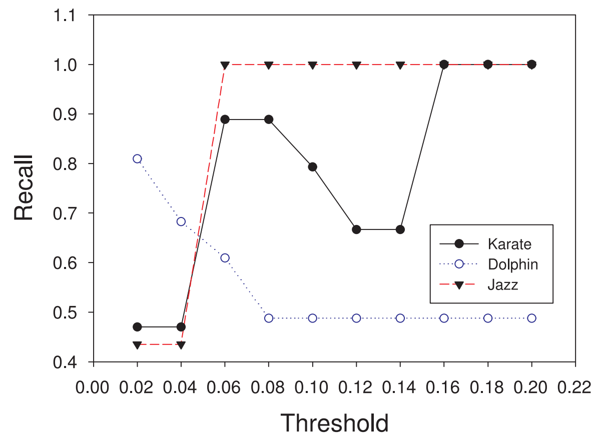


Fig. 10. IC query recall evaluation for datasets.

5.2. Performance evaluation

The performance evaluation of the proposed approach, mainly deals with high precision/recall for query operations. Thus the precision and recall of query operations are concentrated more on the approach. The three dataset are tested with the proposed iceberg cliques query algorithm. The responses of the datasets according to the proposed approach are detailed in the following graph. The experimentation is conducted on different threshold in the dataset. For a given threshold θ that receives a list of iceberg cliques, precision and recall are defined as follows:

- *Precision* is the ratio of the number of relevant iceberg cliques in the IC queries results.
- *Recall* is the ratio of the number of relevant iceberg cliques in the IC queries results to the total number of relevant iceberg cliques.
- *F1-Score* is used to evaluate how well each algorithm can find the iceberg cliques from a graph by fitting the *Precision* and *Recall*, denoted as $F1 = \frac{2*Precision*Recall}{Precision+Recall}$.

For *IC query* precision evaluation in Fig. 9, we separately plot the precision curves for all the datasets. As shown, the precision of *IC queries* for each dataset improves as the threshold θ increases. In particular, the precision of *IC queries* for the large graph dataset *Jazz* reaches the 100% when we find out the iceberg cliques where the number of vertices exceeds the 11. Similarly, as the threshold θ increases, recall for large datasets in Fig. 10 increases in general. For a large graph *Jazz*, the recall of the *IC queries* gets 100% when $\theta \geq 0.06$. Moreover, we also evaluate the proposed

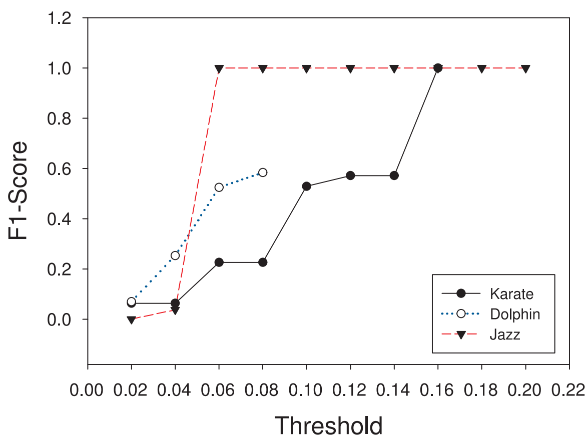


Fig. 11. IC query F1-score evaluation for datasets.

algorithm for three datasets in terms of F1-Score that is used to evaluate how well each algorithm can find the iceberg cliques from a graph. Fig. 11 reports that our algorithm can identify the iceberg cliques efficiently with increase of threshold θ.

The unique feature of the iceberg concept lattice based *IC queries* approach is that it can easily find out the iceberg cliques via constructed query index–*k-Iceberg Equiconcept*. The proposed algorithm represents a high robustness and scalability for a large scale graph according to the above performance analysis.

6. Conclusions

In this paper, we aim to address the query problem of iceberg cliques (*IC queries*) in large graphs. We propose the iceberg concept lattice based iceberg cliques detection algorithms. To devise proposed detection algorithms, we first provide a solution for formal context construction of a social network by using Modified Adjacency Matrix. Then, it is proved that the *IC queries* problem is equivalent to finding the iceberg equiconcepts whose number of elements exceeds $\lfloor\theta|V|\rfloor$. We evaluate the proposed algorithm using three datasets. Experimental results show that proposed algorithm has a high robustness and scalability for a large graph.

Acknowledgments

This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the C-ITRC (Convergence Information Technology Research Center) (IITP-2015-H8601-15-1009) supervised by the IITP (Institute for Information & Communications Technology Promotion) and was supported by Basic Science Research program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (no. NRF-2014R1A1A4A01007190). It was also supported by the Natural Science Foundation for Young Scientists of Shanxi Province, China (grant no. 2015021102).

References

[1] X. Li, Personalized information recommendation based on network bookmarks, in: Proc. of the 2012 International Conference on Cybernetics and Informatics, 2013, pp. 1981–1988.
[2] M.N. Hamid, M.A. Naser, M.K. Hasan, H. Mahmud, A cohesion-based friend-recommendation system, Social Network Anal. Mining 4 (1) (2014) 1–11.
[3] F. Hao, S. Li, G. Min, H.C. Kim, S.S. Yau, L.T. Yang, An efficient approach to generating location-sensitive recommendations in ad-hoc social network environments, IEEE Trans. Services Comput. 8 (3) (2015) 520–533.
[4] P. Krishna, N.H. Vaidya, M. Chatterjee, D.K. Pradhan, A cluster-based approach for routing in dynamic networks, ACM SIGCOMM Comput. Commun. Rev. 27 (2) (1997) 49–64.
[5] K. Biswas, V. Muthukkumarasamy, E. Sithirasenan, M. Usman, An energy efficient clique based clustering and routing mechanism in wireless sensor networks, in: Proc. of IEEE International Conference on Wireless Communications and Mobile Computing (IWCMC), 2013, pp. 171–176.
[6] G.W. Flake, S. Lawrence, C.L. Giles, Efficient identification of web communities, in: Proc. of KDD'2000, 2000, pp. 150–160.
[7] N. Du, B. Wu, X. Pei, Community detection in large-scale social networks, in: Proc. of KDD'07, 2007, pp. 16–25.
[8] K.R. Kim, N. Moon, Content modeling based on social network community activity, J. Inf. Process. Syst. 10 (2) (2014) 271–282.
[9] B. Ganter, R. Wille, Formal concept analysis, in: Mathematical Foundations, Springer, 1999.
[10] M. Fang, N. Shivakumar, H. Garcia-Molina, R. Motwani, J.D. Ullman, Computing iceberg queries efficiently, in: Proc. of VLDB, 1998, pp. 299–310.
[11] W. Chen, Y. Wang, S. Yang, Efficient influence maximization in social networks, in: Proc. of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009, pp. 199–208.
[12] G. Palla, I. Derenyi, I. Farksa, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, Nature 435 (2005) 814–818.
[13] I. Derenyi, G. Palla, T. Vicsek, Clique percolation in random networks, PRL 94 (160202) (2005) 76–85.
[14] G. Palla, A.L. Barabasi, T. Vicsek, Quantifying social group evolution, Nature 467 (2007) 664–667.
[15] E. Gregor, L. Lenzini, C. Orsini, k-clique communities in the internet AS-level topology graph, in: Proc. of ICDCSW 2011, 2011, pp. 134–139.
[16] J.M. Kumpula, M. Kivela, K. Kaski, J. Saramaki, Sequential algorithm for fast clique percolation, Phys. Rev. E 78 (2) (2008) 026109.
[17] V.A. Traag, J. Bruggeman, Community detection in networks with positive and negative links, Phys. Rev. E 80 (3) (2009) id.036115.
[18] A. Padmapriya, T. Priya, Methods for evaluating iceberg queries, Int. J. Comput. Appl. 68 (9) (2013) 6–9.
[19] B. He, H. H. Hsiao, Z. Liu, Y. Huang, Y. Chen, Efficient iceberg query evaluation using compressed bitmap index, IEEE Trans. Knowl. Data Eng. 24 (9) (2012) 1570–1584.
[20] J. Bae, S. Lee, Partitioning algorithms for the computation of average iceberg queries, in: Proc. of the Second International Conference on Data Warehousing and Knowledge Discovery, 2000, pp. 276–286.
[21] H.C. Zhao, A. Lall, M. Ogihara, J. Xu, Global iceberg detection over distributed data streams, in: Proc. of ICDE10, 2010, pp. 557–568.
[22] C. Chen, X. Yan, F. Zhu, J. Han, P.S. Yu, Graph OLAP: a multi-dimensional framework for graph data analysis, Knowl. Inf. Syst. 21 (1) (2009) 41–63.
[23] D. Xin, J. Han, X. Li, B.W. Wah, Star-cubing: computing iceberg cubes by top–down and bottom-up integration, in: Proc. of VLDB, 2003, pp. 476–487.
[24] N. Li, Z. Guan, L. Ren, J. Wu, J. Han, X. Yan, gIceberg: towards iceberg analysis in large graphs, in: Proc. of the 2013 IEEE International Conference on Data Engineering (ICDE '13), 2013, pp. 1021–1032.
[25] G. Stumme, R. Taouil, Y. Bastide, N. Pasquier, L. Lakhal, Computing iceberg concept lattices with TITANIC, J. Data Knowl. Eng. 42 (2) (2002) 189–222.
[26] F. Hao, S.S. Yau, G. Min, L.T. Yang, Detecting k-balanced trusted cliques in signed social networks, IEEE Int. Comput. 18 (2) (2014) 24–31.
[27] F. Hao, G. Min, Z. Pei, D.S. Park, L.T. Yang, k-clique communities detection in social networks based on formal concept analysis, IEEE Syst. J. (2015), doi:10.1109/JSYST.2015.2433294.
[28] Z. Pei, D. Ruan, D. Meng, Z. Liu, Formal concept analysis based on the topology for attributes of a formal context, Inf. Sci. 236 (2013) 66–82.
[29] R. Wille, Line diagrams of hierarchical concept systems, Int. Classification 11 (1984) 77–86.
[30] Y.-R. Syau, E.-B. Lin, Neighborhood systems and covering approximation spaces, Knowl. Based Syst. 66 (2014) 61–67.
[31] K. Qin, Y. Gao, Z. Pei, On covering rough sets, in: J. Yao, P. Lingras, W. Wu (Eds.), Rough Sets and Knowledge Technology, Lecture Notes in Artificial Intelligence, vol. 4481, Springer-Verlag, Berlin/Heidelberg, 2007, pp. 34–41.
[32] W. Zhu, Relationship between generalized rough sets based on binary relation and covering, Inf. Sci. 179 (2009) 210–225.
[33] S. Andrews, In-Close, a fast algorithm for computing formal concepts, in: Proc. of Conference on Conceptual Structures (ICCS), Moscow, 2009.

Fei Hao received the B.Sc. degree in information and computing science and the M.Sc. degree in computer software and theory from Xihua University, China, in 2005 and 2008, respectively, and the Ph.D. degree in computer science and engineering from Soonchunhyang University, South Korea, in 2016. He is currently a faculty member with the School of Computer Science, Shaanxi Normal University, China. He has authored over 60 papers in international journals and conferences. He received five best paper awards from KISM 2012, GreenCom 2013, MUE 2015, UCAWSN 2015 and CUTE 2016. He is a recipient of the IEEE Outstanding Leadership Award at CPSCom 2013 and the 2015 Chinese Government Award for Outstanding Self-Financed Students Abroad. His research interests include social computing, ubiquitous computing, big data analysis and processing and mobile cloud computing.

Zheng Pei received his M.S. and Ph.D. degrees from Southwest Jiaotong University, Chengdu, Sichuan, China, in 1999 and 2002 respectively. Currently, he is a professor in the School of Computer and Software Engineering, Xihua University, Chengdu, Sichuan, China. His research interests are rough set theory, fuzzy set theory, logical reasoning, and linguistic information processing.

Doo-Soon Park He received his Ph.D. in Computer Science from Korea University in 1988. Currently, he is a professor in the Department of Computer Software Engineering at Soonchunhyang University, South Korea. He is president of KIPS (Korea Information Processing Society) and director of Central Library at Soonchunhyang University and director of Wellness Service Coaching Center at Soonchunhyang University. He was editor in chief of JIPS (Journal of Information Processing Systems) at KIPS from 2009 to 2012. He was Dean of the Engineering College at Soonchunhyang University from 2002 to 2003, and was the director of the u-Healthcare Research Center at Soonchunhyang University from 2006 to 2007. He has served as an organizing committee member of international conferences including CUTE 2014, CSA 2014, EMC-14, FutureTech 2014, and MUE 2014. His research interests include data mining, big data processing and parallel processing. He is a member of IEEE, ACM, KIPS, KMS, and KIISE.

Laurence T. Yang received the B.E. degree in computer science and technology from Tsinghua University, China, and the Ph.D. degree in computer science from University of Victoria, Canada. He is a professor with the School of Computer Science and Technology, Huazhong University of Science and Technology, China, as well as with the Department of Computer Science, St. Francis Xavier University, Canada. His research interests include parallel and distributed computing, embedded and ubiquitous/pervasive computing, big data. 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 has been supported by the National Sciences and Engineering Research Council of Canada (NSERC) and the Canada Foundation for Innovation.

Young-Sik Jeong is a professor in the Department of Multimedia Engineering at Dongguk University in Korea. His research interests include multimedia cloud computing, information security of cloud computing, mobile computing, IoT (Internet of Things), and wireless sensor network applications. He received his B.S. degree in Mathematics and his M.S. and Ph.D. degrees in Computer Science and Engineering from Korea University in Seoul, Korea in 1987, 1989, and 1993, respectively. He was a professor in the Department of Computer Engineering at Wonkwang University in Korea from 1993 to 2012. He worked and researched to Michigan State University and Wayne State University as visiting professor in 1997 and 2004 respectively. Since 2002, he has been serving as an IEC/TC 100 Korean Technical Committee member, as the IEC/TC 108 Chairman of Korean Technical Committee, and as an ISO/IEC JTC1 SC25 Korean Technical Committee member. Also he is an EiC (Editor-in-Chief) of Journal of Information Processing Systems, an associate editor of JoS (Journal of Supercomputing), IJCS (International Journal of Communication Systems) and an editor of JIT (Journal of Internet Technology), finally an associate editor of Journal of Human-centric Computing (HCIS). In addition, he has been serving as a Guest Editor for international journals by some publishers: Springer, Elsevier, John Wiley, Oxford Univ. Press, Hindawi, Emerald, Inderscience and so on. He is also is a member of the IEEE. http://ucloud-lab.dongguk.edu .

Dr. James J. (Jong Hyuk) Park received Ph.D. degrees in Graduate School of Information Security from Korea University, Korea and Graduate School of Human Sciences from Waseda University, Japan. From December, 2002 to July, 2007, he had been a research scientist of R&D Institute, Hanwha S&C Co., Ltd., Korea. From September, 2007 to August, 2009, he had been a professor at the Department of Computer Science and Engineering, Kyungnam University, Korea. He is now a professor at the Department of Computer Science and Engineering and Department of Interdisciplinary Bio IT Materials, Seoul National University of Science and Technology (SeoulTech), Korea. He has published about 200 research papers in international journals and conferences. He has been serving as chairs, program committee, or organizing committee chair for many international conferences and workshops. He is a founding steering chair of some international conferences ? MUE, FutureTech, CSA, UCAWSN, etc. He is a president of the Future Technology Research Association International (FTRA) and Korea Information Technology Convergence Society (KITCS). He is editor-in-chief of Human-centric Computing and Information Sciences (HCIS) by Springer, International Journal of Information Technology, Communications and Convergence (IJITCC) by InderScience, and Journal of Convergence (JoC) by FTRA Publishing. He is associate editor/editor of 14 international journals including 8 journals indexed by SCI(E). In addition, he has been serving as a Guest Editor for international journals by some publishers: Springer, Elsevier, John Wiley, Oxford Univ. Press, Hindawi, Emerald, Inderscience. His research interests include security and digital forensics, Human-centric ubiquitous computing, context awareness, multimedia services, etc. He got the best paper awards from ISA-08 and ITCS-11 conferences and the outstanding leadership awards from IEEE HPCC-09, ICA3PP-10, IEE ISPA-11, and PDCAT-11. Furthermore, he got the outstanding research awards from the SeoulTech, 2014. His research interests include Human-centric Ubiquitous Computing, Vehicular Cloud Computing, Information Security, Digital Forensics, Secure Communications, Multimedia Computing, etc. He is a member of the IEEE, IEEE Computer Society, KIPS, and KMMS.