

Article

On the Computation of Concept Stability Based on Maximal Non-Generator for Social Networking Services

Jie Gao ^{1,2}, Fei Hao ^{1,2,*}  and Doo-Soon Park ^{3,*}

¹ Key Laboratory of Modern Teaching Technology, Ministry of Education, Xi'an 710062, China; jaygao@snnu.edu.cn

² School of Computer Science, Shaanxi Normal University, Xi'an 710119, China

³ Department of Computer Software Engineering, Soonchunhyang University, Asan 31538, Korea

* Correspondence: feehao@gmail.com (F.H.); parkds@sch.ac.kr (D.-S.P)

Received: 29 October 2020; Accepted: 30 November 2020; Published: 2 December 2020



Abstract: The concept stability measure under the Formal Concept Analysis (FCA) theory is useful for improving the accuracy of structure identification of social networks. Nevertheless, the stability calculation is an NP-complete task which is the primary challenges in practical. Most existing studies have focused on the approximate estimate to calculate the stability. Therefore, we focus on introducing the Maximal Non-Generator-based Stability Calculation (MNG-SC) algorithm that directly deals with accurate stability calculation to pave the way for FCA's application in structures identification of social networks. Specifically, a novel perspective of stability calculation by linking it to Maximal Non-Generator (MNG) is first provided. Then, the equivalence between maximal non-generator and lower neighbor concept is first proved, which greatly improves scalability and reduces computational complexity. The performed experiments show that the MNG-SC outperforms the pioneering approaches of the literature. Furthermore, a case study of identifying abnormal users in social networks is presented, which demonstrates the effectiveness and potential application of our algorithm.

Keywords: social networks; formal concept analysis; concept stability; maximal non-generator

1. Introduction

With the rapid development of 5G, Internet of Things (IoT) and Artificial Intelligence (AI) in recent years, the degree of integration between the online social network and the entity has further enhanced. In this context, it is required to pay close attention to the human element in the information society. Identifying the cohesive structures, a fundamental task in Social Network Analysis (SNA), facilities to discover valuable hidden patterns and better understand and predict social networks.

Formal Concept Analysis (FCA) [1], a mathematical tool based on lattice theory, is well applied to the research of identifying cohesive structures of the social network [2,3]. Generally speaking, when FCA is used for social network analysis, the topological information of the social network is first used as the input of FCA. Then by the concept generation algorithm, the network structure is transformed into concept form which is extracted knowledge by FCA. The detailed process of how to use FCA in social networks analysis can refer to [2,4]. In practice, the overwhelming size of the extracted knowledge, induced by formal concepts, limits its extensive use. How to understand concepts which represent structural topology is a key issue that affects the recognition accuracy. Considerable work [5,6] has proved that the concept stability measure in the FCA theory community is useful for improving knowledge quality and revealing

interesting patterns, which provides a theoretical basis for improving the recognition accuracy when FCA meets SNA. Indeed, concept stability, a bridge measure between FCA and SNA, is not only efficient for assessing the concept quality around the FCA [6], but also is widely used in numerous applications, e.g., community detection [7,8], virtual machines scheduling in mobile edge computing [9], abnormal users detection in social networks [10]. For example, an illustration of the role of FCA and stability in identifying illegal users in social networks is presented in Figure 1. As shown in Figure 1a, abnormal users may deliver the same fake news to a certain online social network, which makes these users form a cohesive and illegal users group. Indeed, it is hidden common behaviors or characteristics that make these illegal users form relatively cohesive and isolated groups in social networks. Then, FCA can be used to identify various subgraph structures in social networks, but it cannot accurately identify the required structure. However, as shown in Figure 1c, the illegal user group can be accurately identified by using of stability measure.

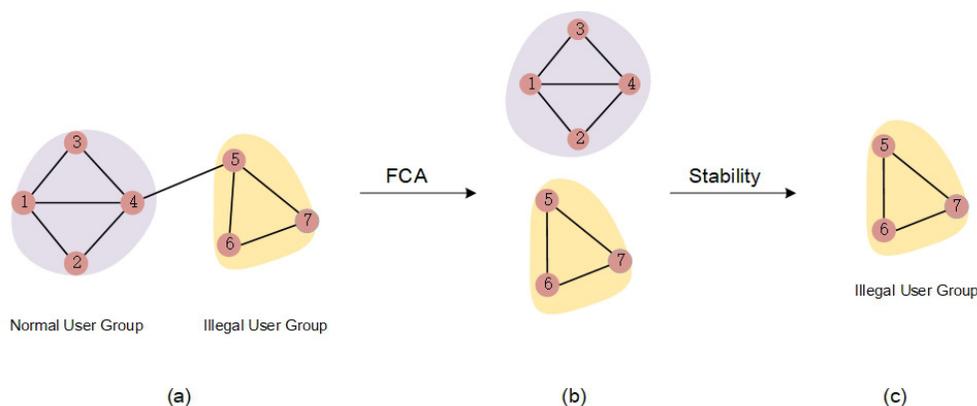


Figure 1. An illustration of the role of FCA and stability in identifying illegal users in social networks. (a) a social network including a normal user group and an illegal user group. (b) the network structure identified by FCA without distinction. (c) an illegal user group accurately identified by stability measure.

Nevertheless, the stability calculation is an NP-complete task [11,12] which is the primary challenges in practical. To this end, in this paper, we focus on stability calculation to pave the way for the downstream tasks of the application of FCA in social network analysis. Although it is shown that the calculation of stability is an NP-complete problem, it is still attracting more and more attention as shown in the recent works [13–20]. The dedicated literature could be roughly divided into two categories: computing approximate or exact values of stability. Most previous approaches rely on the whole structure of the concept lattice to compute approximate or exact values of stability. For example, jay’s algorithm [16] of stability calculation requires browsing the entire concept lattice and summing over the stability already computed in lower levels. Obviously, this not only requires a lot of computing resources but also does not expect scalability when dealing with large concept lattices. A few accurate calculation algorithms, such as DFSP algorithm [18], start from calculating each concept separately and do not make full use of the potential knowledge in the concept lattice to speed up the algorithm. In this paper, an algorithm called MNG-SC that considers the hidden existing knowledge in concept lattice and accurately calculates the exact value of concept stability is developed, which pave the way for the downstream tasks of the application of FCA in social network analysis. Specifically, a novel perspective of stability calculation by linking it to maximal non-generator is first provided. Then the equivalence between maximal non-generator and lower neighbor concept is proved, which makes it sufficient to calculate the concept stability of a concept by only exploring its lower neighbor concepts. Furthermore, the performed experiments show that the MNG-SC outperforms the pioneering approaches of the literature. It demonstrated that MNG-SC algorithm can directly and effectively deal with accurate stability calculation of formal concepts. In addition, a case study

of identifying abnormal users in social networks is presented, which demonstrates the effectiveness and potential application of our algorithm.

In this paper, we focus on solving the theoretical NP-complete problem of stability calculation that hinders the application of FCA to SNA's downstream tasks. The main contributions of this paper are summarized as follows:

- **Equivalence Relation between Maximal Non-generator and Lower Neighbor Concept:** To best of our knowledge, it is the first proved that the equivalence between maximal non-generator and lower neighbor concept. In contrast to the previous algorithm that needs traverse the entire concept lattice to calculate the stability value, the equivalence makes it possible that only exploring the lower neighbor of a formal concept is sufficient to compute its stability, which greatly improves scalability and reduces computational complexity.
- **A Novel Algorithm MNG-SC for Calculating the Exact Stability of Formal Concepts:** Stability calculation is an NP-complete problem. Unlike the approximate calculations of most previous algorithms, we introduce a maximal non-generator-based stability calculation algorithm called MNG-SC for accurately calculating the stability value of the concept.
- **Evaluation:** Experimental results show that our algorithm can directly and effectively deal with accurate stability calculation of formal concepts and it sharply outperforms by many orders of magnitude the pioneering approach of the literature. In addition, it is worth noting that we further present a case study to show the usefulness of our research in improving structural security, such as improving the accuracy of abnormal user identification.

The rest of this paper is organized as follows: Section 2 overviews the related work. The preliminary knowledge and problem statement are provided in Section 3. Section 4 thoroughly describes the MNG-SC algorithm for computing the exact stability values of formal concepts. The experimental results and analysis is presented in Section 5. Finally, Section 6 concludes this paper and discusses future work.

2. Related Work

In this section, the existing work of stability computation is summarized.

The concept lattice theory is hampered by the overwhelming size of the extracted knowledge induced by formal concepts [19]. In addition, the noise contained in the dataset may generate a large number of similar but completely distinct concepts, which seriously affects knowledge quality. Stability measure is more prominent than other interestingness measures for assessing the concept quality and improving the readability of concept lattices. Computing the stability comes down to explore all the power set of the extent of the considered concept and check whether it is a generator [18].

To overcome such an NP-complete problem computation task, the dedicated literature could be roughly divided into two categories: computing approximate or exact values of stability. In the category of calculating approximate values, Kuznetsov et al. [21] used sufficient conditions to select formal concepts with low stability values. Later, Babin and Kuznetsov [15] proposed an approximate calculation algorithm using random Monte Carlo Sampling. Subsequently, Buzmakov et al. [22] provided the lower and upper bounds of stability values which rely on the existence of the concept lattice. Recently, Ibrahim et al. [17] explored and proposed a new set of approximation methods for estimating stability, which are variance reduction techniques that leverage stratification, low-discrepancy and hybridization approaches. In the category of calculating accurate values, Jay et al. [16] presented that computing the concept stability requires browsing all its subconcepts. Later, Zhi [20] studied the inclusion-exclusion principle of the minimal generator to calculate concept stability. It is worth noting that Mouakher et al. [18] pioneered an algorithm called DFSP that handles straightforwardly concepts stability computation by showing that most of this search space could be smartly ignored thanks to the saturation of generators.

Compared with the approximate algorithm, the accurate calculation algorithm can obtain the accurate value of the stability. Jay’s algorithm considers the existing knowledge in the concept lattice, but because it needs to traverse all sub-concepts, its scalability is not well. In contrast, although Zhi’s algorithm does not need to traverse the concept lattice, the process of directly calculating the minimal generator is very complicated, which is also reflected in the experiment section. Although the DFSP algorithm optimizes the process of calculating the minimal generator, it can only calculate one concept at a time, and does not make full use of the potential knowledge in the concept lattice to speed up the algorithm. On the contrary to the previous algorithm, we propose an algorithm called MNG-SC that fully considers the existing knowledge in the concept lattice to reduce computational complexity. The novelty of the MNG-SC algorithm is the proved equivalence between the maximal non-generator and lower neighbor concept, which makes it possible that only exploring its lower neighbor concept is sufficient to compute its stability. It means that our algorithm does not have to traverse all concepts such as Jay’s algorithm, thus ensuring that our algorithm is more scalable. Because the existing knowledge in the concept lattice is fully considered, there is no need to separately design the method of calculating the minimal generator or maximal non-generator (such as Zhi and DFSP), which reduces computational complexity and speeds up the algorithm.

3. Preliminary and Problem Statement

In this section, the key notions used in the remainder of this paper is briefly recalled. In addition, the problem statement is provided at the end.

3.1. Formal Concept Analysis

Formal concept analysis [1] uses a formal context to construct a hierarchy of concepts, more exactly, a concept lattice. A formal context is a triple $K = (G, M, I)$, where G indicates a set of objects, M represents a set of attributes, and $I \subseteq G \times M$ is a binary relation.

Each pair $(g, m) \in I$ expresses that the object $g \in G$ contains the attribute $m \in M$. Given a subset of objects $A \subseteq G$ and a subset of attributes $B \subseteq M$, the following derivation operators are defined:

$$f(A) = \{m \in M | \forall g \in A, (g, m) \in I\}$$

$$g(B) = \{g \in G | \forall m \in B, (g, m) \in I\}$$

where $f(A)$ is the set of attributes common to all objects of A and $g(B)$ is the set of objects sharing all attributes from B . A formal concept is a pair (A, B) , where $A \subseteq G, B \subseteq M$ and $f(A) = B, g(B) = A$. A is called the extent and B is called the intent of the concept (A, B) . Such concepts are then gathered in a hierarchical, lattice-based structure, namely concept lattice. In a concept lattice, a partial order exists between two concepts $(A, B) \leq (C, D)$ if $A \subseteq C (D \subseteq B)$, a pair (A, B) is a subconcept of (C, D) and (C, D) is a superconcept of (A, B) . If $(A, B) \leq (C, D)$ and there is no (X, Y) satisfies $(A, B) \leq (X, Y) \leq (C, D)$, (A, B) is a lower neighbor of (C, D) and (C, D) is an upper neighbor of (A, B) . Each finite lattice has the topmost element with $A = G$, called the top concept, and the lowest element with $B = M$, called the bottom concept.

Example 1. Table 1 shows a formal context K with $G = \{1, 2, 3, 4\}$ and $M = \{a, b, c, d, e\}$, in which \times denotes that there exists the binary relation between an object and an attribute. Since the objects 1, 2 and 3 share the common attributes $\{a, b\}$ and the attributes a and b have the common objects $\{1, 2, 3\}$. Thus, $(\{1, 2, 3\}, \{a, b\})$ is a concept. $\{1, 2, 3\}$ is the extent of the concept, $\{a, b\}$ is the intent of the concept. The corresponding concept lattice is sketched by Figure 2. Each blue node represents a concept. The upper labels and lower labels of the nodes represent intents

and extents of the concepts, respectively. The subconcepts of $(\{3, 4\}, \{c\})$ are $(\{4\}, \{c, d, e\})$, $(\{3\}, \{a, b, c\})$ and $(\{\emptyset\}, \{a, b, c, d, e\})$. The lower neighbor concepts of $(\{3, 4\}, \{c\})$ are $(\{4\}, \{c, d, e\})$ and $(\{3\}, \{a, b, c\})$.

Table 1. Example Formal context K.

K	a	b	c	d	e
1	×	×			×
2	×	×			×
3	×	×	×		
4			×	×	×

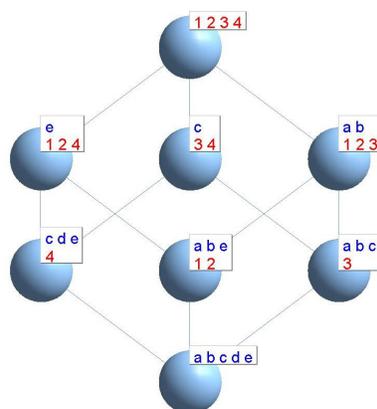


Figure 2. The Concept Lattice of K.

3.2. Concept Stability

Concept lattice straightforwardly illustrates various relationships between concepts and their subconcepts and superconcepts, which is beneficial for knowledge discovery. However, in practice, the effective use of the concept lattice is limited by the large volume of extracted concepts and noise contained in the dataset. Noise contained in the dataset favors the existence of many similar but distinct concepts, which may excessively impair concept readability. Stability is an effective measure used to select the most useful and interesting concepts. The definition of stability is as follows:

Definition 1 ([5] Concept Stability). Let $K = (G, M, I)$ be a formal context. Given a formal concept (A, B) of K , the intentional stability (stability for short) σ of (A, B) is defined as follows:

$$\sigma(A, B) = \frac{|\{C \subseteq A \mid f(C) = B\}|}{2^{|A|}} \tag{1}$$

The stability describes the proportion of subsets of extent whose closure is equal to the intent [5]. It also can be computed by locating and counting the associated generator of A . Calculating the stability of the formal concept will explore all elements of the power sets of the extent to check whether it is a generator. The definition of generator is as follows:

Definition 2 (Generator). Given a formal context $K = (G, M, I)$ and a concept (A, B) of K . If there is a subset P of A satisfy $f(P) = B$, then P is a generator of (A, B) .

Corollary 1. Given a formal concept (A, B) , the stability of (A, B) is the proportion of generator in the power set of A . Therefore, Equation (1) of the stability is simplified as follow:

$$\sigma(A, B) = \frac{|Gen|}{2^{|A|}} \tag{2}$$

where $|Gen|$ is the total number of generators of (A, B) .

Proof. It can be derived naturally from Definition 1 and Definition 2. \square

Example 2. Continue the Example 1, let us consider the formal concept $(\{1, 2, 3\}, \{a, b\})$. The number of the subsets of $\{1, 2, 3\}$ is 8 and only three subsets $\{1, 3\}, \{2, 3\}, \{1, 2, 3\}$ satisfy the Definition 2. Therefore, the number of generators $|Gen|$ is equal to 3 and the stability of $(\{1, 2, 3\}, \{a, b\})$ is equal to 3/8.

3.3. Problem Statement

Computing the intensional stability of a formal concept comes down to explore all the power set of the extent and check whether it is a generator. Formally, given a formal concept $C = (A, B)$, such that $A = \{a_1, a_2, \dots, a_i\}$, computing the intensional stability of this formal concept comes down to explore its power set, i.e., $P(A)$. For each non-empty set, from of $2^{|A|} - 1$, check whether its support is equal to the cardinality of its associated intent, (c.f. Equations (1) and (2)).

In the next section, we will introduce the MNG-SC algorithm to deal with the exact stability calculation of formal concepts. Specifically, we first provide a new perspective of stability computation by linking it to maximal non-generator. Then we prove the equivalence between maximal non-generator and lower neighbor concept, which makes it possible that only exploring the lower neighbor of a formal concept is sufficient to compute its stability.

4. Concept Stability Calculation Based on Maximal Non-Generator

In this section, the MNG-SC algorithm that can accurately calculate the concept stability is introduced in detail. Unlike traditional algorithms that use minimal generators to calculate stability values, we provide a novel perspective of stability calculation by linking it to maximal non-generator. Then thanks to the proved equivalence between the maximal non-generator and the lower neighbor concept, the maximal non-generator is directly obtained from the existing concept lattice. At last, a novel Maximal Non-Generator-based Stability Calculation (MNG-SC) algorithm is proposed.

4.1. Linking Stability Calculation to Maximal Non-Generator

In this part, we first provide a formula (Corollary 2) for calculating stability value leveraging the non-generators. Then, we make use of the special properties (Property 1) of maximal non-generators to obtain a novel perspective (Theorem 1) for calculating the stability of a concept.

Corollary 2. Given a formal concept (A, B) , the stability of (A, B) is the proportion of generator in the power set of A . On the contrary, it also can be computed by the proportion of the total number of non-generator $|NonGen|$. Therefore, Equation (1) of the stability is simplified as follow:

$$\sigma(A, B) = 1 - \frac{|NonGen|}{2^{|A|}} \tag{3}$$

where $|NonGen|$ is the total number of non-generator of (A, B) .

Proof. According to Corollary 1, $|Gen|$ is the total number of (A, B) . Then we have the following equality: $|NonGen| + |Gen| = 2^{|A|}$. Naturally, $\sigma(A, B) = \frac{|Gen|}{2^{|A|}} = \frac{2^{|A|} - |NonGen|}{2^{|A|}} = 1 - \frac{|NonGen|}{2^{|A|}}$. \square

Definition 3 (Maximal Non-generator). Given a formal context $K = (G, M, I)$ and a concept (A, B) of K . If there is a subset P of A which makes $f(P) \neq B$, then P will be a maximal non-generator if there no exists $P_1 \subseteq G$ and $P \subset P_1$ such that $f(P_1) \neq B$.

Property 1 (Property of Definition 3 [18]). Let (A, B) be a formal concept. If P is a maximal non-generator of the extent A , then P fulfills that $\forall P_1 \subseteq P, P_1$ is also a non-generator of A .

Proof. It could be found in [18]. \square

Based on the special Property 1 of maximal non-generator, we have known all subsets of maximal non-generator are non-generators. On this basic, we can naturally calculate the number of subsets of maximal non-generators to obtain the $|NonGen|$ in Equation (3). Therefore, a new theorem for calculating the stability of a concept is proposed. The theorem is as follows:

Theorem 1. Given a formal context $K = (G, M, I)$ and a concept (A, B) of K . If all maximal non-generators of (A, B) are $\{P_1, P_2, P_3 \dots P_n\}$, then

$$\sigma(A, B) = 1 - \left(\sum_{i=1}^n 2^{|P_i|} - \sum_{1 \leq i < j \leq n} 2^{|P_i \cap P_j|} + \dots + (-1)^{n-1} 2^{|P_1 \cap P_2 \cap \dots \cap P_n|} \right) \left(2^{|A|} \right)^{-1} \tag{4}$$

Proof. Since $\{P_1, P_2, P_3 \dots P_n\}$ are the maximal non-generators of (A, B) , based on Property 1, all subsets of P_i are non-generators. Thus, the total number of non-generator is $|NonGen| = |\cup_{i=1}^n \{S | S \subseteq P_i\}|$. Based on Corollary 2 and the Inclusion-Exclusion principle, the total number of non-generators could get via $|\cup_{i=1}^n \{S\}| = \sum_{i=1}^n 2^{|P_i|} - \sum_{1 \leq i < j \leq n} 2^{|P_i \cap P_j|} + \dots + (-1)^{n-1} 2^{|P_1 \cap P_2 \cap \dots \cap P_n|}$. Naturally, Equation (4) can be directly derived. \square

Example 3. Continue the Example 2, we have known the generators of $(\{1, 2, 3\}, \{a, b\})$ are $\{1, 3\}, \{2, 3\}, \{1, 2, 3\}$ and its stability is equal to $3/8$. Obviously, all the subsets of $\{1, 2, 3\}$ except these three generator are non-generators. Thus, the non-generator of $(\{1, 2, 3\}, \{a, b\})$ are $\{1, 2\}, \{1\}, \{2\}, \{3\}, \emptyset$. According to the Definition 3, the maximal non-generators are $\{1, 2\}, \{3\}$.

Suppose we first have know the maximal non-generators $\{1, 2\}, \{3\}$, according the Property 1, the number of subsets of $\{1, 2\}$ and $\{3\}$ is equal to the number of non-generators of the concept. According the Theorem 1, we can get the number of non-generator is 5 and the stability of $(\{1, 2, 3\}, \{a, b\})$ is equal to $1 - \frac{5}{8} = \frac{3}{8}$.

4.2. Extracting Maximal Non-Generator

Summarize the preceding part of this section, we have known that the stability of (A, B) can be calculated by linking it to maximal non-generator. Now, how to obtain all maximal non-generators of (A, B) is becoming important. The brute force method of traversing all subsets of A and checking whether Definition 3 is satisfied undoubtedly wastes computing resources.

Interestingly, we have found the equivalence between maximal non-generator of A and the lower neighbor concept of (A, B) . This equivalence makes it sufficient to calculate the concept stability by only exploring its lower neighbor concepts, which greatly improves scalability and reduces computational complexity. The equivalence theorem is given below:

Theorem 2. Given a formal context $K = (G, M, I)$ and a concept (A, B) of K . The maximal non-generators of A is equivalent to the lower neighbor concepts of (A, B) .

Proof. In order to prove the equivalence of lower neighbor concept and maximal non-generator, we need to prove from two directions: (1) Suppose X is a maximal non-generator of A , then it can form a lower neighbor concept $(X, f(X))$ of (A, B) . (2) If (X, Y) is a lower neighbor concept of (A, B) , then X is a maximal non-generator of A .

(1) Since X is a maximal non-generator of A , then $X \subset A \Leftrightarrow f(A) \subset f(X) \Leftrightarrow g(f(X)) \subset A$. In addition, $X \subseteq g(f(X))$, thus we can get $X \subseteq g(f(X)) \subset A$. However, X is a subset of $g(f(X))$ could never hold, otherwise X could not be maximal. Thus, $X = g(f(X))$, it means that there exists a lower neighbor concept $(X, f(X))$ of (A, B) .

(2) Since (X, Y) is a lower neighbor concept of (A, B) , then $B \subset Y, X \subset A, f(X) = Y$. So X is a non-generator of A . Suppose X is not maximal, there exists a maximal non-generator $\bar{X} \supset X$ of A . From (1), there should exist a concept $(\bar{X}, f(\bar{X}))$. This conclusion contradicts the initial conditions. Thus, X is a maximal non-generator of A . \square

Based on the proved the equivalence of lower neighbor concept and maximal non-generator, we can directly obtain the maximal non-generator form the existing concept lattice. For example, as shown in Figure 3, calculating the maximal non-generators of $\{(1, 2, 3), (a, b)\}$ only needs to directly explore its lower neighbor concepts $\{(1, 2), (a, b, e)\}$ and $\{(3), (a, b, c)\}$.

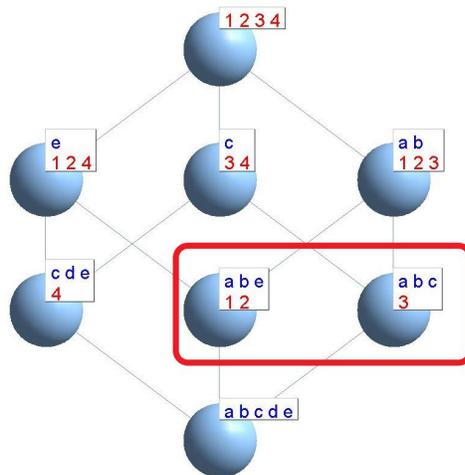


Figure 3. Explored lower neighbor concepts for computing the stability of $\{(1, 2, 3), (a, b)\}$.

Next, on the basis of Theorem 1, the core theorem is proposed as follows. It uses the inclusion-exclusion principle and the lower neighbor concept of (A, B) that represents maximal non-generator of A to calculate concept stability. The core theorem is as follows:

Theorem 3. Let $K = (G, M, I)$ be a formal context and a formal concept (A, B) of K . If all neighbor concepts of (A, B) are $\{(A_1, B_1), (A_2, B_2) \dots (A_n, B_n)\}$, then

$$\sigma(A, B) = 1 - \left(\sum_{i=1}^n 2^{|A_i|} - \sum_{1 \leq i < j \leq n} 2^{|A_i \cap A_j|} + \dots + (-1)^{n-1} 2^{|A_1 \cap A_2 \cap \dots \cap A_n|} \right) (2^{|A|})^{-1} \tag{5}$$

Proof. According to the Theorem 2, all lower neighbor concepts of (A, B) are its maximal non-generators. Then, Equation (5) can be naturally derived from Theorem 1. \square

Corollary 3. Let $K = (G, M, I)$ be a formal context and a formal concept (A, B) of K . If the lower neighbor concept of (A, B) is the bottom concept, the stability of (A, B) is equal to $\frac{2^{|A|}-1}{2^{|A|}}$.

Proof. The extent of the bottom concept is an empty set. So the maximal non-generator is an empty set means only one empty set in the powerset of A is a non-generator. Thus, the stability of (A, B) is equal to $\frac{2^{|A|}-1}{2^{|A|}}$. \square

Example 4. Continue the Example 3, according Theorem 1, we have known how to compute the stability of $\{(1, 2, 3), (a, b)\}$ by using the maximal non-generators. Next, we give a demonstration of how to directly obtain the maximal non-generators according to Theorem 2 and how to calculate the stability by using the obtained maximal non-generators according Theorem 3.

According to Theorem 2, the lower neighbor concepts $\{(1, 2), (a, b, e)\}$, $\{(3), (a, b, c)\}$ are maximal non-generators of $\{(1, 2, 3), (a, b)\}$. As shown in Figure 3, calculating the maximal non-generators of $\{(1, 2, 3), (a, b)\}$ only needs to directly explore its lower neighbor concepts $\{(1, 2), (a, b, e)\}$ and $\{(3), (a, b, c)\}$. Then, according to Theorem 3, the number of distinct subsets of $\{1, 2\}, \{3\}$ which equals to the number of non-generators of $\{(1, 2, 3), (a, b)\}$ is 5. Thus, the stability value of $\{(1, 2, 3), (a, b)\}$ is $\frac{2^3-5}{2^3} = \frac{3}{8}$.

4.3. MNG-SC Algorithm

Based on the previous discussion, the MNG-SC algorithm for the accurate computation of concept stability is proposed. The guiding idea is the fact that the maximal non-generators of a given formal concept is equivalent to its lower neighbor concepts.

The pseudo-code of the MNG-SC algorithm is described by Algorithm 1. The algorithm takes a formal concept (A, B) as input and starts by initializing the stability value σ to 0, the total number of non-generator Nb to 0, the *flag* of the empty maximal non-generator intersection to false (c.f. Line 1). In Lines 3–4, the maximal non-generator objects are assigned to the lower neighbor concepts of (A, B) (c.f. Theorem 2). The goal of the for loop (c.f. Lines 5–13) is to compute the total number of non-generator. To do so, if the *flag* is false, the `getINongenNum` function will be invoked. The goal of the function is to calculate the number of non-generator obtained by i maximal non-generator objects. The flag is true means that the intersection of i or more maximal non-generator is empty. Therefore, Nb is accumulated according to Equation (3). In Lines 14–15, the stability σ of the concept is calculated (c.f. Equation (3)) and returned.

As shown in Algorithm 2, the `getINongenNum` function inputs the maximal non-generator objects, the number of maximal non-generator objects to be intersected and outputs the number of non-generator obtained by i maximal non-generators. It starts by initializing INb to 0 (c.f. Line 2). In the for loop on the i elements of the objects M (c.f. Line 3), the intersection of any i maximal non-generator objects is calculated and the number of non-generators is accumulated and assigned to INb . In Lines 7–9, if the INb is equal to 0, the *flag* represented the empty set of IM is assigned to true. At last, the number of non-generator obtained by i maximal non-generators INb is returned.

Algorithm 1: Maximal Non-generator-based Stability Calculation Algorithm.

Input: A formal concept $C = (A, B)$
Output: Stability σ of concept C

- 1 **Initialize** $\sigma = 0, Nb = 0, flag = false$
- 2 **begin**
- 3 $MaxNonGen = C.lowerNeighbors$
- 4 $N = |MaxNonGen|$
- 5 **for** ($i = 1; i \leq N; i++$) **do**
- 6 **if** $flag == false$ **then**
- 7 $Nb = Nb + \text{getINongenNum}(MaxNonGen, i)$
- 8 **else if** $flag == true \ \&\& \ i\%2 == 0$ **then**
- 9 $Nb = Nb + \binom{N}{i}$
- 10 **else if** $flag == true \ \&\& \ i\%2 \neq 0$ **then**
- 11 $Nb = Nb - \binom{N}{i}$
- 12 **end if**
- 13 **end for**
- 14 $\sigma = 1 - Nb/2^{|A|}$
- 15 **return** σ
- 16 **end**

Algorithm 2: getINongenNum Function.

Input: M : The maximal non-generator objects of A
 i : The number of maximal non-generator objects to be intersected
Output: INb : The number of non-generator obtained by i maximal non-generators

- 1 **begin**
- 2 $INb = 0$
- 3 **for** $M_1, M_2, \dots, M_i \in M$ **do**
- 4 $IM = M_1 \cap M_2 \cap \dots \cap M_i$
- 5 $INb = INb + |IM|$
- 6 **end for**
- 7 **if** $INb == 0$ **then**
- 8 $flag = true$
- 9 **end if**
- 10 **return** INb
- 11 **end**

5. Experiments

The main goal of our experiment evaluation is to investigate the following key questions:

- (Q1): Is MNS-SC more efficient than the state-of-the-art stability calculation algorithms?
- (Q2): Is MNG-SC useful when FCA applied to SNA's downstream tasks, such as abnormal users identification?

In order to solve Q1, we examine the performance of MNG-SC algorithm with compared to the other three state-of-the-art stability calculation algorithms. In order to answer Q2, a case study in Section 5.3 is presented to illustrate the usefulness of MNG-SC algorithm and potential application scenarios of our

research. All experiments are conducted on a Core i7 PC machine with 1.80 GHz 1.99 GHz CPU and 16 GB RAM running Windows 10 system.

5.1. Experimental Setup

5.1.1. Datasets

Four available public datasets commonly used in the network analysis realm were used in these experiments. The critical characteristics of these datasets are summarized in Table 2 including the number of vertices, the number of edges and the generated concept number $\#concepts$ of each dataset. The number of generated concepts by datasets is in different orders of magnitude.

Table 2. Statistics of Datasets.

DataSet	Vertices	Edges	$\#Concepts$
Karate	34	78	134
Terrorist	64	243	165
Football	115	613	3271
Neural	297	2148	17,442

The statics description of the datasets is as follows: Karate (<http://www-personal.umich.edu/>) [23] is a typical social network among the 34 members of the Karate club of a university in the USA. Terrorists (<http://konect.uni-koblenz.de/>) [24] is an undirected network including contacts between suspected terrorists involved in the train bombing of Madrid on 11 March 2004 as reconstructed from newspapers. Football (<http://www-personal.umich.edu/>) [25] is a network of American football games between Division IA colleges during regular season Fall 2000. Neural Network (<http://www-personal.umich.edu/>) [26] is a network representing the neural network of C.Elegans.

A network is represented as a graph $G = (V, E)$ where the vertex set V represents individuals and the edge set E denotes the relationship between individuals. Here, a network can be formalized as the formal context $K = (V, V, I)$ by the following modified adjacency matrix, in which I is the binary relationship between vertices.

Definition 4 ([2] Modified Adjacency Matrix). Let $G = (V, E)$ be a graph with vertices v_1 to v_n . The $n \times n$ matrix M' is called a modified adjacency matrix, in which

$$M' = \begin{cases} a_{ij} = 1 & \text{if } (v_i, v_j) \in E \\ a_{ij} = 1 & \text{if } i = j \\ a_{ij} = 0 & \text{otherwise} \end{cases} \tag{6}$$

5.1.2. Comparison Approaches

Three state-of-the-art stability calculation algorithms were selected as comparisons. All algorithms are all implemented in Java. For each algorithm, we run 20 times and report the average value. The descriptions are as follows.

- **Jay [16]:** it is a stability calculation algorithm that requires browsing the entire concept lattice and summing over the stability already computed in lower levels.
- **DFSP [18]:** it is the first algorithm that does not rely on the structure of concept lattice and handles exact stability computation of formal concepts.

- **Zhi [20]:** it is an algorithm based on the inclusion-exclusion principle of the minimal generator to calculate concept stability.

5.2. Experimental Results

Figure 4 reports the number of generated concepts $\#concepts$ and the running time of Jay, DFSP and MNG-SC algorithm on the tested datasets. As we can see, the performance of both DFSP and our MNG-SC algorithm will degrade with the increase in the number of concepts. At a glance, our algorithm significantly outperforms than its competitors on all considered datasets.

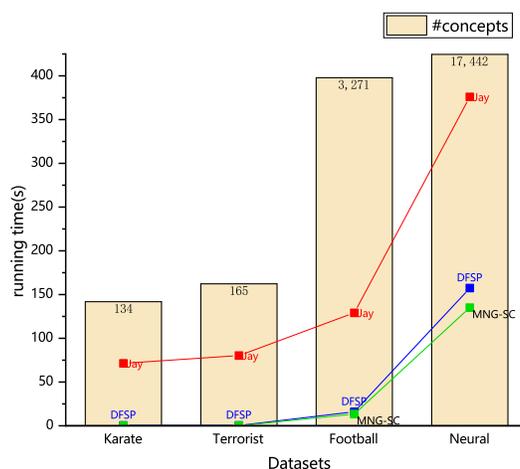


Figure 4. The number of generated concepts and the running time of Jay, DFSP and MNG-SC on the experimental datasets.

Table 3 provides detailed information about a comparison between the MNG-SC algorithm and other algorithms for computing stability in terms of running time. In the second column of the table, the MNG-SC algorithm sharply outperforms Zhi’s algorithm by many orders of magnitude. It is mainly related to the time-consuming process of extracting minimal generators of Zhi’s algorithm. In contrast, MNG-SC can directly obtain maximal non-generator through the proven equivalence between maximal non-generator and lower neighbor concept (c.f. Theorem 2). Besides, “–” means that the Zhi’s algorithm end with memory overflow on the remaining Football and Neural datasets. It is because the extracting minimal generator method of Zhi’s algorithm is sensitive to the number of minimal non-generators. Compared with the third and fourth column, the MNG-SC algorithm also significantly outperforms than Jay’s algorithm and DFSP algorithm. Indeed, because Jay’s algorithm needs to explore the entire concept lattice to calculate the concept stability value, MNG-SC only explores the lower neighbors of the formal concept to calculate the stability. Besides, our algorithm does not need to design a separate method of calculating minimal generators such as DFSP, which reduces the computational complexity. DFSP does not rely on existing knowledge in concept lattice to calculate stability value, which may cause a waste of computing resources, even in the case of small data.

However, it is because our algorithm relies on the lower neighbor concepts of the concept (A, B) to speed up the stability calculation. The more number of the concept’s lower neighbor concept, the more obvious the limitation of our algorithm. Noted that an interesting situation in the experiment that time-consuming concepts have a common feature that the number of their lower neighbor concepts is more than 25. For example, when our algorithm runs on Football dataset, a concept that has 25 lower

neighbors accounts for about 69% of the computation time of all concepts. In addition, another concept that has 24 lower neighbors accounts for 12% of the total time, while the remaining 3269 concepts only account for 19%. However, the number of such time-consuming concepts accounts for a small proportion of the total, and it is easy to limit it by adding judgment conditions.

Table 3. The running time comparison of stability calculation algorithms on different datasets.

DataSet	Zhi(s)	Jay(s)	DFSP(s)	MNG-SC(s)
Karate	124.106	71.483	0.385	0.017
Terrorist	136.500	80.240	0.520	0.026
Football	–	129.135	15.940	13.402
Neural	–	375.970	157.420	135.090

5.3. Abnormal Users Identification of a Social Network: A Case Study

In this section, a case study of identifying abnormal users in social networks is presented to clarify the usefulness and practical application of our algorithm. Noted that our research focuses on providing a powerful stability calculation algorithm for FCA applied to SNA’s downstream tasks.

Aforementioned as Figure 1a, it is some hidden common behaviors or characteristics, such as delivering the same fake news to a certain online social network, that make these illegal users form relatively cohesive and isolated groups in social networks. In this context, abnormal user recognition can be modeled as an isolated maximal clique enumeration problem [10]. In [7], Ibrahim et al. adopted concept stability to develop a community detection algorithm called COIN which based on removing isolated maximal cliques. COIN can be divided into three steps to identify the community based on stability value. Step 1 extracts the concepts that represent cliques and bridges. Step 2 is in charge of using approximated stability values to cut noisy bridges and detect isolated maximal cliques which represent abnormal users groups. The remaining cliques which represent the normal cohesive group are percolated in Step 3. Here, we only use Steps 1,2 of COIN to identify abnormal users called $COIN_{12}$, and develop an optimized algorithm called $COIN_{12}^*$. The only difference between $COIN_{12}$ and $COIN_{12}^*$ is the stability calculation algorithm. $COIN_{12}$ use the approximate calculation algorithm LDS [27] to calculate the stability value, while $COIN_{12}^*$ uses our MNG-SC algorithm. F1-measure is used to evaluate the accuracy of detection. The higher the F1 scores, the higher the accuracy.

Table 4 shows the detailed information about a comparison between $COIN_{12}$ and $COIN_{12}^*$ in terms of running time and F1 score. Obviously, it can be seen that by adopting the stability calculation algorithm proposed in this paper, namely MNG-SC, $COIN_{12}^*$ needs less running time than $COIN_{12}$, but the F1 score of abnormal users detection is higher. For example, on Terrorist (Figure 5), the running time ratio between $COIN_{12}$ and $COIN_{12}^*$ is about $\frac{6}{4}$ and F1 scores ratio is about $\frac{4}{6}$. This is because MNG-SC fully considers the existing knowledge in the concept lattice, reduces the computational complexity and shortens the running time. In addition, the algorithm can accurately calculate the stability value, so the recognition accuracy is higher. In summary, this case study not only shows that MNG-SC algorithm can complete the stability calculation task of FCA efficiently and accurate, and also shows that MNG-SC can provide better support for the downstream tasks of SNA, such as abnormal users identification.

Table 4. The running time and F1-score of $COIN_{12}$ and $COIN_{12}^*$ on the experimental datasets.

Datasets	Running Time(s)		F1-Measure	
	$COIN_{12}$	$COIN_{12}^*$	$COIN_{12}$	$COIN_{12}^*$
Karate	2.687	1.635	0.610	0.886
Terrorist	2.920	1.792	0.591	0.850
Football	58.7321	33.722	0.651	0.835
Neural	385.818	189.813	0.622	0.804

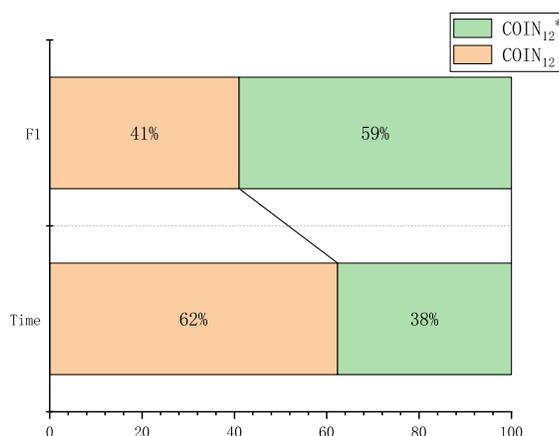


Figure 5. The running time ratio and F1 ratio on Terrorist between $COIN_{12}$ and $COIN_{12}^*$.

6. Conclusions

FCA has been widely used in the identification of social network structures, and stability measures can further improve the accuracy of structure identification. However, stability calculation is an NP-complete task which is the primary challenges in practical. In this paper, we focused on stability calculation and proposed an efficient algorithm MNG-SC to pave the way for the downstream tasks of the application of FCA in social network analysis, such as enhancing the structural security of social networks. Specifically, a novel perspective of stability calculation by linking it to maximal non-generator is first presented. Then, the equivalence between maximal non-generator and the lower neighbor concept is first proved, which makes it sufficient to calculate the concept stability by only exploring its lower neighbor concepts. Experiments demonstrated that the proposed algorithm significantly outperforms several comparative approaches for addressing the concept stability calculation problem and provides better theoretical support for the downstream tasks of SNA, such as abnormal users identification.

Future work for further exploration is as follows: (1) explore the relationship between the minimal generator and maximal non-generator to find the inner connection of stability calculation algorithms. Then consider the special properties of social networks, such as homogeneity, to develop an efficient stability calculation algorithm specifically suitable for social networks. (2) learn the accurate relationship between topological structure and stability value to improve the accuracy of structure recognition. Generally speaking, the more cohesive structure has a higher stable value. (3) conduct more practical applications of stability measure, such as community detection, social network security, etc.

Author Contributions: Conceptualization, J.G. and F.H.; methodology, J.G.; software, J.G.; validation, J.G. and F.H.; formal analysis, F.H.; investigation, J.G.; resources, F.H.; data curation, J.G.; writing—original draft preparation, J.G.; writing—review and editing, F.H. and D.-S.P.; visualization, J.G.; supervision, F.H. and D.-S.P.; project administration, J.G.; funding acquisition, F.H. and D.-S.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the National Natural Science Foundation of China (Grant No. 61702317), the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 840922, the National Research Foundation of Korea (No. NRF-2020R1A2B5B01002134), and the Fund Program for the Scientific Activities of Selected Returned Overseas Professionals in Shaanxi Province (Grant No. 2017024).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

FCA	Formal Concept Analysis
MNG-SC	Maximal Non-Generator Based Stability Calculation Algorithm
MNG	Maximal Non-generator
IoT	Internet of Things
AI	Artificial Intelligence
SNA	Social Network Analysis

References

1. Ganter, B.; Wille, R. *Formal Concept Analysis: Mathematical Foundations*; Springer: Berlin/Heidelberg, Germany, 2012.
2. Hao, F.; Min, G.; Pei, Z.; Park, D.S.; Yang, L.T. K-Clique Community Detection in Social Networks Based on Formal Concept Analysis. *IEEE Syst. J.* **2015**, *11*, 250–259. [[CrossRef](#)]
3. Hao, F.; Pei, Z.; Yang, L.T. Diversified top-k maximal clique detection in Social Internet of Things. *Future Gener. Comput. Syst.* **2020**, *107*, 408–417. [[CrossRef](#)]
4. Hao, F.; Park, D.S.; Pei, Z. When social computing meets soft computing: opportunities and insights. *Hum.-Centric Comput. Inf. Sci.* **2018**, *8*, 8. [[CrossRef](#)]
5. Klimushkin, M.; Obiedkov, S.; Roth, C. Approaches to the selection of relevant concepts in the case of noisy data. In Proceedings of the 8th International Conference on Formal Concept Analysis, Agadir, Morocco, 15–18 March 2010; pp. 255–266.
6. Buzmakov, A.; Kuznetsov, S.O.; Napoli, A. Is concept stability a measure for pattern selection? *Procedia Comput. Sci.* **2014**, *31*, 918–927. [[CrossRef](#)]
7. Ibrahim, M.H.; Missaoui, R.; Messaoudi, A. Detecting communities in social networks using concept interestingness. In Proceedings of the 28th Annual International Conference on Computer Science and Software Engineering, Markham, ON, Canada, 29–31 October 2018; pp. 81–90.
8. Messaoudi, A.; Missaoui, R.; Ibrahim, M.H. Detecting Overlapping Communities in Two-mode Data Networks using Formal Concept Analysis. In *Extraction et Gestion des Connaissances: Actes de la Conférence EGC’2019*; BoD-Books on Demand: Norderstedt, Germany, 2019; Volume 79.
9. Hao, F.; Pang, G.; Pei, Z.; Qin, K.Y.; Zhang, Y.; Wang, X. Virtual machines scheduling in mobile edge computing: a formal concept analysis approach. *IEEE Trans. Sustain. Comput.* **2019**, *5*, 319–328. [[CrossRef](#)]
10. Jie, G.; Fei, H.; Erhe, Y.; Yixuan, Y.; Geyong, M. Concept Stability Based Isolated Maximal Cliques Detection in Dynamic Social Networks. In Proceedings of the 15th International Conference on Green, Pervasive and Cloud Computing, Xi’an, China, 13–15 November 2020; pp. 1–14.
11. Kuznetsov, S. Stability Tablity as an Estimate of Thie Degree of Substantiation of Hypotheses Derived on the Basis of Operational Similarity. *Autom. Documentation Math. Linguist.* **1990**, *12*, 21–29. Available online: <https://economics.hse.ru/data/2012/12/07/1300659254/ADMLv24n6-1990.pdf> (accessed on 1 December 2020).
12. Kuznetsov, S.O. On stability of a formal concept. *Ann. Math. Artif. Intell.* **2007**, *49*, 101–115. [[CrossRef](#)]

13. Kuznetsov, S.O.; Makhalova, T. On interestingness measures of formal concepts. *Inf. Sci.* **2018**, *442*, 202–219. [[CrossRef](#)]
14. Roth, C.; Obiedkov, S.; Kourie, D.G. On succinct representation of knowledge community taxonomies with formal concept analysis. *Int. J. Found. Comput. Sci.* **2008**, *19*, 383–404. [[CrossRef](#)]
15. Babin, M.A.; Kuznetsov, S.O. Approximating concept stability. In Proceedings of the 10th International Conference on Formal Concept Analysis, Leuven, Belgium, 7–10 May 2012; pp. 7–15.
16. Jay, N.; Kohler, F.; Napoli, A. Analysis of social communities with iceberg and stability-based concept lattices. In Proceedings of the 6th International Conference on Formal Concept Analysis, Montreal, QC, Canada, 25–28 February 2008; pp. 258–272.
17. Ibrahim, M.H.; Missaoui, R. Approximating concept stability using variance reduction techniques. *Discret. Appl. Math.* **2020**, *273*, 117–135. [[CrossRef](#)]
18. Mouakher, A.; Yahia, S.B. On the efficient stability computation for the selection of interesting formal concepts. *Inf. Sci.* **2019**, *472*, 15–34. [[CrossRef](#)]
19. Mouakher, A.; Ktayfi, O.; Ben Yahia, S. Scalable computation of the extensional and intensional stability of formal concepts. *Int. J. Gen. Syst.* **2019**, *48*, 1–32. [[CrossRef](#)]
20. Zhi, H.I. On the calculation of formal concept stability. *J. Appl. Math.* **2014**, *2014*, 917639. [[CrossRef](#)]
21. Kuznetsov, S.; Obiedkov, S.; Roth, C. Reducing the representation complexity of lattice-based taxonomies. In Proceedings of the 15th International Conference on Conceptual Structures, Sheffield, UK, 22–27 July 2007; pp. 241–254.
22. Buzmakov, A.; Kuznetsov, S.O.; Napoli, A. Scalable estimates of concept stability. In Proceedings of the 12th International Conference on Formal Concept Analysis, Cluj-Napoca, Romania, 10–13 June 2014; pp. 157–172.
23. Zachary, W.W. An information flow model for conflict and fission in small groups. *J. Anthropol. Res.* **1977**, *33*, 452–473. [[CrossRef](#)]
24. Hayes, B. Connecting the dots. *Am. Sci.* **2006**, *94*, 400–404. [[CrossRef](#)]
25. Girvan, M.; Newman, M.E. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 7821–7826. [[CrossRef](#)] [[PubMed](#)]
26. Watts, D.J.; Strogatz, S.H. Collective dynamics of small-world networks. *Nature* **1998**, *393*, 440. [[CrossRef](#)]
27. Ibrahim, M.H.; Missaoui, R. An Efficient Approximation of Concept Stability Using Low-Discrepancy Sampling. In *Graph-Based Representation and Reasoning*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 24–38.

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).