Knowledge-Based Systems 237 (2022) 107681

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Inferring the densest multi-profiled cross-community for a user

Kamal Taha [a,*], Paul D. Yoo [b], Fatima Zohra Eddinari [c], Siniya Nedunkulathil [d]

[a] Department of Electrical and Computer Science, Khalifa University, Abu Dhabi, United Arab Emirates
[b] Department of Computer Science and Information Systems, Birkbeck College, University of London, UK
[c] Department of Sociology, University of Texas at Arlington, USA
[d] Calicut University, Department of Computer Science, India

ARTICLE INFO

Article history:
Received 19 April 2021
Received in revised form 29 October 2021
Accepted 31 October 2021
Available online 10 December 2021

Keywords:
Community detection
Social network
Cross-community
Multi-profiled cross-community
Social profile

ABSTRACT

The most realistic and widely formed social structures are granular multi-profiled cross-communities constructed from users sharing the traits of common adaptive multi-social profiles. The more important types of such cross-communities that deserve attention are the *densest* holonic ones with various adaptive *multi-social profiles*, because they exhibit many interesting properties. The likelihood of an exact match between an active user and his/her cross-community's interests increases as the cross-community becomes denser. Moreover, the denser such a cross-community is, the more distinct is its interests and the more distinguishable these interests from the ones of other cross communities. Unfortunately, methodologies that emphasize the detection of such granular multi-profiled cross-communities have been understudied. To overcome this, this study proposes a novel methodology that analyzes hierarchically overlapped social profiles to detect the smallest and the most granular multi-profiled cross-communities. The methodology is implemented in a working system named OMACexplorer. The system can detect the *densest* multi-profiled cross-communities from heterogeneous information and social networks. It can also infer an active user's densest multi-profiled cross-community *that matches his/her own social traits*. We evaluated OMACexplorer by comparing it experimentally with eight well-referenced methods. Based on the experiment results, the improvement of OMACexplorer over the other eight methods are 47% and 51% in terms of ARI and F1-score, respectively, which demonstrate the high efficiency and effectiveness of the proposed method.

1. Introduction

An enormous number of complicated scientific problems have been delineated using network representation for empirical studies. These problems belong to various branches of scientific fields, such as ecosystems [1], biological systems [2], scientific citations [3], and information systems [4]. Social media ecosystem problems are the most successfully depicted ones using representational networks for discovering community structures. A society's organization is better understood by clustering it users into community structures based on a specific criterion, such as social attribute. Such community-based clusters can reveal social groups of different sizes, including social media collaborators, research groups, religious, and ethnic-based groups, and colleagues.

Approaches that carry out clustering according to attribute data can be categorizes as follows: (1) approaches that employ nodes' links to guide the clustering procedure [5], yet disregard node attributes that hold important clustering characteristics; (2) approaches that use nodes' attributes to guide the clustering procedure [6], however, overlook important structural relationships between nodes; and (3) approaches that employ both node attributes and link structure to guide the clustering procedure [7], grouping by connectivity density and common attribute similarity. Most network structure-based clustering approaches use probabilistic generative models to identify a community's posterior userships [8,9]. Most of the aforementioned approaches can detect realistic communities based on some real-world properties. However, a large number of them can only detect heterogeneous communities or communities that possess specific topological structures [10,11]. To overcome this, community detection methods were proposed for heterogeneous information networks [12,13]. Most real-world applications contain interacting multi-typed components/objects.

A social community is a unit with certain social rules and commonalities. We observe that most established social communities are formed based on commonalities with the ingredients of adaptive social profiles. A community characterized by an adaptive social profile commonality is a one formed based on

* Corresponding author.
E-mail address: kamal.taha@ku.ac.ae (K. Taha).

https://doi.org/10.1016/j.knosys.2021.107681
0950-7051/

an inherent natural adaptation to some social trait as opposed to being formed by enforcement or involuntary circumstances (*e.g.*, a collegial work group). The most common adaptive social profiles are culture, ethnicity, demography, religion, and the like. That is why this study focuses on such adaptive social profiles. The most realistic and widely formed such social structures are granular multi-profiled cross-communities constructed from users sharing the traits of common adaptive multi-social profiles. The most important types of such cross-communities are the *densest* holonic ones with various adaptive *multi-social profiles*, because they exhibit many interesting properties. For example, such a cross-community can depict a segment of users from a same ethnic background, who follows a same religion, who resides within a same neighborhood, and who belong to a same age-range. The denser a multi-profiled cross-community is, the more granular and holonic it is and the larger the number of its users, whose concerns and interests are manifested in the common concerns and interests of the entire cross-community. That is, the likelihood of an exact match between an active user and his cross-community's interests increases as the cross-community gets denser. Moreover, the denser such a cross-community is, the more distinct is and distinguishable its interests from other cross communities. The concerns and interests of a multi-profiled cross-community are the aggregation of the concerns and interests of the different social profiles (attributes), from which it is constructed.

Belák et al. [14] stated that maximizing the spread of a message across as many communities as possible requires identifying users, who belong to cross maximal communities. High spread of messages corresponds to high diversity or heterogeneity of the communities, to which the cross actors belong [14]. Guo et al. [15] stated that the prediction of human social behavior based on the information of cross many communities is likely to be accurate. Park et al. [16] stated that an effective mechanism for dismantling hacking forums is to identify their dense cross-communities, because they are likely comprised of the prominent hackers who disseminate information to their respective forums. Frank et al. [17] stated that identifying the largest cross forum community, to which a prominent hacking suspect belongs, is an effective mechanism for uncovering the forums, to which the hacker leaks information from the cross-forum. Park et al. [18] stated that the degree of prominence of an individual in a social network can be inferred from the size of cross-community, to which he belongs (*i.e.*, the more communities comprising the cross-community the more prominent is the individual). Guo et al. [15] stated that to obtain a comprehensive understanding and prediction of human social behaviors, we need to study their multi-profiled cross-communities.

Unfortunately, methods that emphasize the detection of such granular multi-profiled cross-communities have been understudied. Current methods may detect multi-profiled communities without regard of their granularities. To overcome this, we propose a novel methodology that analyzes hierarchically overlapped social profiles to detect the smallest and most granular multi-profiled cross-communities for an active user. The methodology is implemented in a working system named OMACexplorer (**O**verlapping **M**ulti-**A**ttribute **C**ommunity **Explorer**). The system can detect the *densest* multi-profiled cross-communities for an active user from heterogeneous social networks. It infers the active user's densest multi-profiled cross-community that matches his/her own social traits. Detecting such cross-communities is important because there are always new users, who seek to join existing established multi-profiled cross-communities that match their own social traits.

To the best of our knowledge, this is the first study that: (1) analyzes hierarchically overlapped social profiles to detect the smallest and most granular multi-profiled cross-communities, and (2) infers the densest multi-profiled cross-community for an active user that matches his own social traits. This study focuses on social profiles that characterize social traits such as culture, ethnicity, demography, religion, or the like. We propose a novel graphical miniature that depicts cross-profiles and their ontological relationships. This modeling technique considers all cross-profiles that come to existence from the interrelations between hierarchically overlapped social profiles. We propose a methodology for extracting the set of prominent keywords from the messages associated with a social group. We also introduce novel representational graphs that depict the relationships between maximal union of k-cliques of users within a same single attributed community. The graph also depicts the interrelationships between maximal union of k-cliques that belong to different attributed communities. The demo application of our system OMACexplorer can be accessed through the following link: http://134.209.27.183/

The remainder of the paper is organized as follows. Section 2 introduces previous work in which inference methods were adopted. Section 3 describes major concepts used in this study as well as an overview of our proposed approach. In Section 4, we describe our proposed graphical miniature, which depicts cross-profiles and their ontological relationships. Section 5 presents our methodology for detecting the densest multi-profiled cross-community, to which an active user belongs. Section 6 compares the experimental results of the proposed methodology with those of other models. Section 7 concludes the paper with an overview of future work.

2. Related work

Many methods adopted unsupervised taxonomy construction. After learning a representation space for the terms, these methods perform clustering to divide terms into various topics. Chang et al. [19] proposed an unsupervised approach based on the Distributional Inclusion Hypothesis [20] to infer hypernym–hyponym pairs. Zhang et al. [21] proposed a method called TaxoGen to detect fine-grained topics using the combination of local-corpus embedding and spherical clustering. Atzori et al. [22] proposed an unsupervised NLP-based method called HyperRank for detecting hypernyms from a corpus. This method employs the cosine distance between words in the vector space and does not require manually-labeled training sets. Gao et al. [23] proposed a neuron-based weakly supervised relation extraction method that depicts each relation as a vector. It was proven to be effective in the extraction of few short relations (FSL). Devlin et al. [24] proposed a contextualized text representation method called BERT, which is based on deep transformation. It learns task-agnostic representations efficiently [25]. Renau et al. [26] proposed an unsupervised hybrid hypernym-detection method that combines lexical patterns, paradigmatic, and syntagmatic measures for detecting hypernymy among nouns. For a given noun, the method keeps detecting the chains of its hypernyms until the most general node in the taxonomy is reached. Singh et al. [27] proposed an unsupervised entity representation method. Each entity is regarded as a probability distribution of its occurrences. Despite the success of these methods, they suffer from the limitation of extracting corpus with many irrelevant terms because of not being able to capture the interests of users in their corpus.

Shen et al. [28] proposed a seed-guided taxonomy construction method called HiExpan. This method adopts depth and width tree expansion of the seed taxonomy by including sibling nodes to the ones that share common parents. It employs the set expansion algorithm [29], which computes the similarity among terms based on the skip-gram features. Dash et al. [30] proposed a neuron-based method based on order embedding. The

method considers the partial orders among words. Shi et al. [31] proposed an adaptive Bayesian NMF method for detecting overlapping communities. He et al. [32] also proposed a method based on NMF that summarizes and detects communities concurrently. The success of these NMF-based methods depends on the appropriateness of the manually selected threshold which determines community memberships.

Huang et al. [33] *proposed a method that produces a complete taxonomy from an input corpus and a seed taxonomy*. It comprises of relation transferring module and concept learning modules. The relation transferring module transfers the learned information downwards and upwards to uncover the first-layer topics and subtopics. The concept learning module improves each concept node's semantics by detecting its discriminative topical clusters and incorporating taxonomy and text joint accordingly. Ye et al. [34] proposed a nonnegative matrix factorization method designed for identifying discrete overlapping communities. This method employs a combination of discriminative pseudo supervision and kernel regression techniques. A node's discrete community usership is identified without a post-processing step.

Lim and Kim [35] proposed a recommendation method based on the emotions manifested in users' tags. The method employs the rating that a user makes for an item as the basic feeling of his/her tag. The unique emotion value of the tag is determined using the emotion dictionary, namely, SenticNet [36]. The weight of the tag is adjusted based on the positive/negative valence of the item. Zafarani and Liu [37] investigated user friendships and popularity across different sites. The authors found that as a user joins different sites, his/her friendships and popularity show certain patterns such as the following: (a) the user's maximum number of friends increases linearly while his/her minimum drops exponentially, (b) the likelihood of getting fewer friends increases, and (c) the user's average popularity converges to a fixed value. De Meo et al. [38] studied the tagging and social networking behavior of users across different Social Sharing platforms. The authors found the following: (a) some of the information embodied in users' profiles depends on the Social Sharing platform, (b) the topic variety of users' profiles differs across Social Sharing platforms, and (c) there are correlations between the intensity of tagging and friending activities. The algorithms that deal with heterogeneous information in multiplex networks can be classified based on the techniques they employ for aggregating the information from the different layers as follows: (a) algorithms that consider layer aggregation (they collapse the layers of a multilayer network into a single layer), (b) algorithms that employ tensor decomposition (they decompose a tensor, which can be regarded as a multidimensional array, to decrease its dimension and maintain the features of the original data [11]), and (c) algorithms that consider collection aggregation (they gather the raw data of all nodes in a sink node and express it as a summary format [39]).

Many approaches have been proposed for inferring communities based on node attribute information. These methods combine attribute information and clustering analysis. Aggarwa et al. [40] proposed a method that detects balanced communities from a heterogeneous network using local succinctness property. The contribution of this study lies in proving the following: inferring community structure based on local behavior is superior to employing a global view, which is due to the challenge of local density variation in constructing a technique for community detection. The authors explored the locality characteristics of social networks and developed an algorithm accordingly based on local behavior. Sun et al. [41] proposed a framework that solves the problem of incomplete node attribute information in heterogeneous information networks. The contribution of this study lies in developing a probabilistic clustering framework for modeling the varying importance of different types of semantic links in heterogeneous information networks with incomplete attributes. Qi et al. [42] proposed a clustering algorithm that integrates the content and structure of networks that depict social media with outlier links. The algorithm is based on heterogeneous random fields. The key contribution of this study lies in combining linkage information and social cues after removing the abnormalities in the linkages to improve the consistency of clustering social media objects. Cruz et al. [43] proposed a method that solves the problems of community detection problems in attributed networks. Its key contribution lies in integrating the two dimensions that describe attributed graphs: the structural dimension (which embodies the social graph) and the compositional dimension (which describes actors).

3. Concepts used in the paper and outline of the approach

3.1. Concepts used in the paper

3.1.1. Concept of Single-Attributed Community (SAC)

We use the term "attribute" throughout the article to refer to an *adaptive social profile* that characterizes some social domain of a group of users within a heterogeneous information network. Even though the methodology proposed in this study is applicable to all types of attributes, we focus on adaptive social profiles that characterize *well-known* and *established* attributes, such as culture, ethnicity, religion, demography, belief, and area of activity. We use the term "Single-Attributed community" (SAC) throughout the article to refer to an aggregation of individual users who share a single adaptive social profile commonality (i.e., a community with a single attribute). This concept is formalized in Definition 1.

Definition 1. Single-Attributed Community (SAC): SAC is a group G of users within a social network (V, E), where V is a (finite) set of vertices, and E is a (finite) set of edges linking the vertices, with schema (R, L), where each $x, y \in G (x \neq y)$ shares a single attribute mapping $\psi : V \rightarrow R$ and relation mapping $\partial : E \rightarrow L$.

3.1.2. Concept of Maximal K-Clique Sub-SAC (MKCSS)

An effective mechanism for identifying the influential nodes in a SAC is to first represent it using k-clique model. A k-clique is a defined as complete graph with k nodes (e.g., users). A pair of adjacent k-cliques shares $k-1$ nodes. We now formalize the concepts of clique and k-clique in Definition 2.

Definition 2. Clique: A clique ç of users in a graph $\mathbb{G}$ is a subset of the nodes of $\mathbb{G}$ such that every two nodes (i.e., users) in ç are adjacent. Thus, ç is a complete induced subgraph. A k-clique is a clique of a sub-community that has k nodes. Each two adjacent k-cliques $ç_i$ and $ç_j$ in the sub-community share $k-1$ users (i.e., nodes). That is, $ç_i \bigcap ç_j = k-1$.

We introduce the concept of **M**aximal **K-C**lique **S**ub-**S**AC (**MKCSS**). A MKCSS is a sub-community of users within a SAC formed from the maximal union of k-clique of users within the SAC, where each two k-cliques in the sub-community is k-clique connected. It is fully connected (complete) sub-community of k-cliques within a SAC. For example, Fig. 1b shows 11 MKCSSs. We now formalize this concept in Definition 3.

Definition 3. Maximal k-clique Sub-SAC (MKCSS): It is a maximal union of k-cliques of users within a SAC, where each pair of k-cliques $ç_i$ and $ç_j$ in the sub-communsity is k-clique connected. $ç_i$ and $ç_j$ are k-clique connected, if there is a series of k-cliques $ç_x, \ldots, ç_y$, such that each two adjacent cliques in the series $ç_i, ç_x, \ldots, ç_y, ç_j$ share $k-1$ nodes.

Lemma 1. *Any two k-cliques in a MKCSS are k-clique connected.*

Proof. If we consider the scenario of $|MKCSS| = k + 1$, any two k-cliques ς_1^k and ς_2^k in the MKCSS share $k + 1$ nodes. This is because $\varsigma_i^k \cap \varsigma_2^k = k + 1$. Similarly, if we consider the scenario of $|MKCSS| = k+2$, any two k-cliques ς_1^{k+1} and ς_2^{k+1} in the MKCSS share k + 1. Since: (1) ς_i^k is connected to any k-clique in ς_1^{k+1}, and (2) ς_2^k is connected to any k-clique in ς_2^{k+1}, ς_i^k and ς_2^k are k-clique connected. The above holds for any $|MKCSS| > k$.

3.1.3. Concept of MKCSS relationship graph

We introduce the concept of **M**KCSS **R**elationship **G**raph to depict the relationships between the MKCSSs of a same SAC as well the interrelationships between the MKCSSs that belong to different SACs. Each node in the MKCSS Relationship Graph represents a MKCSS. Two nodes in the MKCSS Relationship Graph are connected by an edge if they share at least one node. For example, Fig. 1c shows a MKCSS Relationship Graph constructed based on the MKCSSs in Fig. 1b. Definition 4 formalizes the MKCSS Relationship Graph concept.

Definition 4. MKCSS Relationship Graph: It is an undirected graph $\mathbb{G}(V, E)$, where V is the set of nodes and E is the set of edges in the graph. Each node in $\mathbb{G}$ represents a MKCSS in some SAC. The weight of a MKCSS node is the number of its unique k-cliques. For example, the weight of node 1 in Fig. 1c is 5, which is the number of its k-cliques in Fig. 1b. Two nodes $n_i, n_j \in V$ are linked by an edge $e \in E$, if n_i and n_j share at least one common user.

Theorem 1. *Since the weight of a MKCSS node is the number of its unique k-cliques, the weight of the MKCSS node equals MKCSS! (k! (MKCSS - k)!).*

Proof. The number of unique k-cliques in a MKCSS node is the number of unique k-combination of a subset of k distinct nodes that belong to the MKCSS node. The number of these k-combinations equals the binomial coefficient, which can be depicted using factorial: MKCSS! (k! (MKCSS - k)!).

3.1.4. Concept of association edge

It is an edge that depicts the Cross-Users of two Interrelated Cross-MKCSSs Mi and Mj (Mi and Mj belong to two different SACs). It connects nodes Mi and Mj in the MKCSS Relationship Graph to depict that the two nodes share Cross-Users. For example, there are 4 Association Edges in the MKCSS Relationship Graph in Fig. 2c: edges (8, 2), (8, 3), (6, 4), and (5, 4).

***Running Example 1*:** Consider the fragment of social media messaging network in Fig. 1a. Table 1 shows the seven SACs depicted in Fig. 1a and their attributes. The following figures are constructed accordingly:

- Fig. 1b depicts the MKCSSs of only two of the seven SACs: SACs NBHD$(\mathfrak{y}_y)$ and ETH (ϱ_x). Fig. 1b is constructed based on the 4-clique modeling. NBHD$(\mathfrak{y}_y)$ consists of MKCSSs 5–11 and ETH(ϱ_x) consists of MKCSSs 1–4. NBHD$(\mathfrak{y}_y)$ and ETH(ϱ_x) share user members 8, 9, 11, 12, and 13 (marked in black).
- Fig. 1c shows the MKCSS Relationship Graph that corresponds to the MKCSSs in Fig. 1b. Node i in Fig. 1c represents MKCSS i in Fig. 1b. For example, node 1 in Fig. 1c represents MKCSS 1 in Fig. 1b. The weight of node i (i.e., w(i)) in Fig. 1c is the numbers of 4-clique subsets of four distinct nodes in node i in Fig. 1b. An edge connecting two MKCSSs in Fig. 1c signifies that the two MKCSSs share at least one cross-user in the corresponding Fig. 1b.

3.2. Outline of the approach

First, messages associated with explicitly declared SACs are collected. This can be social media ecosystem messages (*e.g.*, Facebook messages) associated with explicitly declared SACs. We model the overlapping multi-attributed communities resulting from the overlapping of the different SACs by incorporating and expanding the concept of *Property Graph* [44]. We call our Property Graph-based graphical model an **O**verlapping **M**ulti-**A**ttributed **C**ommunities **G**raph (**OMACGraph**). It graphically depicts overlapping multi-attributed communities and their hierarchical relationships. In an OMACGraph, each SAC is depicted by a node. An overlapping multi-attributed community came to existence at level i of the graph from the overlapping of a set of SACs {§i, §j, …} at level $i-1$. The ontological relationship between the node {§i, §j, …} and each of the nodes in the set is represented by an edge connecting them. An OMACGraph considers all the overlapping multi-attributed communities that come to existence from the interrelations between SACs of different attributes.

An OMACGraph $\mathbb{G}$ is a tuple $(V, E, \rho, \lambda, \sigma)$, where: (a) V is a finite set of nodes, each representing an overlapping multi-Attributed community $\ddot{o}$, which is a set of SACs of different attributes. If $\ddot{o}$ is located at a hierarchical level $\mathbb{L}$, it will consist of at least $\mathbb{L}$ SACs, (b) E is a finite set of edges. Each pair of overlapping multi-Attributed communities $\ddot{o}_i$ and $\ddot{o}_j$ shares at least one common SAC. The pair is linked by an edge. E is formalized as follows:

$$E = \{\text{edge}(\varsigma_i, \varsigma_j) : \varsigma_i, \varsigma_j \in V; \varsigma_i \cap \varsigma_j \neq 0; \varsigma_i \text{ lies at level } L \text{ and } \varsigma_j \text{ lies at level } L+1 \text{ in OMACGraph}\}$$

We outline below the sequential steps taken by OMACexplorer (See Appendix A for the algorithm):

1. *Constructing the OMACGraph (Section 4):*
   - ***(a)*** *Constructing level zero of the OMACGraph:* Each SAC is depicted by a node and placed at level zero of the graph. Each SAC is represented by the prominent buzzwords extracted from its messages in a preprocessing step. That is, prominent buzzwords are extracted from messages in a preprocessing step before applying the proposed system. The system extracts the buzzwords using the technique described in the third and fourth paragraphs of Section 5.1 (see Fig. 3).
   - ***(b)*** *Constructing level one of the OMACGraph (Section 4.1):* A multi-profiled cross-community node {Ṡ, Ṩ} comes to existence at level 1 of the OMACGraph, if SACs Ṡ and Ṩ at level 0 are "related". Ṡ and Ṩ at level 0 are "related", if they have at least one pair of Interrelated Cross-MKCSSs, whose Cross-Users are strongly associated with both Ṡ and Ṩ. The extent to which the Cross-Users of Ṡ and Ṩ are associated with the two SACs is determined by the "*influence*" of the Association Edge that connects the pair of Interrelated Cross-MKCSSs containing the Cross-Users. The hierarchical ontological relationships between the node {Ṡ, Ṩ} at level 1 and nodes Ṡ and Ṩ at level 0 are depicted by paths (i.e., edges) that originate from Ṡ and Ṩ and converge at level 1 to create the new multi-profiled cross-community.
   - ***(c)*** *Constructing the remaining levels of OMACGraph (Section 4.2):* Let $\dot{S}$ be the set of cross-community nodes at level ℓ. The system converges each subset $\acute{s} \subseteq \dot{S}$ at level $(\ell + 1)$ to form a new cross-community node if: (1) there exists one or more common SAC in each node $\in \acute{s}$, and, (2) $\acute{s}$ does not contain more than one SAC with the same attribute (Section 8.1).

Table 1
Descriptions of the seven SACs and four attributes used in our running example. Each SAC is annotated with its attribute.

SAC	Attribute	Description	SAC	Attribute	Description
NBHD (η_x), NBHD (η_y)	Neighborhood-based	Users who reside in neighborhoods NBHD(η_x) and NBHD(η_y), respectively	ETH(ϱ_x), ETH(ϱ_y)	Ethnicity-based	Descendant of ethnicities ETH(ϱ_x) and ETH(ϱ_y), respectively
RLG (r_x), RLG (r_y)	Religion-based	Users who follow religions RLG(r_x) and RLG(r_y), respectively	ORG(o_x)	Region-based	Users from national origin ORG(o_x)

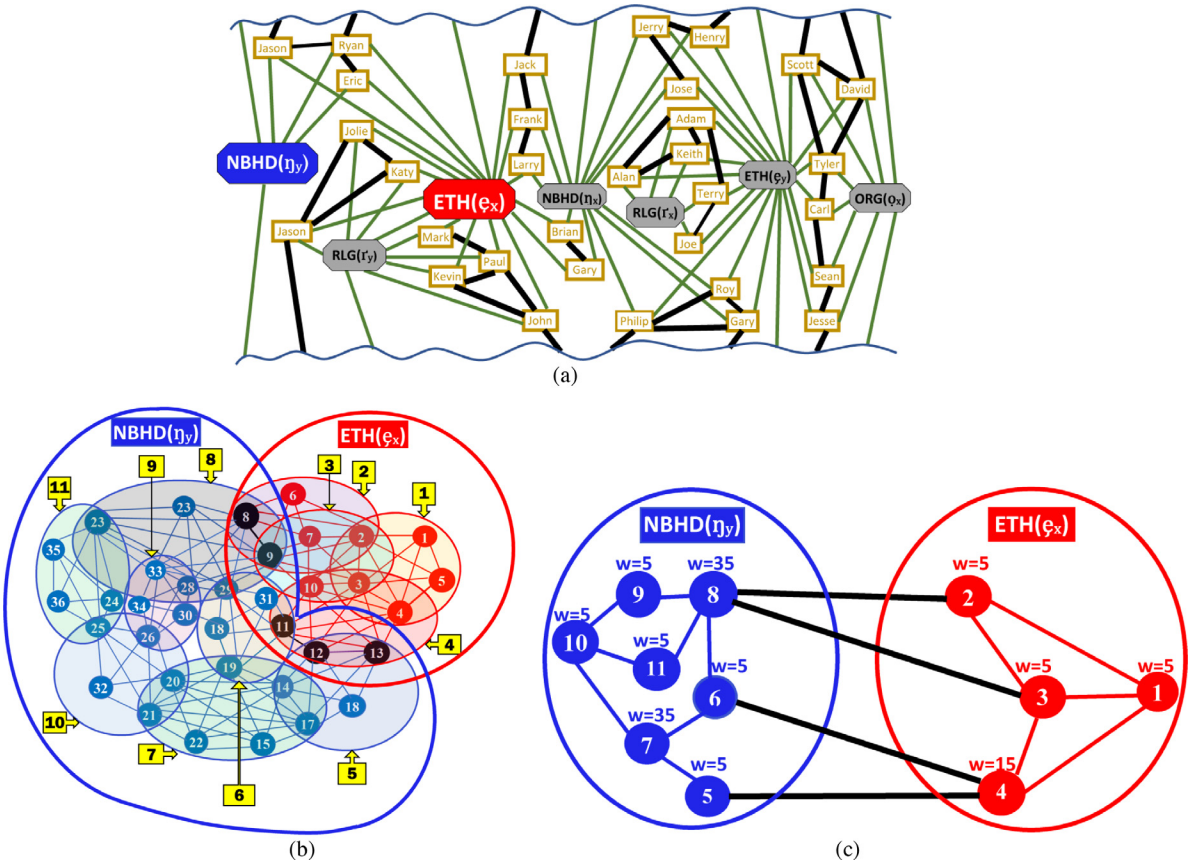

Fig. 1. (a) A fragment of social media messaging network, (b) the MKCSSs of SACs NBHD(η_y) and ETH(ϱ_x) constructed by applying 4-clique modeling on the structure of the network in Fig. 1a, and (c) the MKCSS Relationship Graph constructed based on the MKCSSs of SACs NBHD(η_y) and ETH(ϱ_x) shown in Fig. 1b.

2. *Identifying the densest cross-community to which an active user belongs (Section 5):*

   (a) *Identifying the SACs at Level 0 that Match the Active User's Own Social Traits (Section 5.1):* First, the dominant keywords in the messages associated with the user are extracted. These keywords are matched against the keywords of the SACs. The matching SACs represent the active user's SACs.

   (b) *Identifying the Densest Multi-Profiled Cross-Community to which the Active User Belongs (Section 5.2):* The system locates the convergence of the longest paths that originate from a new user's known SACs at level 0. The cross-community resulting from this convergence represents the densest multi-profiled cross-community, to which the active user should be assigned.

For convenient reference, we list in Table 2 the major concepts and abbreviations used in the article.

Table 2
Description of the major concepts and abbreviations used in the article.

Concept/Abbreviation	Description
SAC	Single-Attributed Community
OMACGraph	Overlapping Multi-Attributed Communities Graph
MKCSS	Maximal *k*-clique Sub-SAC
MKCSS Relationship Graph	It is a graph that depicts the relationships between MKCSSs. Each node represents a MKCSS. Two MKCSS nodes are connected by an edge if they share at least one user member.
Association edge	It is an edge that represents the Cross-Users of two cross-MKCSSs. It connects two Interrelated Cross-MKCSS nodes in the MKCSS Relationship Graph to depict their Cross-Users association.

4. Constructing the OMACGraph

Level zero of the OMACGraph is constructed by depicting each SAC by a node and placing it on the top layer of the graph. Thus, level zero at the top layer of the OMACGraph comprised of the

set of nodes representing the SACs. Each SAC is labeled with an abbreviation of the SAC's attribute (e.g., adaptive social profile). Each SAC is represented by the set of prominent buzzwords extracted from the messages associated with its members.

4.1. Constructing level one of the OMACGraph

A multi-profiled cross-community node $\{\dot{S}, \dot{S}\}$ comes to existence at level 1 of the OMACGraph, if SACs $\dot{S}$ and $\dot{S}$ at level 0 are "related". The two SACs $\dot{S}$ and $\dot{S}$ at level 0 are "related", if they share at least one pair of interrelated cross-MKCSSs, whose cross-users are strongly associated with both $\dot{S}$ and $\dot{S}$. The hierarchical ontological relationships between the node $\{\dot{S}, \dot{S}\}$ at level 1 and nodes $\dot{S}$ and $\dot{S}$ at level 0 are depicted by paths (*i.e.*, edges) originating from $\dot{S}$ and $\dot{S}$ and converge at level 1 to create the new multi-profiled cross-community node $\{\dot{S}, \dot{S}\}$. The system determines the extent to which the cross-users of $\dot{S}$ and $\dot{S}$ are associated with the two SACs based on the "influence" of the Association Edge connecting the interrelated cross-MKCSSs. For example, consider Fig. 1b and 1c. The extent to which the cross-users 8 and 9 of the interrelated cross-MKCSSs 2 and 8 (see Fig. 1b) are associated with SACs NBHD$(\mathfrak{y}_y)$ and {ETH(ϱ_x) is determined based on the "influence" of the Association Edge (2, 8), which connects the interrelated cross-MKCSSs 2 and 8 (see Fig. 1c). We aim at quantifying the influence of each Association Edge by assigning it a score that reflects its degree of influence. SACs $\dot{S}$ and $\dot{S}$ will be considered related if the score of at least one of their Association Edges is greater than a heuristically determined threshold.

Below describes our mechanism for quantifying the degree of influence of an Association Edge. Consider that a node n that belongs to a SAC $\dot{S}$ needs to send a message to a node $\acute{n}$ residing in another SAC $\dot{S}$. Consider that there are more than one Association Edge connecting $\dot{S}$ and $\dot{S}$. From these Association Edges, intuitively, the message will be sent through the one, whose path to n and $\acute{n}$ is the shortest. Consider that there are $\mathscr{P}$ shortest paths from n to all the Association Edges. Consider that P out of the $\mathscr{P}$ paths leads to the Association Edge e. The chance of e to pass the message increases as the ratio $\mathrm{P}/\mathscr{P}$ increases. Thus, as this ratio increases, the chance of e increases *at the expense* of the chances of the other Association Edges. Therefore, the *influence* of e can be assessed by its average chance of passing the messages sent by all nodes in $\dot{S}$ to the nodes in $\dot{S}$. As an example, consider Fig. 1c and that node 5 in SAC NBHD $(\mathfrak{y}_y)$ needs to send a message to some node in SAC RLG$(\mathfrak{r}_x)$. There are four Association Edges that interrelate NBHD$(\mathfrak{y}_y)$ and RLG$(\mathfrak{r}_x)$: edges (8, 2), (8, 3), (6, 4), and (5, 4). The shortest path from node 5 to RLG$(\mathfrak{r}_x)$ contains 2 edges. There are 6 shortest paths with 2 edges from node 5 to the four Association Edges, as follows: 2 paths to edge (8, 2); 2 paths to edge (8, 3); 1 path to edge (6, 4), and 1 path to edge (5, 4). The chance of each of the four Association Edges in passing the message is: 2/6 for edge (8, 2), 2/6 for edge (8, 3), 1/6 for edge (6, 4), and 1/6 for edge (5, 4).

Based on the above observations, we heuristically constructed the formula shown in Eq. (1) to compute the influence score $\mathfrak{t}_v^u$ of an Association Edge (u, v). Let $\mathfrak{y}$ be the number of edges in the shortest path from some node n to the edge (u, v) in the MKCSS Relationship Graph. Let edge $(\ddot{u}, \ddot{v})$ be another Association Edge in the MKCSS Relationship Graph, where $\ddot{u}$ resides in the same SAC of u and $\ddot{v}$ resides in the same SAC of v. We constructed the formula in such a way that the score $\mathfrak{t}_v^u$ keeps increasing as the following ratio gets higher: (1) the multiplication of: (a) the number of paths with $\mathfrak{y}$ edges from n to the edge (u, v), and (b) the sum of the weights of u and v, and (2) the multiplication of: (a) the number of paths with $\mathfrak{y}$ edges from n to each other edge $(\ddot{u}, \ddot{v})$, and (b) the sum of the weights of $\ddot{u}$ and $\ddot{v}$.

$$\mathfrak{t}_v^u = \frac{\sum_{n \in \mathrm{MRG}} \mathfrak{C}_{u,v}^n}{\tilde{N}} \tag{1}$$

$$\mathfrak{C}_{u,v}^n = \frac{\sigma_{u,v}^{\mathfrak{y}} \times (\mathfrak{w}(u) + \mathfrak{w}(v))}{\sigma_{\ddot{u},\ddot{v}}^{\mathfrak{y}} \times \sum (\mathfrak{w}(\ddot{u}) + \mathfrak{w}(\ddot{v}))}$$

- $\mathfrak{t}_v^u$: The influence score of an Association Edge (u, v) in the MKCSS Relationship Graph.
- $\mathfrak{C}_{u,v}^n$: The chance of a message sent by some MKCSS node n to pass through the Association Edge (u, v).
- $\tilde{N}$: Number of MKCSS nodes in the MKCSS Relationship Graph, whose $\mathfrak{C}_{u,v}^n > 0$.
- $\mathfrak{y}$: Number of edges in the shortest path from n to the Association Edge (u, v).
- $\sigma_{u,v}^{\mathfrak{y}}$: Number of paths with $\mathfrak{y}$ edges from n to the Association Edge (u, v).
- $\ddot{u}$ and $\ddot{v}$: Nodes at the end point of some other Association Edge $(\ddot{u}, \ddot{v})$ in the MKCSS Relationship Graph.
- $\sigma_{\ddot{u},\ddot{v}}^{\mathfrak{y}}$: Number of paths with $\mathfrak{y}$ edges from n to Association Edge $(\ddot{u}, \ddot{v})$.
- $\mathfrak{w}(u)$, $\mathfrak{w}(v)$, $\mathfrak{w}(\ddot{u})$, $\mathfrak{w}(\ddot{v})$: The weights of nodes u, v, $\ddot{u}$ and $\ddot{v}$ respectively.

Theorem 2. *Since the weight $\mathfrak{w}(u)$ of a MKCSS node u at the end point of an Association Edge (u, v) is the set of k-cliques that includes each user within u and each user within the set $|u \cap v|$, the weight $\mathfrak{w}(u)$ can be computed as:*

$$\mathfrak{w}(u) = \frac{\left|\cup_{\Re \epsilon |u \cap v|} \Pi_{\Re}^{u}\right| \times k}{|u|}$$

where $\prod_{\Re}^{u}$ is the number of k-cliques that includes each user in u and each user $\Re$.

Proof. Each individual user who belongs to a MKCSS $\mathcal{M}$ appears in all the k-cliques formed from all possible $k-1$ combinations of the set $\dot{S}$ of individual users, who belong to $\mathcal{M}$. The number of these k-cliques equals the fraction $k/|\dot{S}|$ of overall number of k-cliques.

As the value of $\mathfrak{y}$ gets larger, the overall relative chance of the Association Edge (u, v) in passing the flow of information from node n to the nodes in the other SAC decreases. This is because as $\mathfrak{y}$ gets larger, the relative chance of each other Association Edges $(\ddot{u}, \ddot{v})$ in passing the flow of information from n increases at the expense of the edge (u, v). We revised Eq. (1) to accommodate the impact of $\mathfrak{y}$ as shown in Eq. (2).

$$\mathfrak{C}_{u,v}^n = \left(\frac{\sigma_{u,v}^{\mathfrak{y}} \times (\mathfrak{w}(u) + \mathfrak{w}(v))}{\sigma_{u,v}^{\mathfrak{y}} \times \sum (\mathfrak{w}(\ddot{u}) + \mathfrak{w}(\ddot{v}))} \right)^{\mathfrak{y}} \tag{2}$$

We need to discriminate between large and small variation in the range of $\mathfrak{y}$. Towards this, we employ logarithm to account for the variations of $\mathfrak{y}$ by increasing $\mathfrak{C}_{u,v}^n$ *invertly* as $\mathfrak{y}$ increases. We revised Eq. (2) accordingly as Eq. (3) shows.

$$\mathfrak{C}_{u,v}^n = \left(\frac{\sigma_{u,v}^{\mathfrak{y}} \times (\mathfrak{w}(u) + \mathfrak{w}(v))}{\sigma_{u,v}^{\mathfrak{y}} \times \sum (\mathfrak{w}(\ddot{u}) + \mathfrak{w}(\ddot{v}))} \right)^{\log_2 \mathfrak{y}} \tag{3}$$

If the influence score of $\mathfrak{t}_v^u$ an Association Edge (u, v) between SACs $\dot{S}$ and $\dot{S}$ is greater than a threshold β, u and v are strongly associated. If there is at least one Association Edge, whose influence score $\mathfrak{t}_v^u > \beta$, $\dot{S}$ and $\dot{S}$ are considered "related".

4.2. Constructing the remaining levels of the OMACGraph

The system constructs the remaining levels of the OMACGraph as follows. To create new nodes at some hierarchical level $\ell + 1$, the system first enumerates all distinct combinations of multi-profiled cross-community nodes at level ℓ. Consider that $\acute{S}$ is the set of combinations at level ℓ. The system will create paths that originate from each subset $\acute{s} \subseteq \acute{S}$ and converges the paths at level $\ell + 1$ to form a new multi-profiled cross-community node, if the following conditions are met: (1) there exists at least one common SAC shared by the cross-community nodes in the subset $\acute{s}$, and, (2) $\acute{s}$ does not contain more than one SAC with the same attribute. The new convergence cross-community node is depicted by the set of SACs composing $\acute{s}$. This process terminates when no further new multi-profiled cross-community nodes can be created at a higher hierarchical level.

Running Example 2: Consider Fig. 1a, 1b, and 1c of our running example. We demonstrate below how the Association Edges' influence scores are computed and how the OMACGraph is constructed:

- Fig. 2a shows the influence scores of the Association Edges (2, 8), (3, 8), (4, 6), and (4, 5) after applying Eq. (3).
- Fig. 2b shows a fragment of the OMACGraph after adding level 1 to level 0 of the graph.
- Fig. 2c shows the complete OMACGraph. The paths that originate from two nodes at level ℓ converge at level $(\ell + 1)$ to form a new multi-profiled cross-community node, if the two nodes: (1) share at least one common SAC, and (2) do not include two or more SACs that have the same attribute. In Fig. 2c, for example, the convergence multi-profiled cross-community node $\{\text{ETH}(\varrho_y), \text{RLG}(\mathfrak{r}_y), \text{NBHD}(\mathfrak{y}_x)\}$ at level 2 represents a fraction of users of the same ethnic group $\text{ETH}(\varrho_y)$, who follows the same religion $\text{RLG}(\mathfrak{r}_y)$, and resides in the same neighborhood $\text{NBHD}(\mathfrak{y}_x)$.

5. Identifying the densest cross-community to which an active user belongs

5.1. Identifying the SACs at level 0 that match the active user's own social traits

Each SAC at level zero is represented by the prominent keywords extracted from the messages associated with the users who belong to the SAC. This set will serve as a distinguishing characteristic of the SAC. It will be used for determining an active user's SACs, as follows. The active user's messages will be fetched for occurrences of the prominent keywords of each SAC. The user will be considered to belong to the SAC(s), whose prominent keywords have significance occurrences in the messages associated with this user.

The system extracts a set of buzzwords from the messages associated with each SAC and the messages associated the active user. The system filters these buzzwords to keep only the prominent ones to act as representatives of the SAC/user. It does so by assigning a score to each keyword to reflect its dominance status relative to the other buzzword. A buzzword is a word that occurs *frequently* in many messages associated with a SAC or an active user. There are several available tools that provide lists of buzzwords specific to profiling entities (*e.g.*, disciplines, professions, adaptive social profiles, etc.) [45]. In the framework of OMACexplorer, the list of buzzwords specific to a profiling entity under consideration is uploaded to the system from such tools. OMACexplorer would fetch the messages associated with each SAC/active user to extract the buzzwords that match the buzzwords uploaded from the tool. Then, OMACexplorer will filter these extracted buzzwords to determine the *dominant* ones (the ones that occur *frequently* in a *large number* of messages associated with the SAC/active user).

OMACexplorer identifies the dominant buzzwords as follows. It assigns a pairwise *looses* and *beats* indicator to each keyword by constructing a loose-beat table. The rationale behind the loose-beat table is that it is an effective and efficient mechanism for identifying the subset of keywords with the highest occurrences in the largest number of messages. Such a subset will contain the dominant keywords that can uniquely characterize and distinguish a SAC, whose messages contain these keywords, from other SACs. That is why the system uses these dominant keywords for determining to which SAC(s) a new user belongs by fetching his messages for occurrences of the dominant keywords of each SAC. The loose-beat table is constructed as follows. The table's entries are $(\mathfrak{k}_i, \mathfrak{k}_j)$, where $\mathfrak{k}_i$ and $\mathfrak{k}_j$ denote keywords i and j respectively. Let: "the *number of times* that the number of occurrences of $\mathfrak{k}_i$ is higher than that of $\mathfrak{k}_j$ is greater than "the *number of times* that the number of occurrences of $\mathfrak{k}_j$ is higher than that of $\mathfrak{k}_i$" in the messages associated with a specific MKCSS. If this is the case, the entry $(\mathfrak{k}_i, \mathfrak{k}_j)$ in the table will be given the symbol "+". If not, it will be given the symbol "-". If they are equal, entry $(\mathfrak{k}_i, \mathfrak{k}_j)$ will be assigned the indicator symbol "0". Definition 5 formalizes the concept of pairwise score:

Definition 5. A keyword's pairwise score: Let $(\mathfrak{k}_i > \mathfrak{k}_j)$ denote that the number of incidents where the number of occurrences of $\mathfrak{k}_i$ is greater than that of $\mathfrak{k}_j$ in the messages associated with a specific MKCSS. The pairwise score of the keyword $\mathfrak{k}_i$ equals the following:

$$\left|\left\{\mathfrak{k}_j \in \mathfrak{k}_m : \mathfrak{k}_i > \mathfrak{k}_j\right\}\right| - \left|\left\{\mathfrak{k}_j \in \mathfrak{k}_m : \mathfrak{k}_j > \mathfrak{k}_i\right\}\right|$$

where $\mathfrak{k}_m$ is the set of keywords in the MKCSS's messages.

To this end, the keyword $\mathfrak{k}_i$ will be assigned a dominance score S, which is determined as follows. The dominance score $S_{\mathfrak{k}_i}$ of the keyword $\mathfrak{k}_i$ is characterized as shown in Eq. (4):

$$S_{\mathfrak{k}_i} = N_{beat} - N_{lost} \tag{4}$$

where N_{beat} is the number of times that $\mathfrak{k}_i$ has the highest occurrences among all keywords in the set of messages and N_{lost} is the number of times that $\mathfrak{k}_i$ does not have the highest occurrences.

If we sum the dominance scores of all keywords, we will find that the result is zero. The largest score is $(\mathfrak{y} - 1)$ and the smallest score is $-(\mathfrak{y} - 1)$, where $\mathfrak{y}$ is the number of keywords. Then, each of the candidate keywords is assigned a normalized score $\bar{S}$. A candidate keyword is considered prominent if its normalized score is greater than a threshold β. As shown in Eq. (5), β is a value that is less than the normalized score's mean by the standard error's mean.

$$\beta = \frac{1 - \sqrt{\sum_{\forall \mathfrak{k}_{j \in \mathbb{K}_\mathbb{M}}} \left(\bar{S}_{\mathfrak{k}_j} - \frac{1}{|\mathbb{K}_\mathbb{M}|}\right)^2}}{|\mathbb{K}_\mathbb{M}|} \tag{5}$$

Running Example 3: Fig. 3a shows hypothetical 10 keywords occurring in 3 messages associated with SAC $\textbf{NBHD}(\mathfrak{y}_y)$ in our running example (recall Table 1). Fig. 3b shows the corresponding dominance score S and normalized score $\bar{S}$ of each keyword computed based on its number of occurrences shown in Fig. 3a.

5.2. Identifying the densest multi-profiled cross-community to which the active user belongs

We describe in this section the methodology adopted by the system for inferring the *densest* multi-profiled cross-community, to which an active user should be assigned. The methodology

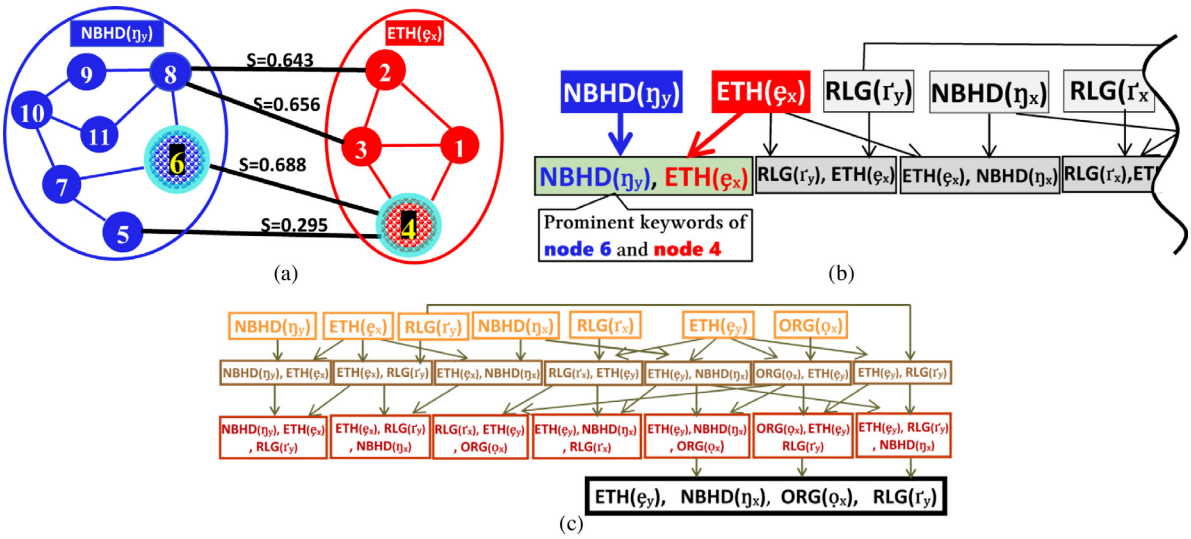

Fig. 2. (a) The influence scores of the Association Edges (2, 8), (3, 8), (4, 6), and (4, 5) after applying Eq. (3), (b) a fragment of OMACGraph after adding level 1 to level 0 of the graph, and (c) the complete OMACGraph.

	$\mathbb{K}_1$	$\mathbb{K}_2$	$\mathbb{K}_3$	$\mathbb{K}_4$	$\mathbb{K}_5$	$\mathbb{K}_6$	$\mathbb{K}_7$	$\mathbb{K}_8$	$\mathbb{K}_9$	$\mathbb{K}_{10}$
$\mathfrak{m}_1$	2	4	0	0	3	0	1	2	4	1
$\mathfrak{m}_2$	0	2	2	3	4	2	0	0	1	3
$\mathfrak{m}_3$	3	5	1	4	5	2	3	1	2	0

(a)

Keyword	$\mathbb{K}_1$	$\mathbb{K}_2$	$\mathbb{K}_3$	$\mathbb{K}_4$	$\mathbb{K}_5$	$\mathbb{K}_6$	$\mathbb{K}_7$	$\mathbb{K}_8$	$\mathbb{K}_9$	$\mathbb{K}_{10}$
$\mathbb{K}_1$	0	+	-	+	+	-	-	-	+	-
$\mathbb{K}_2$	-	0	-	-	0	-	-	-	-	-
$\mathbb{K}_3$	+	+	0	+	+	+	+	0	+	+
$\mathbb{K}_4$	-	+	-	0	+	-	-	-	-	0
$\mathbb{K}_5$	-	0	-	-	0	+	-	-	-	-
$\mathbb{K}_6$	+	+	-	+	-	0	+	-	0	+
$\mathbb{K}_7$	+	+	-	+	+	-	0	0	+	0
$\mathbb{K}_8$	+	+	0	+	+	+	0	0	+	-
$\mathbb{K}_9$	-	+	-	+	+	0	-	-	0	-
$\mathbb{K}_{10}$	+	+	-	0	+	-	0	+	+	0
S	+1	+8	-8	+4	+6	-2	-3	-5	+2	-3
$\bar{S}$	0.11	0.20	0	0.15	0.17	0.08	0.06	0.04	0.13	0.06
Dominant keywords	$\mathbb{K}_1$	$\mathbb{K}_2$		$\mathbb{K}_4$	$\mathbb{K}_5$				$\mathbb{K}_9$	

(b)

Fig. 3. (a) 10 keywords ($\mathbb{K}_1 - \mathbb{K}_{10}$) occurring in 3 messages ($\mathfrak{m}_1 - \mathfrak{m}_3$) that belong to the users of SAC **NBHD**($\mathfrak{y}_y$) of our running example (recall Table 1), and (b) the pairwise scores, dominance score S and normalized score $\bar{S}$ of each of the 10 keywords in Fig. 3a, calculated based on their *looses* and *beats* indicators.

assigns an active user to the densest and most granular multi-profiled cross-community that matches his/her own social traits. It does so by navigating the paths of OMACGraph that originate from the active user's SACs at level zero of the graph. Below are the processing steps taken by OMACexplorer to locate the user's densest cross-community:

(a) Marking the SAC nodes at level 0 of OMACGraph, to which the active user belongs. The system considers that an active user belongs to a certain SAC, if his/her messages include a significant number of buzzwords, whose *ontological concepts* fall under the prominent buzzwords of this SAC. Let $\mathbb{C}_{\dot{S}} \longmapsto \mathbb{C}_{\mathbb{K}}$ denote: the ontological concepts of some set $\mathbb{K}$ of buzzwords that fall under the ontological concepts of the prominent buzzwords of some SAC $\dot{S}$. The system fetches the messages of the active user to identify the following for each SAC node $\dot{S}$ al level 0: $\mathbb{C}_{\dot{S}} \longmapsto \mathbb{C}_{\mathbb{K}}$. The matching SACs will be marked. For example, let d be a SAC node. The dominant buzzwords of d will be considered the root ontology of d. After the system extracts the buzzwords from the messages associated with the active user (recall the second paragraph of Section 5.1), it will fetch these buzzwords for the ones, whose ontological concepts fall under the root ontology of d. If a significant number of the user's buzzwords fall under the root ontology of d, d will be marked as one of the SACs, to which the active user belongs. For example, consider that the term "entertainer" is one of the dominant buzzwords of d. Fig. 4 shows a sample of the ontological concepts of this term. The ontologies are determined in a preprocessing step before applying the proposed system. To facilitate ontology building, OMACexplorer is constructed on top of Stanford CoreNLP [46] (to derive the linguistic annotations for text) and Protégé [47] (for building the ontologies).

(b) Traversing the paths originating from the marked SAC nodes at level 0. This will lead to locating the convergence of the *longest* paths originated from the marked SAC nodes at level 0. The node located by this convergence represents the *densest and most granular* multi-profiled cross-community, to which the new user should be assigned.

If the longest paths originated from n marked nodes at level 0, the new user's densest multi-profiled cross-community will consist of more than n SACs. That is, if the number of initially identified SACs for the user at level 0 is n, the densest multi-profiled cross-community of the user will consist of more than n SACs. These extra SACs are ***implicity*** identified based on the interrelations between the different cross-communities in OMACGraph. This is one of the key advantages of our proposed system OMACexplorer,

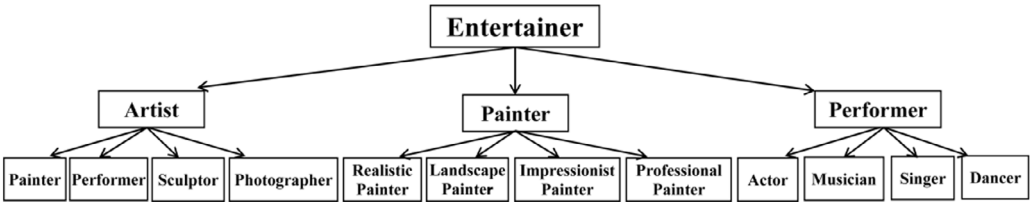

Fig. 4. A sample of the ontological concepts of the term "entertainer".

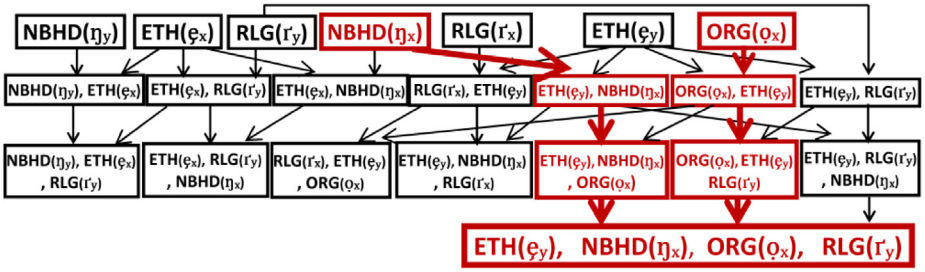

Fig. 5. The convergence of the longest paths originated from SAC nodes ORG(o_x) and NBHD(n_x) at level 0, to which the active user described in Example 4 belongs. The *densest* cross-profiled cross-community of this user is {**ETH** $(\mathbf{e_y})$, **NBHD** $(\mathrm{n}_\mathbf{x})$, **ORG** $(\mathrm{o}_\mathbf{x})$, **RLG**$(\mathrm{r}_\mathbf{y})$}.

since the messages associated with an active user may not necessarily include buzzwords referring directly (*i.e.*, explicitly) to these extra SACs.

Running Example 4: Consider our running OMACGraph shown in Fig. 5. Now, let us identify the *densest* cross-profiled cross-community, to which an active user ų should be assigned. Consider that the system inferred neighborhood NBHD(n_x) and national origin ORG(o_x) as the SAC nodes at level 0, to which ų belongs. The system will create paths originating from NBHD(n_x) and ORG(o_x) at level 0 and locate the longest convergence of these paths. As Fig. 5 shows, the convergence node is {**ETH** $(\mathbf{e_y})$, **NBHD** $(\mathrm{n}_\mathbf{x})$, **ORG** $(\mathrm{o}_\mathbf{x})$, **RLG**$(\mathrm{r}_\mathbf{y})$}, which represents the fraction of users of the same ethnic group ETH(e_y), who reside in the same neighborhood NBHD(n_x), who are descendants of a matching national origin ORG(o_x), and who follow the same religion RLG(r_y). As such, it can be observed that the paths initiated from only *two* SAC nodes and that the densest multi-profiled cross-community node of user ų contains *four* SAC nodes. That is, the system could infer SACs RLG(r_y) and ETH(e_y) ***implicitly*** based on the interrelations between the different overlapping cross-communities in the OMACGraph.

6. Experimental results

6.1. Baseline methods

Our proposed model emphasizes the analysis of the hierarchical interrelationships of communities for uncovering the granularities of their cross-communities and detecting the smallest holonic cross-profiled cross-community. As demonstrated by the example graph shown in Fig. 5, our model can detect cross-communities in a hierarchically granular fashion (*e.g.*, level 0, level 1, level, 2, ….). Unfortunately, we could not find a baseline model that detects granular cross-communities to compare our method with. Moreover, we could not find a baseline model that can detect the densest cross-profiled cross-community to compare our method with. The baseline models that we selected for comparing our method with can detect cross-profiles without regard to their granularities. With reference to Fig. 5, these baseline models can detect up to level 1 of the graph (*i.e.*, levels 0 and 1). Moreover, the ground-truth datasets we used for the evaluation (*i.e.*, Facebook Social circles, DBLP, and com-Friendster) are not designed for evaluating the granularity of detected cross-communities. As a result, we compared our method with the selected baseline models to evaluate its accuracy for detecting communities at only levels 0 and 1 of the OMACGraph. That is, we could only evaluate levels 0 and 1 of OMACGraph and could not evaluate the remaining levels of the OMACGraph.

We selected eight baseline models that employ node attribute information for detecting communities to compare our method OMACexplorer with. We describe below each of the eight models:

1. ***CoRel:*** Huang et al. [33] proposed a method called CoRel for constructing a seed-guided topical taxonomy. The method produces a complete taxonomy from an input corpus and a seed taxonomy. It incorporates the following two modules: (1) relation transferring module, which analyzes the relationships between the parent–child nodes of the seed taxonomy and transfers the learned information downwards and upwards to uncover the first-layer topics and subtopics, and (2) concept learning module, which enhances each concept node's semantics by detecting its discriminative topical clusters and incorporating taxonomy and text joint accordingly.
2. ***DNMF:*** Ye et al. [34] proposed a nonnegative matrix factorization method called DNMF. It is designed for identifying discrete overlapping communities. It adopts a combination of discriminative pseudo supervision and kernel regression techniques. Anode's discrete community usership is identified without a post-processing step.
3. ***LOCD:*** Ni et al. [48] proposed a method called LOCD for identifying local overlapping communities. It adopts a bottom-up intermediary score maximization procedure. From a set of nodes belonging to at least two communities, the method selects a subset as seed nodes. The method detects the community of each seed node. If the fuzzy relation between a seed node and a given node is large enough, the two nodes are considered to belong to a same community.

4. ***Neo4j:*** This is a graph database engine that implements the Property Graph model [49]. It allows name-value pairs properties to be attached to nodes and relationships. It employs labels to represent a node's roles in a graph. A node can have multiple labels and an attribute property can have one value. The method employs path-based graph operations. It uses Cypher [49] as its native declarative graph query language. Cypher employs "patterns" as its main building blocks and path-oriented graph mechanism for its database operations. It maps a query's variables to a graph based on the patterns in the graph. It matches the variables against the graph. It outputs variables representing the maximum number of labels and property values relevant to the query's input variables.
5. ***RCF:*** This method was proposed by Guesmi et al. [50] for detecting communities from a heterogeneous information network using relational concept analysis. It builds a set of lattices called Concept Lattice Family (CLF). It employs an iterative procedure that generates a set of concept lattices at each iteration. As an input, it takes a query path (qb). Each qp_i is a pair $[(qp_s, L_s), (qp_t, L_t)]$, where L_s (source lattice), L_t (target lattice) $\subset$ CLF. After navigating through the lattices, the method outputs the detected communities. Query navigation begins by dealing with all the concepts of the source lattice L_s to identify the corresponding concepts of the initial query path. The result of the query consists of the iteratively generated set of concepts related to qp_s.
6. ***GKS:*** Sharma et al. [51] proposed a method called Generalized Katz Score (GKS) that employs the concept of group accretion inspired by dyadic link prediction (DLP) techniques. It is an extension of the Katz approach [52]. It outputs a score that captures the degree of possibility an external actor can be absorbed in a specific group. The score is the average proximity of the different actors in the group to the external actor. The method enumerates the paths of a network and uses an unsupervised path counting procedure inspired by DLP.
7. ***BRWS:*** Sharma et al. [51] proposed a method called Birandom Walk Score (BRWS) that employs the concept of group accretion inspired by DLP techniques. It employs a network alignment procedure to quantify the affinity between actors based on their both internal and external links, using a semi-supervised learning method. It analyzes cycles to learn the affinity of a group of users to an external actor. It uncovers the cycles that pass through each group of nodes and other groups of nodes.
8. ***GLPS:*** Sharma et al. [51] proposed a method called Group Label Propagation Score (GLPS) that employs the concept of group accretion inspired by DLP techniques. It uses a semi-supervised procedure based on techniques associated with hypergraph label propagation. The approach diffuses labels by random walks. Upon the stabilization of the random walks, each external node's final label is regarded as its affinity score to the given group.

The key differences between GKS, BRWS, and GLPS can be summarized as follows:

1. While GKS measures the affinity of each group of users to an external actor separately, BRWS measures the affinity of a subgroup within a group to an external actor (*e.g.*, the external actor is known by multiple users in the group). So, the GKS approach employs an incremental accretion procedure, which refers to the incremental joining of external actors to an existing group, while the BRWS approach employs a subgroup accretion procedure, which refers to the collaboration of external actors with a subset of an existing group.
2. While the GKS and BRWS approaches consider paths and cycles over a *network of actors*, the GLPS approach considers a *network of groups* (NOG) by employing label propagation score. The score is based on the hypergraph structure of the NOG.

The codes of the above methods are available as follows:

- The code of CoRel [33] is available at https://github.com/teapot123/CoRel.
- The code of LOCD [48] is available at https://github.com/ahollocou/multicom.
- The code for DNMF [34] is available at https://github.com/smartyfh/DNMF.
- As for GKS, GLPS, and BRWS, we used the same dataset and followed the same experimental setup employed for evaluating the three methods as described in Sharma et al. [51].

6.2. Evaluation setup

We implemented OMACexplorer in Java, ran it in Intel(R) Core(TM) i7-6820HQ processor with 32 GB RAM and 2.70 GHz CPU under Windows 10 Pro. We used a version of OMACexplorer that builds level one of OMACGraph based on interrelated MKCSSs. The demo application of our system OMACexplorer can be accessed through the following link: http://134.209.27.183/

We considered each community in a dataset as a SAC. Let S be the set of communities in a dataset. For each different subset $\left(\acute{S} \subset S\right)$, we computed the intersection of the communities in $\acute{S}$ to obtain the subset η of nodes resulted from the intersection. We aim at using η as a ground-truth cross-community of the communities in $\acute{S}$ (*i.e.*, the overlapping of the communities in $\acute{S}$ resulted the cross-community η). That is, we evaluated the accuracy of each method for detecting the cross-community resulting from the overlapping of each $\left(\acute{S} \subset S\right)$. We compared the results of each method with the subset η. The same procedure is repeated for each $\left(\acute{S} \subset S\right)$.

We compared the methods in terms of F1-score and Adjusted Rand Index (ARI). F1-score is the harmonic average of recall and precision. It is defined as: F1-score $=$ 2 (Precision $\times$ Recall) / (Precision $+$ Recall). ARI calculates the expected similarity of all pair-wise comparisons between a pair of clusters. We used it to compute the pair-wise comparisons between a community detected by a method and the corresponding ground-truth community. It is defined as shown in the formula below: ARI $=$ ((Index–Expected index) / (Maximum index–Expected index))

$$\text{where: Index} = \sum_{ij} \binom{n_{ij}}{2},$$

$$\text{Expected index} = \left[\sum_{i} \binom{a_i}{2} \sum_{j} \binom{b_j}{2}\right] / \binom{n}{2},$$

$$\text{and Max index} = \frac{1}{2}\left[\sum_{i} \binom{a_j}{2} + \sum_{j} \binom{b_j}{2}\right]$$

6.3. Identifying OMACexplorer's parameters values that achieve best accuracy results

In this test, we aim at determining the values for the OMACexplorer's parameters shown in Table 3 that achieve the best accuracy results. We ran OMACexplorer under various values of parameters k, β, and β against the DBLP, Friendster, and Facebook Social Circles datasets. Fig. 6 plots the accuracy results in terms of

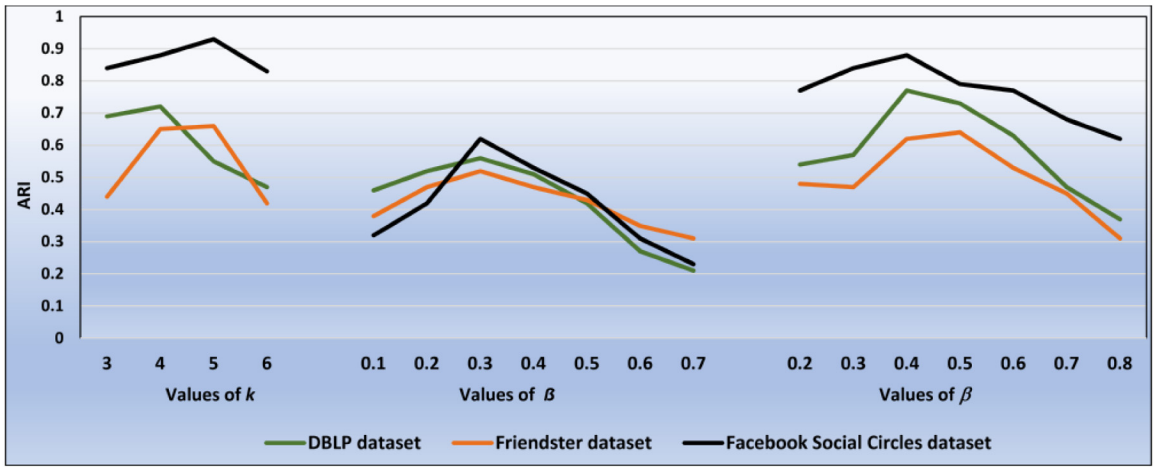

Fig. 6. The average accuracy of OMACexplorer in terms of ARI under different values of the parameters using the DBLP, Friendster, and Facebook Social Circles datasets.

Table 3
The parameters employed by OMACexplorer.

Parameter	Description of the parameter
k	The value of k in k-clique
β	The Association Edge influence score threshold that determines the degree of association between the MKCSSs at the end points of the Association Edge (recall Eq. (1)).
β	The normalized score threshold that determines the prominence of a candidate keyword (recall Eq. (5))

ARI under the different values of k, β, and β. Based on the results, we determined that the values that achieved best accuracy results are as follows: 0.44 for k, 0.28 for β, and 0.37 for β. We used these values for processing OMACexplorer in the experiments, whose results are reported in Sections 6.4–6.7.

We observed from the experimental results that the values for the three parameters have impact on the accuracy of OMACexplorer. We observed that the accuracy kept improving as the value of each of the three parameters increased up to a certain value and then it kept declining. After investigating this phenomenon, we observed that increasing the values of the three parameters led to the following:

1. Enhancing the MKCSSs. As the values of the parameters increased, the MKCSSs kept retaining only strongly associated nodes and discarded other nodes.
2. Increasing the percentage of nodes that became non-member of any MKCSS. As the values increased, the constraint for a node to be retained in an MKCSS increased.

Thus, observation 1 increases the accuracy while observation 2 decreases the accuracy. The accuracy kept improving (observation 1) until a point where the percentage of non-member nodes (observation 2) became large enough to degrade the accuracy. That is, the accuracy degraded as the percentage of non-member nodes increased.

6.4. Evaluating the accuracies of the methods for detecting the DBLP communities

In this evaluation, we used the DBLP Citation Network Dataset (DBLP-Citation-network V11) [53–55] compiled by the AMiner.org project (http://www.arnetminer.org/developer#introduction). The original data was extracted from the DBLP compiled by the Stanford Network Analysis Project (SNAP) [56]. Besides the information from DBLP, the DBLP-Citation-network incorporates the abstracts of papers and the citation relationship between the papers. Specifically, each paper is associated with an abstract and title, authors, year, and venue. Authors who published in a certain conference or journal form a community. Each group participated in a publication venue is considered a ground-truth community. In our experiments, we considered each of these communities as a SAC. We represented each SAC by the set of prominent buzzwords extracted from the abstracts of the papers associated with the SAC. The DBLP-Citation-network V11 consists of 4,107,340 papers and 36,624,464 citation relationships. The networks consist of 317,080 nodes, 1,049,866 edges, and 13,477 communities. We downloaded the DBLP-Citation-network V11 from (https://www.aminer.org/citation).

To apply the ontology on the DBLP Citation Network Dataset, we carried out the following. We imported the Computer Science Ontology v.3.2 (https://cso.kmi.open.ac.uk/downloads) to Protégé [47] (recall Section 5.2). A cross-community node {Ṡ, Ṣ} comes to existence at level 1 of the OMACGraph, if there is a significant number of the common dominant buzzwords representing both SACs Ṡ and Ṣ fall under the same root ontologies (i.e., there is a significant number of the common dominant buzzwords extracted from the abstracts associated with the two SACs fall under the same root ontologies). As an example, decision support systems, expert systems, intelligent robots, and knowledge based systems fall under the same "artificial intelligence" ontology. Table 4 shows the accuracies of the methods for determining the DBLP communities in terms of ARI. Table 5 shows the accuracies of the methods for determining the DBLP communities in terms of F1-score.

We also compared the methods by following the same experimental setup described in [51]. These include:

1. The training and test periods for the main splits. Each split was marked with a fixed end year. Papers published between 2004–2007 are used for training and papers from 2008 to 2010 are used for testing (see Table 5).
2. Using metrics "Precision@N_{top} (IA)" and Recall@N_{top} (IA) for $N_{top} = 100$, against the DBLP dataset.

Using the divisions of the dataset presented in Table 6, Table 7 lists the accuracy of the methods. The values in Table 7 that

Table 4
The accuracies of the nine methods for detecting the DBLP communities in terms of ARI.

	LOCD	DNMF	CoRel	RCF	GKS	GLPS	BRWS	Neo4j	OMDCexplorer
Minimum	0.41	0.28	**0.49**	0.24	**0.14**	0.21	0.26	0.43	**0.47**
Average	0.54	0.42	**0.61**	0.34	**0.19**	0.33	0.31	0.5	**0.63**
Maximum	0.67	0.56	**0.72**	0.6	**0.28**	0.47	0.52	0.63	**0.79**

Table 5
The accuracies of the nine methods for detecting the DBLP communities in terms of F1-score.

	LOCD	DNMF	CoRel	RCF	GKS	GLPS	BRWS	Neo4j	OMDCexplorer
Minimum	0.16	0.09	**0.18**	0.07	**0.022**	0.11	0.063	0.14	**0.23**
Average	0.21	0.12	**0.23**	0.109	**0.04**	0.123	0.073	0.2	**0.27**
Maximum	0.26	0.14	**0.29**	0.12	**0.08**	0.18	0.13	0.25	**0.36**

In each row, the highest value is highlighted in blue color and each bolded value with gray background is either the highest, second highest, or lowest value.

Table 6
Dividing the dataset into testing and training splits.

Boundary year	Split no.	Train	Test
2007	Main split	2004–2007	2008–2010

belong to BRWS, GKS, and GLPS are also listed in [51]. The metrics are defined as follows:

$$\text{Precision@N}_{top}\ (\text{IA}) = (\text{Number of correctly predicted groups using IA from top–N}_{top}\ \text{list})/\text{N}_{top}$$

$$\text{Recall@N}_{top}\ (\text{IA}) = (\text{Number of correctly predicted collaborations using IA from top–N}_{top}\ \text{list}) / (\text{Number of actual IA groups})$$

where IA is the incremental accretion, N_{top} is the N top sorted IA set, and Top-N_{top} is the largest score in N top sorted IA set.

6.5. Evaluating the accuracies of the methods for detecting the facebook social circles

In this evaluation, we used the Facebook Social circles compiled by the SNAP [56] as a ground-truth dataset. The Facebook Social circles dataset was compiled by surveying 4039 Facebook users. The dataset includes Ego networks and a human social network. Each of the 4039 users is represented by a node. Each of these nodes has its own Ego network, which includes the list of friends (*i.e.*, circle) of the user. The Ego networks are constructed as follows: (1) each of the 4039 nodes is considered a focal node *(i.e.*, ego), (2) each other node *(i.e.*, an alter) is linked with the focal node by an edge, if the two have some type of social relationship s (*e.g.*, share similar interests), (3) each alter node has its own Ego network, and (4) the interrelationships among all ego networks are represented by a human social network. This network contains 88,234 edges.

The Facebook Social circles dataset also includes the profile of each user. That is, the dataset includes profiles (node features). In the experiments, we represented each SAC by the set of prominent buzzwords extracted from the profiles of the users associated with the SAC. To apply the ontology on the Facebook Social circles dataset, we imported the Socially Interconnected Online Communities (SIOC) ontology (http://rdfs.org/sioc/ns#) to Protégé [47]. The ontology is compiled by the SIOC project (http://sioc-project.org/). For the sake of the evaluation, we considered the Ego networks as ground-truth communities. For evaluating our system OMACexplorer, a ground-truth OMACGraph is constructed from the Ego networks, as follows. Each Ego network is depicted by a node at level zero of OMACGraph. We used a version of OMACexplorer that builds level one of OMACGraph based on SACs' interrelated MKCSSs. The interrelationships between the overlapping communities at level i are represented by new overlapping communities at level $i + 1$ of the graph.

For the testing part, we randomly selected 25% of the users of each Ego network and evaluated the accuracy of OMACexplorer for identifying the cross-profiled cross-communities at level 1 of OMACGraph, to which these users belong. The features and profiles of these users are compared with the corresponding ones of the Ego network nodes at the root level of OMACGraph to identify the matching nodes. For the GKS, GLPS, and BRWS methods, the human social network of the Facebook Social circles is used as a network of groups. For the RCF method [50], it builds a set of lattices (CLFs) from the ego networks and the human social network of the Facebook Social circles. Each ego network including the circle associated with each of the 4039 focal nodes is used by Neo4j [49] as properties. All the methods were run against the Facebook Social circles dataset. Finally, we compared the accuracy of the nine methods for detecting the Facebook Social Circles communities in terms of ARI (Table 8) and F1-score (Table 9).

6.6. Evaluating the accuracies of the methods for detecting the Friendster communities

In this evaluation, we used the Friendster dataset, which consists of real-world ground-truth networks compiled by the SNAP [56]. This dataset includes communities detected from an on-line gaming network and a social networking site. Users form groups and declare friendships. These user-defined declared groups are considered ground-truth communities. The networks consist of 65,608,366 nodes, 1,806,067,135 edges, and 957,154 communities. In the experiments, we represented each SAC by the set of prominent buzzwords extracted from the profiles of the users in the Friendster dataset who are associated with the SAC. To apply the ontology on the dataset, we imported the SIOC ontology to Protégé. Tables 10 and 11 show the accuracies of the methods for determining the Friendster communities in terms of ARI and F1-score respectively.

6.7. Evaluating the effectiveness of representing a SAC by its prominent buzzwords

In this test, we aim at evaluating our methodology for representing each SAC by the set of prominent buzzwords extracted from the messages associated with its members. As described in Section 5.1, the set of prominent buzzwords of a SAC is used to

Table 7
The prediction accuracies of the methods.

	GKS	GLPS	BRWS	Neo4j	RCF	DNMF	LOCD	CoRel	OMACexplorer
AvgPrecision@100(IA)	**0.0210**	0.0349	0.0355	0.0879	0.0662	0.0559	0.1208	**0.1419**	**0.1476**
AvgRecall@100(IA)	**0.3176**	0.6034	0.6050	**0.7257**	0.6761	0.6749	0.7105	0.7214	**0.8083**

In each row, the highest value is highlighted in blue color and each bolded value with gray background is either the highest, second highest, or lowest value.

Table 8
The accuracies of the nine methods for detecting the Facebook Social circles communities in terms of ARI.

	LOCD	DNMF	CoRel	RCF	GKS	GLPS	BRWS	Neo4j	OMDCexplorer
Minimum	0.59	0.47	**0.68**	0.54	**0.33**	0.42	0.36	0.61	**0.74**
Average	0.7	0.54	**0.80**	0.63	**0.38**	0.54	0.44	0.68	**0.94**
Maximum	0.81	0.61	**0.91**	0.78	**0.52**	0.64	0.56	0.84	**0.99**

In each row, the highest value is highlighted in blue color and each bolded value with gray background is either the highest, second highest, or lowest value.

Table 9
The accuracies of the nine methods for detecting the Facebook Social circles communities in terms of F1-score.

	LOCD	DNMF	CoRel	RCF	GKS	GLPS	BRWS	Neo4j	OMDCexplorer
Minimum	0.54	0.46	**0.59**	0.37	**0.23**	0.44	0.45	0.52	**0.66**
Average	0.64	0.54	**0.68**	0.48	**0.27**	0.49	0.52	0.61	**0.78**
Maximum	0.74	0.61	**0.76**	0.63	**0.41**	0.61	0.58	0.74	**0.87**

In each row, the highest value is highlighted in blue color and each bolded value with gray background is either the highest, second highest, or lowest value.

Table 10
The accuracies of the nine methods for detecting the Friendster communities in terms of ARI.

	LOCD	DNMF	CoRel	RCF	GKS	GLPS	BRWS	Neo4j	OMDCexplorer
Minimum	0.38	0.33	**0.45**	0.31	0.24	**0.19**	0.28	0.34	**0.43**
Average	0.47	0.4	**0.51**	0.38	**0.29**	0.29	0.33	0.39	**0.55**
Maximum	0.55	0.47	**0.56**	0.45	**0.32**	0.39	0.37	0.43	**0.66**

In each row, the highest value is highlighted in blue color and each bolded value with gray background is either the highest, second highest, or lowest value

Table 11
The accuracies of the nine methods for detecting the Friendster communities in terms of F1-score.

	LOCD	DNMF	CoRel	RCF	GKS	GLPS	BRWS	Neo4j	OMDCexplorer
Minimum	0.29	0.21	**0.34**	0.24	0.18	**0.16**	0.21	0.27	**0.34**
Average	0.33	0.27	**0.38**	0.31	0.23	**0.22**	0.27	0.33	**0.41**
Maximum	0.37	0.32	**0.42**	0.37	**0.27**	0.28	0.33	0.39	**0.47**

In each row, the highest value is highlighted in blue color and each bolded value with gray background is either the highest, second highest, or lowest value

represent it at level zero of the OMACGraph. Towards this, we compared an original version of our system OMACexplorer with a modified version of the system that represents a SAC by the general buzzwords extracted from its messages instead of the prominent buzzwords in the messages. We ran the original and modified versions of OMACexplorer against the Facebook Social circles dataset The original version was set to represent each SAC in the Facebook Social circles dataset by the *prominent buzzwords*, which the system extracted from the profiles in the dataset, which belong to the users of the SAC. The modified version was set to represent each SAC by only the *general buzzwords* extracted from the profiles of the SAC's users in the Facebook Social circles dataset. We compared the results of the two versions in terms of F1-score and ARI as shown in Table 12. It is evident from the results that the concept of buzzword dominancy played a considerable role in the quality of detected communities.

Table 12
The accuracies of the original and modified versions of OMACexplorer for detecting the Facebook Social circles communities. The original and modified versions represent a SAC by its prominent and general keywords respectively.

	Representing a SAC by the general buzzwords	Representing a SAC by the dominant buzzwords
Average F1-score	0.54	0.68
Average ARI	0.67	0.83

6.8. Statistical test of significance

In this section, we employ z-test [57] to assess whether the differences between the individual accuracy values of the results described in Sections 6.4–6.6 for each of the nine methods are large enough to be statistically significant. z-score is defined as the distance between a sample mean and the population mean in

Table 13
The population mean (pm) of the values of the accuracies reported in Sections 6.4–6.6 and the average standard deviation (avg) of the mean for the nine methods.

		LOCD		DNMF		CoRel		RCF		GKS		GLPS		BRWS		Neo4j		OMACexplorer	
		pm	avg	pm	avg	pm	avg	pm	avg	pm	avg	pm	avg	pm	avg	pm	avg	pm	avg
DBLP	ARI	0.54	0.057	0.42	0.091	0.62	0.058	0.39	0.064	0.20	0.103	0.34	0.095	0.36	0.098	0.52	0.067	0.61	0.059
	F1-score	0.21	0.041	0.12	0.038	0.23	0.034	0.10	0.043	0.05	0.074	0.14	0.052	0.09	0.056	0.20	0.047	0.29	0.037
Friendster	ARI	0.47	0.059	0.40	0.093	0.52	0.061	0.38	0.066	0.28	0.105	0.29	0.097	0.33	0.101	0.39	0.069	0.54	0.056
	F1-score	0.33	0.046	0.27	0.049	0.38	0.038	0.31	0.052	0.24	0.083	0.22	0.062	0.27	0.055	0.33	0.039	0.40	0.042
Social Circles	ARI	0.70	0.079	0.54	0.099	0.80	0.082	0.63	0.092	0.38	0.107	0.54	0.102	0.44	0.109	0.68	0.098	0.94	0.073
	F1-score	0.64	0.076	0.54	0.086	0.68	0.073	0.49	0.083	0.27	0.101	0.49	0.085	0.52	0.096	0.61	0.079	0.78	0.066

units of the standard error. It is computed as $Z = (ms - pm)/se$, where ms is the mean sample, pm is the population mean, $se = avg/sqrt(sz)$ is the standard error of the mean in which avg is the average standard deviation of the mean, and sz is the sample size. Table 13 shows the population mean (pm) of the values of the accuracies reported in Sections 6.4–6.6 and the average standard deviation (avg) of the mean for the nine methods. As the values of avg in Table 13 show, the individual accuracy values, and the prediction performance of all methods (except for GKS, BRWS, and GLPS) did not vary substantially.

7. Discussion of the results and potential application scenarios of OMACexplorer

7.1. Discussion of the results

7.1.1. Strengths of OMACexplorer

The results demonstrated that OMACexplorer inferred communities of nodes with multiple attributes with marked accuracy. These attributes are included in the profiles of the users in the Facebook Social circles dataset. We attribute this to the combination of OMACexplorer's: (1) graphical modeling of cross-profiles, (2) concept of SACs' interrelated MKCSSs, and (3) methodology of extracting the prominent buzzwords from the messages associated with communities. Based on our observation of the experimental results, OMACexplorer's detection accuracy increased as the number of attributes of an inferred community increased, while the other methods experienced no such increase. To confirm the above, we categorized the communities inferred by each method into sets according to their number of attributes. For each method, each set included communities with the same number of attributes. We then computed the overall average F1-score for each set, as shown in Fig. 7. As Fig. 7 shows, the detection accuracy of OMACexplorer increases consistently as the number community attributes increased. This is attributed to OMACexplorer's capability for detecting cross-profiles with various attributes using the concepts of OMACGraph, MKCSS Relationship Graph, and Interrelated Cross-MKCSSs. This leads to the detection of smaller cross-communities. As a community grows smaller, its interests become more granular and specific, which is evident in the Facebook Social circles dataset used in our experiments where interests are included in the user profiles in the dataset.

The experimental results confirmed that OMACexplorer's methodology for extracting the set of prominent buzzwords from the messages associated with a community had a significant impact on its performance over the other methods. Based on our analysis of the experimental results, we observed that the modified version of OMACexplorer, which did not employ the concept of prominent buzzwords (recall Table 12), gave many misleading similarity results. We observed that the modified

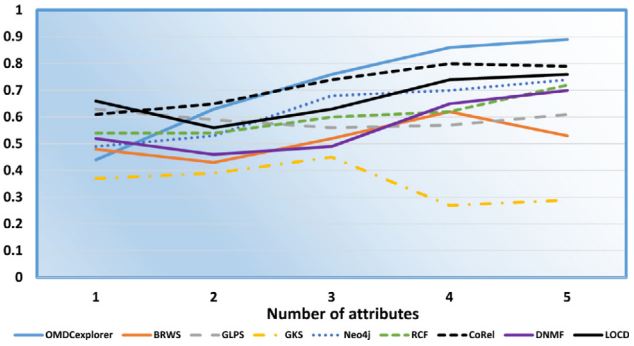

Fig. 7. The average F1-score of each set of inferred communities containing the same number of attributes.

version employed many uninformative terms (*e.g.*, conversational jargon terms, etc.), which led to, mistakenly, considering several different communities as similar. The co-occurrence of such uninformative terms in the messages associated with two communities may not be an accurate indicative of their relationship. Our concept of prominent keywords overcomes this by considering only prominent buzzwords that are reflective of a community's attributes. A buzzword is considered prominent only if it has a large number of occurrences in the majority of the messages associated with a community.

Finally, the constant enhancement of the training OMACGraph contributed also to the performance of OMACexplorer. This is because, every time OMACexplorer identified the multi-profiled cross-community, to which a user in the Facebook Social circles dataset belongs, it enhanced OMACGraph accordingly. This constant enhancement played a significant role in the prediction accuracy of OMACexplorer because it led to optimizing the number of buzzwords' occurrences (recall Fig. 3a) and pairwise and dominance scores (recall Fig. 3b) in the training profiles associated with each Facebook Social community.

7.1.2. Limitations of OMACexplorer

We observed from the experimental results that OMACexplorer achieved modest results when the percentage of incomplete users' profiles in a detected community was rather large. Specifically, OMACexplorer's performance tended to be relatively modest when the percentage of users' profiles that contain prominent buzzwords was small. We observed that OMACexplorer's performance tended to keep deteriorating as the percentage of users' profiles in a community containing prominent buzzwords decreases. To confirm this, we performed the follow-

Table 14
The average percentage of users' profiles containing all the prominent buzzwords of a detected community and the corresponding F1-score achieved by OMACexplorer for detecting the community.

Percentage of users' profiles containing all dominant buzzwords of a detected community	100%	75%	50%	25%
F1-score of the detected community	0.78	0.72	0.64	0.52

ing: (1) we computed the average percentage of users' profiles containing all prominent buzzwords of each community detected by OMACexplorer, and (2) we computed the corresponding F1-score achieved by OMACexplorer for detecting this community. Table 14 lists the results. As the table shows, the performance of OMACexplorer keeps deteriorating as the percentage of users' profiles in an inferred community, whose profiles contains all the prominent buzzwords of the community decreases. We will investigate approaches for overcoming this limitation in a future work.

7.1.3. Strengths and limitations of the methods proposed by Sharma et al. [51] (GKS, BRWS, & GLPS)

We observed from the experimental results that the GKS method detected with acceptable accuracy some of the communities that have high degree of internal homogeneity. However, its detection accuracy was poor for most of the communities that have low degree of cross-profiles heterogeneity. This manifested in the quality of the overlapping communities detected by the method. We attribute this shortcoming to GKS' Katz score, which is ineffective in determining which feature of a specific attribute is significant to be used for measuring the similarity between an ego and its circle friends. The score is incapable of identifying the features in the profiles of a circle that are of relevance to the focal node. As a result, the score may fail in identifying cross attribute similarities. We also observed that the Katz score is sensitive to small differences between the profiles of an ego and an 'alter'. A small deviation of the interests of an alter from the corresponding interests of an ego resulted in significant changes in the Katz score. This manifested in scenarios where two alters have minor changes in their profiles, yet they achieved significantly different Katz scores.

We observed that the BRWS method detected with acceptable accuracy most of the communities that have extensive *direct links* between external and internal actors. However, the method had an inconsistency in measuring the affinity between external actors and an internal target actor through *indirect links*. This is because the bi-random walk technique employed by BRWS suffers the problem of random fluctuations when measuring the affinities between different external actors and an internal actor through indirect links connecting them. Therefore, it may give misleading affinity scores. We also observed that the random walk procedure employed by BRWS may make short walks in the human social network of the Facebook Social circles stuck in the ego network. The GLPS method detected with good accuracy most of the communities, whose nodes have loose dependencies with each other. However, the hypergraph-based clustering technique employed by the method can cause dependencies among nodes. Many nodes can be independent. Moving these nodes to an overlapped partition may require other dependent nodes to be incorrectly moved with them. Also, the method produces partitions that do not stick to tighter balancing constraints.

7.1.4. Strengths and limitations of RCF

We observed from the experimental results that the RCF method successfully constructed global sequence of context sets and their corresponding sequence of lattice sets from the DBLP dataset. This is because, the Concept Analysis (FCA)-based technique employed by RCF could successfully convert the links between objects in the DBLP dataset into attributes and extract a set of lattices whose concepts are linked by relations. It could successfully do so in relations composed of a small to moderate number of instances of the relations. However, the method was not as successful in extracting sets of lattices, whose concepts are linked by relations composed of a relatively large number of instances of the relations. The major limitation of the method is that it considers a limited number of quantifiers. The method performed miserably in relations that required a combination of quantifiers that are not in the considered set.

7.1.5. Strengths and limitations of Neo4j

We observed from the results that the property graph mechanism adopted by Neo4j was very effective in scenarios where nodes and edges have various types of meta-information. We observed that the pattern matching of nodes and relationships performed by Neo4j when traversing a graph played a big role in the accuracy of communities detected by the method. This is attributed to the effective path-oriented queries adopted by Neo4j and its native declarative graph query language Cypher. Even in complex types of deep graphs that embody many levels, the results revealed that the method was effective in clustering the graphs into correct communities. This is because the ecosystem and its associated functionality on top of Neo4j permit the method to store infinite levels. Based on our observations, we outline the limitations of Neo4j as follows: (1) it allowed only one label on each edge and one value per attribute property while some of the datasets have multiple labels and values, (2) Cypher employed no-repeated-edge semantics, and (3) it had indexing limitations.

7.1.6. Strengths and limitations of DNMF

In general, the method achieved an acceptable accuracy results, which is attributed to employing the mutual guidance of two types of information, which are learnt in a unified manner: pseudo supervision module (which adopts unsupervised procedure for uncovering discriminative information) and community memberships. Based on our observation of the experimental results, the method achieved good results when the tradeoff parameter (α) was not assigned large values. Unfortunately, the method achieved poor results when: (1) it was assigned relatively large values, and (2) the number of overlapping nodes or number of overlapping memberships were large. It achieved good execution time, which is attributed to its algorithm, which decomposes the objective function to independent subproblems without the need for post-processing.

7.1.7. Strengths and limitations of LOCD

We observed from the experimental results that as the number of representative seed nodes selected by the method increased, the performance of the method improved. This is because, as the number of seed nodes increased, the number of other nodes bordered by these seed nodes and shared with them the same characteristics in detected communities increased. We also observed that as the fuzzy relation threshold increased, the number of correctly selected seed nodes increased. Specifically, we observed that the method tended to achieve good results by setting the fuzzy relation threshold to at least 0.87, which we used in the evaluations. The method outperformed most other

methods in detecting communities whose some of their nodes belonged to disconnected subnetworks. This was the case in many disconnected subnetworks of the Friendster and Facebook Social Circles datasets. This is attributed to the method's local community detection technique, which overcomes the problem of missing global information in disconnected subnetworks. The key limitation of the method is that its detection accuracy is greatly dependent on parameters. It could not correctly identify the community of many nodes as a result.

7.1.8. Strengths and limitations of CoRel

The method achieved outstanding results, which is attributed, largely, to its methodology of identifying related terms for each concept using constructed taxonomies. This methodology and our prominent keywords concept have similar motivations, which makes the CoRel method the closest to our method. We observed from the experimental results that the method was successful in identifying the related terms for many concepts associated with a community's profile/property terms. It could effectively extract distinctive terms for many network nodes. We also observed that the co-clustering technique employed by the method to disregard inconsistent subtopics played a significant role in its performance. The key limitation of the method lies in its enriching procedure, which could not enforce term distinctiveness in several networks.

7.2. Potential application scenarios of OMACexplorer

The OMACexplorer methodology can be used for solving various real world problem settings. We list below some potential application scenarios of the methodology:

- *Studying the dynamics of a social network:* Each social network has a certain degree of interaction between its members [58]. The degree of dynamicity and interaction between the members of a social network increase as the density of network's cross-communities increases. A social network with many dense cross-communities can imply a fast rate of information flow between the members of the network.
- *Analyzing the patterns of behavior that lead to the spread and outbreak of a disease:* Planning for epidemiological outbreaks and anticipating its trajectory and level of contagion require studying the inter-communities' patterns of behavior that can lead to the spread of diseases. The denser a quarantined infected cross-community is, the more effective the closure is in terms of tackling the spread of the disease. Health care researchers study densely infected cross-communities to: (a) understand the correlation between human mobility and the spread of diseases, and (b) trace diseases inter-communities' routes (within-community routes are easy to uncover unlike inter-community routes). This will help policy recommendation makers to impose tailored measures that amplify potential pandemics.
- *Identifying the influential individuals in a criminal organization:* In information forensic setting, densely connected cross-groups within a criminal network can be examined for identifying the influential commanders who issue instructions or serving as gatekeepers, who receive and distribute information and goods to other members of the network. Usually, the members of a cross-group are the key players, who oversee the groups. Their cross-group gathering aims at contemplating holonic plots. Removing them can disrupt the communication flow in the network [59] or other criminal acts such as spams [60].
- *Expanding business potentials:* In information systems setting, densely connected cross-communities can be studied for expanding businesses' potentials. Targeting the members of a cross-community can help companies in viral marketing for promoting their products, hoping that they can trigger a cascade of influence in their respective communities (*e.g.*, "word-of-mouth" marketing). It can also help in improving online services such as recommendations across communities (*i.e.*, cascading recommendations) [37].

8. Conclusion

We proposed a novel methodology named OMACexplorer for discovering granular multi-profiled cross-communities. To the best of our knowledge, this is the first work that: (1) analyzes hierarchically overlapped social profiles to detect the smallest and most granular multi-profiled cross-communities, and (2) infers the densest multi-profiled cross-community for an active user that matches his own social traits. We proposed novel graphical miniatures that consider all cross-profiles that come to existence from the interrelations between hierarchically overlapped social profiles. We proposed a methodology for extracting the set of prominent buzzwords from the messages associated with a social group. We also introduced novel representational graphs that depict the interrelationships between maximal union of k-cliques of users.

We evaluated OMACexplorer by comparing it with eight well-referenced methods. The results demonstrated that OMACexplorer outperformed the other methods. We attribute this to the combination of OMACexplorer's: (1) graphical modeling of cross-profiles, (2) concept of SACs' interrelated MKCSSs, and (3) methodology of extracting the prominent buzzwords from the messages associated with communities. We highlight and summarize below the findings of the experiments:

- The improvement of OMACexplorer over the other eight methods are 47% and 51% in terms of ARI and F1-score, respectively, which demonstrates the high efficiency of the proposed method.
- OMACexplorer was slightly outperformed by CoRel [33] in the minimum accuracies of DBLP-ARI (Table 4), Friendster-ARI (Table 10), and Friendster-F1-score (Table 11) due to its limitation of dealing with incomplete profiles.
- OMACexplorer's detection accuracy increased as the number of communities' attributes increased, as follows: 0.44, 0.63, 0.76, 0.86, and 0.89 for one, two, three, four, and five attributes respectively. This is advantageous to OMACexplorer, since in real-world the sizes of communities increase constantly as new members join in, which in turn results in increasing the number of attributes.
- The values of the OMACexplorer's parameters that achieved the best accuracy results were as follows: 0.44 for k, 0.28 for β (influence score threshold), and 0.37 for β (buzzword dominance score threshold).

OMACexplorer achieved modest results when the percentage of incomplete users' profiles in detected communities was rather large. We will investigate approaches for overcoming this limitation in a future work.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Creating an algorithm for building OMACGraph

The Appendix describe algorithm "Construct-Training-OMACGraph", which builds an OMACGraph. The input to the algorithm is a set of SACs. The output is the corresponding training OMACGraph. Line 4 invokes the subroutine presented in Appendix A.1, which constructs level zero of the graph using the techniques described in Section 4. Line 6 invokes the subroutine presented in Appendix A.2, which constructs level one of the graph based on messages' pairwise similarity (Section 4). Line 8 invokes the subroutine presented in Appendix A.3, which constructs level one of the graph based on interrelated MKCSSs (recall Section 4.1). Line 10 invokes the subroutine presented in Appendix A.5, which constructs the remaining levels of the graph (recall Section 4.2).

Algorithm 1: Construct-Training-OMACGraph

Input: $\{Я\}$ *//Set of SACs*
Output: $\mathbb{N}(\mathbb{V},\mathbb{E})$ *//Training OMACGraph*
1. $\{\mathbb{R}\} \leftarrow$ NIL *//Set of rootnodes*
2. $\{Ш_i\} \leftarrow$ NIL *//Set of over lapped nodes (i.e., ON) at level i*
3. $\{\mathbb{N}\} \leftarrow$ NIL
4.Create-Level-Zero $(\{Я\})$ *//Invoke subroutine Create − Level − Zero*
//Is level 1 creation based on messages' similarity or interrelated MKCSSs?
5. **if** (Create level 1 based on messages'similarity)
6. Create-Level-1-Using-Messages-Pairwise-Similarity $(\{Я\},\mathbb{N})$
7. **else**
8. Create-Level-1-Using-Messages-Pairwise-Similarity $(\{Я\},\mathbb{N})$
9. **end**
10. Create-Level-1-Using-Messages-Pairwise-Similarity $(\{Я\},\mathbb{N})$
11.**Return** $\mathbb{N}(\mathbb{V},\mathbb{E})$

A.1. Subroutine create-level-zero

For each SAC, (1) line 2 identifies the most influential MKCSS in the SAC (recall Section 3.1.2), (2) line 7 extracts the prominent keywords from the messages associated this MKCSS (recall Section 5.1), and (3) line 8 uses these keywords to represent the SAC in level zero of the graph.

Subroutine: Create-Level-Zero

Input: $\{Я\}$ *//Set of SACs*
Output: $\{\mathbb{N}\}$ *//A fragment of OMAC Graph containing level zero*
1. **for each** $(\mathbb{C}(x) \in \{Я\})$
2. $\mathbb{L} \leftarrow$ Identify−Influential−MKCSS $(\mathbb{C}(x) \in \{Я\})$
3. $\{\xi_{\mathbb{C}(\mathbb{L})}\} \leftarrow \{\not{м}_1, \not{м}_2, \ldots\}$ *//Set of messages associates with* $\mathbb{L}$
4. **for each** $\not{м} \in \{\xi_{\not{C}(\mathbb{L})}\}$
5. $\{K_{\mathbb{C}(\mathbb{L})}\} \leftarrow$ Extract − Keywords $(\not{м})$ *//Set of key words in* $\not{м}$
6. **end**
7. $\{đ_{\mathbb{C}(\mathbb{L})}\} \leftarrow$ Identify − Prominent − Keywords $(K_{\mathbb{C}(\mathbb{L})}, \xi_{\mathbb{C}(\mathbb{L})})$
8. $₦\{đ_{\mathbb{C}(x)}\} \leftarrow$ Create − Root − Node $(đ_{\mathbb{C}(x)})$
9. $\{\mathbb{R}\} \leftarrow \{\mathbb{R} \cup ₦\{đ_{\mathbb{C}(x)}\}\}$
10.**end**
11.**Return** $\{\mathbb{N}\} \leftarrow \{\mathbb{N} \cup \{\mathbb{R}\}\}$

A.2. Subroutine: Create-level-1-using-messages-pairwise-similarity

After invoking the subroutine in A.4, line 4 checks if the frequency of the common prominent keywords in SACs $₦\{đ_{\mathbb{C}(x)}\}$ and $₦\{đ_{\mathbb{C}(y)}\}$ is greater than a threshold. If so, the overlap between the two SACs is represented by a node at level 1 of OMACGraph, which is represented by the set of common prominent keywords in the two SACs (line 6).

Subroutine: Create-Level-1-Using-Messages-Pairwise-Similarity

Input: $\{Я\},\{\mathbb{N}\}$ *//*$\{\mathbb{N}\}$ *is a fragment of OMACGraph containing level 0*
Output: $\{\mathbb{N}\}$ *//*$\{\mathbb{N}\}$ *is a fragment of OMACGraph containing levels 0 and 1*
1. $i \leftarrow 1$
2. **for each** $(₦\{đ_{\mathbb{C}(x)}\}, ₦\{đ_{\mathbb{C}(y)}\}) \in \{\mathbb{R}\}$
3. Check−Pairwise−Similarity $(₦\{đ_{\mathbb{C}(x)}\}, ₦\{đ_{\mathbb{C}(y)}\})$ *//Recall A.4*
//Is the frequency of common prominent keywords greater than β
4. **if** $(F(₦\{đ_{\mathbb{C}(x)}\}, ₦\{đ_{\mathbb{C}(y)}\})) > \beta$
5. $И \leftarrow (₦\{đ_{\mathbb{C}(x)}\} \cap ₦\{đ_{\mathbb{C}(y)}\})$ *//Set of common prominent keywords*
6. $ON\{₦\{đ_{\mathbb{C}(x)}\} \cap ₦\{đ_{\mathbb{C}(y)}\}\} \leftarrow$ Create−Level 1−Node $(И)$
7. $\{Ш_i\} \leftarrow \{Ш_i \cup ON\{₦\{đ_{\mathbb{C}(x)}\} \cap ₦\{đ_{\mathbb{C}(y)}\}\}\}$
8 **end**
9.**end**
10. **Return** $\{\mathbb{N}\} \leftarrow \{\mathbb{N} \cup \{Ш_i\}\}$

A.3. Subroutine: Create-level-1-using-interrelated-MKCSS

Line 1 constructs the MKCSS Relationship Graph by invoking subroutine Construct-MRG in Appendix A.8. For each two interrelated MKCSS nodes $\ddot{u}$ and $\ddot{v}$, Lines 2–12 computes the influence score of the Association Edge $(\ddot{u}, \ddot{v})$ by invoking Eqs. (3) and (6) as described in Section 6.1. Line 14 creates a node in level one of OMACGraph.

Subroutine: Create-Level-1-Using-Interrelated-MKCSS

Input: $\{\text{Я}\}, \{\mathbb{N}\}$ // $\{\mathbb{N}\}$ *is a fragment of OMACGraph containing level 0*
Output: $\{\mathbb{N}\}$ // $\{\mathbb{N}\}$ *is a fragment of OMACGraph containing levels 0 and 1*
1..$\mathbb{G} \leftarrow \text{Construct}-\text{MRG}(M)$ //*ConstructMKCSSRelationshipGraph*
//*Nodes* $\ddot{u}$ *and* $\ddot{v}$ *belong to two SACs connected by association edge* $(\ddot{u}, \ddot{v})$
2. **for each** two interrelated nodes $\ddot{u}, \ddot{v} \in \mathbb{G}$
3. $\mathbb{W}(\ddot{u}) \leftarrow \text{Number}-\text{of}-\text{k}-\text{cliques}\ (\ddot{u})$
4. $\mathbb{W}(\ddot{v}) \leftarrow \text{Number}-\text{of}-\text{k}-\text{cliques}\ (\ddot{v})$
5. **for each** un-interrelated node $\eta \in \mathbb{G}$
6 $\text{m} \leftarrow$ Number of edges in the shortest path from η to $\ddot{u}$
7. $\sigma^{\text{m}}(\ddot{u}, \ddot{v}) \leftarrow$ Number of edges in the shortest path from η to $\ddot{u}$ that passes through the edge $(\ddot{u}, \ddot{v})$.
8. $\sigma^{\text{m}}(\text{ц}, \text{Ч}) \leftarrow$ # of paths with m edges from η to interrelated node $\text{ц} \neq \ddot{u}$ that passes through the edge $(\text{ц}, \text{Ч})$.
9. $\text{Ş}^{\eta}(\ddot{u}, \ddot{v}) = [(\sigma^{\text{m}}(\ddot{u}, \ddot{v})(\mathbb{W}(\ddot{u}) + \mathbb{W}(\ddot{v})))] / [\sigma^{\text{m}}(\ddot{u}, \text{Ч}) \sum (\mathbb{W}(\text{ц}) + \mathbb{W}(\text{Ч}))]^{\log_2 \text{m}}$
10. $\text{Ş}(\ddot{u}, \ddot{v}) \leftarrow \text{Ş}(\ddot{u}, \ddot{v}) + \text{Ş}^{\eta}(\ddot{u}, \ddot{v})$
11. **end**
12.**end**
13.$\ddot{\Upsilon} \leftarrow$ Association [Edge with highest influence score $(\text{Ş}(\ddot{u}, \ddot{v}))$
14.$\text{ON}\left\{\mathbb{N}\left\{\underline{\mathfrak{d}}_{\mathbb{C}}\left(\ddot{\Upsilon}\right)\right\}\right\} \leftarrow \text{Create}-\text{Level1}-\text{Node}\left(\ddot{\Upsilon}\right)$ //ON : *overlapped node*
15.$\{\text{Ш}_{\text{i}}\} \leftarrow \left\{\text{Ш}_{\text{i}} \cup ON\left\{\mathbb{N}\left\{\underline{\mathfrak{d}}_{\mathbb{C}}\left(\ddot{\Upsilon}\right)\right\}\right\}\right\}$ //*Set of overlapped nodes at level* 1
16. **Return** $\{\mathbb{N}\} \leftarrow \{\mathbb{N} \cup \{\text{Ш}_{\text{i}}\}\}$

A.4. Subroutine: Check-pairwise-similarity

Component Pairwise Similarity Identifier (recall Fig. 1) uses this subroutine to compute the frequency of common prominent keywords in the messages associated with two input SACs. Line 4 determines the set of prominent keywords that is common the messages associated with the two SACs. Based on application-specific requirements, the impact of rare events may be needed to be diminished. If so, line 8 will compute the formula in the line. Otherwise, line 13 will compute the formula in the line.

Subroutine: Check-Pairwise-Similarity

Input: $\{\mathbb{N}\{\underline{\mathfrak{d}}_{\mathbb{C}(x)}\}, \mathbb{N}\{\underline{\mathfrak{d}}_{\mathbb{C}(y)}\}\}$ //*SACs*$\mathbb{N}\{\underline{\mathfrak{d}}_{\mathbb{C}(x)}\}$ *and* $\mathbb{N}\{\underline{\mathfrak{d}}_{\mathbb{C}(y)}\}$
Output: $\text{F}\left(\mathbb{N}\{\underline{\mathfrak{d}}_{\mathbb{C}(\text{x})}\}, \mathbb{N}\{\underline{\mathfrak{d}}_{\mathbb{C}(\text{y})}\}\right)$ //*Common prominent keywords' frequency in the SACs*
1.$\left\{\acute{M}\right\} \leftarrow$ NIL //*Set of messages containing common prominent keywords*
2.**for each** $\text{m} \in \mathbb{N}\{\underline{\mathfrak{d}}_{\mathbb{C}(\text{y})}\}$ //$\not{m}$ *is a message associated with* $\mathbb{N}\{\underline{\mathfrak{d}}_{\mathbb{C}(y)}\}$
3. **if** $\left(\text{m}\{\mathbb{N}\{\underline{\mathfrak{d}}_{\mathbb{C}(\text{x})}\} \cap \mathbb{N}\{\underline{\mathfrak{d}}_{\mathbb{C}(\text{y})}\}\} \neq 0\right)$
4. $\left\{\acute{\text{M}}\right\} \leftarrow \left\{\acute{\text{M}}\right\} \cup \not{m}$ //*Set of common prominent keywords*
5. **end**
6.**end**
7.**if** (Impact of rare events needs to be diminish)
8. **Return** $\text{F}\left(\mathbb{N}\{\underline{\mathfrak{d}}_{\mathbb{C}(\text{x})}\}, \mathbb{N}\{\underline{\mathfrak{d}}_{\mathbb{C}(\text{y})}\}\right) \leftarrow \log\left(1 + |\mathbb{N}\{\underline{\mathfrak{d}}_{\mathbb{C}(\text{x})}\}| / \left|\left\{\acute{\text{M}}\right\}\right|\right)$
9.**else**
10. $\check{N} \leftarrow$ NIL //*Maximum number of prominent keywords' occurrences*
11. **for each** $\text{K} \in \{\mathbb{N}\{\underline{\mathfrak{d}}_{\mathbb{C}(\text{x})}\} \cap \mathbb{N}\{\underline{\mathfrak{d}}_{\mathbb{C}(\text{y})}\}\}$
12. **for each** $\not{m} \in \mathbb{N}\{\underline{\mathfrak{d}}_{\mathbb{C}(\text{x})}\}$
13. $\overline{\text{O}} \leftarrow \text{Number}-\text{of}-\text{Occurrences}(\text{K}, \not{m})$
14. **if** $\left(\overline{\text{O}} > \check{\text{N}}\right)$
15. $\check{N} \leftarrow \text{O}$
16. **end**
17. **end**
18. **end**
19. **Return** $\text{F}\left(\mathbb{N}\{\underline{\mathfrak{d}}_{\mathbb{C}(\text{x})}\}, \mathbb{N}\{\underline{\mathfrak{d}}_{\mathbb{C}(\text{y})}\}\right) \leftarrow \log_{\mathbf{2}}\left(1 + \left(\check{\text{N}}/|\acute{\text{M}}|\right)\right)$
20.**end**
21.**Return F** $\left(\mathbb{N}\{\underline{\mathfrak{d}}_{\mathbb{C}(\text{x})}\}, \mathbb{N}\{\underline{\mathfrak{d}}_{\mathbb{C}(\text{y})}\}\right)$

A.5. Subroutine: Create-remaining-levels

This subroutine constructs the remaining levels of OMACGraph as described in Section 4.2. If there is at least one common SAC among a combination of overlapped nodes at level i of the OMACGraph, line 5 creates a new overlapped node at level i + 1 of the graph to represent the overlap of the two nodes. This process continues until no new overlapped node at a different can be created (line 11).

Subroutine: Create-Remaining-Levels

Input:$\{Я\}, \{\mathbb{N}\}$ *//A fragment of OMACGraph containing levels 0 and 1*
Output:$\{\mathbb{N}\}$ *//* $\{\mathbb{N}\}$ *: The complete OMACGraph*
1.$\mathbf{i} \leftarrow 1$
2.**while** (a new overlapped node can be created at level $\mathbf{i+1}$)
3 **for each** $\left(\text{combination of overlapped nodes } \mathring{\mathrm{U}} \subset \{\mathrm{m_i}\}\right)$
4. if $\left(\text{there is at least one common SAC} \in \mathring{\mathrm{U}}\right)$
5. $\mathrm{ON}\left\{\mathring{\mathbf{U}}\right\} \leftarrow \mathrm{Create-Level i+1-Node}\left(\mathring{\mathrm{U}}\right)$ *//ON : overlapped node*
6 $\{\mathrm{III_{i+1}}\} \leftarrow \left\{\{\mathrm{III_{i+1}}\} \cup \mathbf{ON}\left\{\mathring{\mathrm{U}}\right\}\right\}$
7. **end**
8. **end**
9. $\{\mathbb{N}\} \leftarrow \left\{\mathbb{N} \cup \{\mathrm{III_{i+1}}\}\right\}$
10. $\mathrm{i} \leftarrow \mathrm{i}+1$
11.**end**
12.**Return**$\{\mathbb{N}\}$

A.6. Subroutine: Identify-influential-MKCSS

This subroutine identifies the most influential MKCSS in a SAC as described in Section 3.1.2. Lines 3–24 computes the influence score of each MKCSS v as stated in Eq. (1), as follows. Lines 9–14 compute the number of paths between each two MKCSS nodes μ_i and μ_j and also the fraction of these paths that pass-through v. Line 15 computes the weights of nodes μ_i and μ_j. Lines 17 and 19 computes the weights of μ_i and μ_j if they pass through v. Lines 21 and 23 compute the influence score of v as stated in Eq. (1). Line 25 ranks the MKCSSs based on their influence scores. Line 26 returns the MKCSS with the highest influence score as the most influential one.

Subroutine: Identify-Influential-MKCSS

Input: $\{Я\}$
Output: $\mathcal{S}(v)$ *// The most influential MKCSS node*
1.$\mathcal{S}(v) \leftarrow$ NIL *//Call subroutine Detect−MKCSS in A.8*
2.$M \leftarrow Detect-MKSSs(\{Я\})$ *//*M : *ThesetofMKCSSs*
3.**for each** $v \in \mathrm{M}$ *//Check each pair* μ_i *and* μ_j *of MKCCs in the set* M
4. **for each** $\mu_\mathrm{i}, \mu_\mathrm{j} \in \mathrm{M}$
5. $\sigma(\mu_\mathrm{i}, \mu_\mathrm{j}) \leftarrow$ NIL
6. $\sigma_v(\mu_\mathrm{i}, \mu_\mathrm{j}) \leftarrow$ NIL
7. $\mathbb{W}(\mu_\mathrm{i}, \mu_\mathrm{j}) \leftarrow$ NIL
8. $\mathbb{W}_v(\mu_\mathrm{i}, \mu_\mathrm{j}) \leftarrow$ NIL
9. **for each** path from μ_i to μ_j *//Number of shortest paths from* μ_i *to* μ_j
10. $\sigma(\mu_\mathrm{i}, \mu_\mathrm{j}) \leftarrow \sigma(\mu_\mathrm{i}, \mu_\mathrm{j}) + 1$
11. **if** the path passes through v *//Number of shortest paths from* μ_i *to* μ_j *that pass−through v*
12. $\sigma_v(\mu_\mathrm{i}, \mu_\mathrm{j}) \leftarrow \sigma_v(\mu_\mathrm{i}, \mu_\mathrm{j}) + 1$
13. **end**
14. **end**
15. $\mathbb{W}(\mu_\mathrm{i}, \mu_\mathrm{j}) \leftarrow$ Weight of MKCSSs μ_i and μ_j
16. **if** the path passes through v *//Weight of* μ_i *and* μ_j *if the path between them passes through v*
17. $\mathbb{W}_v(\mu_\mathrm{i}, \mu_\mathrm{j}) \leftarrow \mathbb{W}(\mu_\mathrm{i} + \mu_\mathrm{j})$
18 **else**
19. $\mathbb{W}_v(\mu_\mathrm{i}, \mu_\mathrm{j}) \leftarrow \mathbf{0}$
20.. **end** *//Recall Equation 1*
21. $\mathcal{S}(v) \leftarrow \mathcal{S}(v) + \log_2\left[\left(\sigma_v(\mu_\mathrm{i}, \mu_\mathrm{j}) / \sigma(\mu_\mathrm{i}, \mu_\mathrm{j})\right) \times \left(\mathbb{W}_v(\mu_\mathrm{i}, \mu_\mathrm{j}) / \mathbb{W}(\mu_\mathrm{i} + \mu_\mathrm{j})\right)\right]$
22. **end** *//Compute the influence score of v as stated in Equation1*
23. $\ddot{I}(v) \leftarrow \log_2 \mathcal{S}(v)$
24.**end**
25.Rank MKCSSs based on their influence scores $\ddot{I}(v)$
26.**Return** v with the highest score $\ddot{I}(v)$

A.7. Subroutine: Detect-MKCSSs

This subroutine constructs MKCSSs as described in Section 3.1.2 Line 5 detects k cliques of users within SAC Ş. Line 6 augments related *k*-cliques in a cross-community $_M$. Line 10 further augments $_M$ by including each *k*-clique of users connected to a user already in $_M$. After no more *k*-clique can be added, line 13 constructs MKCSS *M*, which is a maximal union of *k*-clique of users. Eventually, line 19 returns all MKCSSs within SAC Ş.

Subroutine: Detect-MKCSSs

Input: {Я}
Output: $\{M\}$ *//Set of MKCSSs*
1. **for each** SAC $Ş \in \{Я\}$
2. $\{M\} \leftarrow$ NIL *//For each user ù who belongs to SAC Ş*
3. **for each** node ий $\in$ Ş
4. **while** {{M} is not a maximal union of k−cliques} *//Find $k-$clique of users containing user* ий
5. $\{A\} \leftarrow \text{Find}-\text{k}-\text{adjacent}-\text{user}-\text{nodes}(\text{ий}, Ş)$ *//M is a cross $-$ community of $k-$ cliques*
6. $\{_M\} \leftarrow \{_M\} \cup \{A\}$ *//For each user node § that belongs to $k-$ clique A*
7. **for each** node § $\in \{A\}$ *//Find $k-$ clique of users containing user §*
8. $\left\{\text{Å}\right\} \leftarrow Find-k-adjacent-user-nodes(§, Ş)$
9. **if**$\left(\left\{\text{Å}\right\} \neq \{A\}\right)$ *//Augment cross $-$ community $_M$ by adding the $k-$ clique of Å*
10. $\{_M\} \leftarrow \{_M\} \cup \left\{\text{Å}\right\}$
11. **end**
12. **end** *//Add $_M$ to MKCSS* M
13 $\{M\} \leftarrow \{M\} \cup \{_M\}$
14. ий $\leftarrow \left(\text{ü} \in \left\{\text{Å}\right\}\right)$
15. **end** *//Exclude the set of discovered MKCSSs*
16. $Ş \leftarrow Ş - \{M\}$
17. **end**
18.**end** *//Returns all MKCSSs within SAC Ş*
19.**Return**$\{M\}$

A.8. Subroutine: Construct-MRG

This subroutine constructs MKCSS Relationship Graph as described in Section 3.1.3 Line 7 creates a node for each MKCSS in the MKCSS Relationship Graph and line 11 compute the weight of the node. If two MKCSSs share at least one node (line 12), an edge linking the two nodes is created in the MKCSS Relationship Graph (line 14).

Subroutine: Construct-MRG

Input: {Я}
Output: MRG *//MKCSS Relationship Graph*
1. $\{M\} \leftarrow Detect-MKCSSs(\{Я\})$ *//Call the subroutine in Section A.7*
2. $\{\xi\} \leftarrow$ NIL *//Set of edges in the MKCSS Relationship Graph*
3. $\left\{\tilde{N}\right\} \leftarrow$ NIL *//Set of nodes in the MKCSS Relationship Graph*
4. $\{\omega\} \leftarrow$ NIL *//Weights of the nodes in the MKCSS Relationship Graph*
5. **for each** $§_i \in \{M\}$ *//For each MKCSS $§_i$ in the set of MKCSSs $\{M\}$*
6. $\aleph(§_i) \leftarrow \text{Create}-\text{MRG}-\text{Node}(§_i)$ *//Create a node $\aleph(§_i)$ for $§_i$ in MRG*
7. $\left\{\tilde{N}\right\} \leftarrow \left\{\tilde{N}\right\} \cup \aleph(§_i)$
8. $\text{w}(§_i) \leftarrow \text{Number of k}-\text{cliques in } §_i$ *//Weight of $§_I-$ Theorem 1*
9. $\Psi \leftarrow \Psi \cup \text{w}(§_i)$
10. **end**
11.MRG $\leftarrow$ MRG $\cup \left\{\tilde{N}\right\} \cup \Psi$
12. **for each** $§_i, §_j \in \{M\}$ *//For each two MKCSSs $§_i$ and $§_j$ in $\{M\}$*
13. **if** $((§_i \cap §_j) \neq 0)$ *//If MKCSSs $§_i$ and $§_j$ share a common node(s)*
14. $\{E\} \leftarrow \{E\} \cup edge(\aleph(§_i), \aleph(§_j))$ *//Create edge $(\aleph(§_i), \aleph(§_j))$*
15. MRG $\leftarrow$ MRG $\cup \{E\}$
16. **end**
17.**end**
18.**Return** MRG *//Return the MKCSS Relationship Graph*

Appendix B

We compare in Table 15 the methods described in Section 2 (Related Work) in terms of five important criteria. Refer to Section 2 for detailed descriptions of the methods.

Table 15
Comparing the methods reported in Section 2 in terms of five criteria.

	Consider network content	Consider incomplete attributes	Consider network structure	Remove uninformative terms/outliers	Consider missing links
[19]	×	×		×	
[21]	×	×	×		
[22]	×		×		×
[23]	×	×	×		×
[24]	×	×		×	×
[26]	×		×	×	
[27]	×		×		×
[28]	×	×	×		
[30]	×	×	×		
[31]		×	×		×
[32]			×		×
[33]	×	×	×		×
[34]			×		×
[40]		×	×	×	×
[41]	×	×	×		×
[42]	×		×	×	
[43]		×	×		
This paper	×	×	×	×	

References

[1] J. Camacho, R. Guimerà, L. Amaral, Robust patterns in food web structure, Phys. Rev. Lett. 88 (2002) 228102.
[2] G. Palla, I. Derenyi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, Nature 435 (2005) 814–818.
[3] M.E.J. Newman, Scientific collaboration networks: II. Shortest paths, weighted networks, and centrality, Phys. Rev. E 64 (2001) 016132.
[4] G. Flake, S. Lawrence, C. Giles, Efficient identification of web communities, in: 6th ACM SIGKDD, NY, USA, 2000.
[5] Y. Zhou, H. Cheng, X. Yu, Graph clustering based on structural/attribute similarities, in: VLDB End., 2009, France, 2009.
[6] J. Yang, J. McAuley, J. Leskovec, Community detection in networks with node attributes, in: Proceedings of the IEEE International Conference on Data Mining, 2013, USA, 2013, pp. 1151–1156.
[7] L. Akoglu, H. Tong, B. Meeder, C. Faloutsos, PICS: parameter-free identification of cohesive subgoups in large attributed graphs, in: Proceedings of the SIAM International Conference on Data Mining, 2012, USA, 2012, pp. 439–450.
[8] E.J. Newman, A. Clauset, Structure and inference in annotated networks, Nature Commun. 7 (2015) 11863.
[9] Z. Xu, Y. Ke, Y. Wang, H. Cheng, J. Cheng, GBAGC: a general Bayesian framework for attributed graph clustering, ACM Trans. Knowl. Discov. Data 9 (2014) 5:1–5:43.
[10] M. Berlingerio, F. Pinelli, F. Calabrese, Abacus: Frequent pattern mining-based community discovery in multidimensional networks, Data Min. Knowl. Discov. 27 (3) (2013) 294–320.
[11] M. Al Zaabi, K. Taha, T. Martin, CISRI: A crime investigation system using the relative importance of information spreaders in networks depicting criminals communications, IEEE Trans. Inf. Forensics Secur. 10 (10) (2015) 2196–2211.
[12] K. Taha, P. Yoo, Detecting overlapping communities of nodes with multiple attributes from heterogeneous networks, in: 15th EAI International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom). London, Great Britain, 2019.
[13] K. Taha, Detecting disjoint communities in a social network based on the degrees of association between edges and influential nodes, IEEE Trans. Knowl. Data Eng. (TKDE) 33 (3) (2021) 935–950.
[14] V. Belák, S. Lam, C. Hayes, Towards maximising cross-community information diffusion, in: IEEE Advances in Social Networks Analysis and Mining, 2012, 2012, pp. 171–178.
[15] B. Guo, Z. Yu, D. Zhang, X. Zhou, Cross-community sensing and mining, IEEE Commun. Mag. 52 (8) (2014) 144–152.
[16] A. Park, R. rank, A. Mikhaylov, M. Thomson, Hackers hedging bets: a cross-community analysis of three online hacking forums, in: IEEE/ACM International Conference on Advances in Social Networks, August, 2018, Barcelona, Spain.
[17] R. Frank, M. Thomson, A. Mikhaylov, A. Park, Putting all eggs in a single basket: A cross-community analysis of 12 hacking forums, in: IEEE International Conference on Intelligence and Security Informatics, 2018, pp. 136–141.
[18] A.J. Park, R. Frank, A. Mikhaylov, M. Thomson, Hackers hedging bets: A cross-community analysis of three online hacking forums, in: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2018, pp. 798–805.
[19] H. Chang, Z. Wang, L. Vilnis, A. McCallum, Distributional inclusion vector embedding for unsupervised hypernymy detection, in: NAACL-HLT, 2017.
[20] M. Zhitomirsky-Geffet, I. Dagan, The distributional inclusion hypotheses and lexical entailment, in: ACL, 2005.
[21] C. Zhang, F. Tao, X. Chen, J. Shen, M. Jiang, B. Sadler, M. Vanni, J. Han, TaxoGen: Unsupervised topic taxonomy construction by adaptive term embedding and clustering, in: KDD '18, 2018.
[22] M. Atzori, S. Balloccu, Fully-unsupervised embeddings-based hypernym discovery, Information 11 (5) (2020) 268.
[23] T. Gao, X. Han, Z. Liu, M. Sun, Hybrid attention-based prototypical networks for noisy few-shot relation classification, in: AAAI, 2019.
[24] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: NAACL-HLT, 2019.
[25] L. Soares, N. FitzGerald, J. Ling, T. Kwiatkowski, Matching the blanks: Distributional similarity for relation learning, in: ACL, 2019.
[26] I. Renau, R. Marín, G. Ferraro, A. Balvet, R. Nazar, Pruning and repopulating a lexical taxonomy: experiments in Spanish, English and French, J. Intell. Syst. 30 (1) (2020) 376–394.
[27] S. Singh, A. Hug, A. Dieuleveut, M. Jaggi, Context mover's Distance & barycenters: Optimal transport of contexts for building representations, in: International Conference on Artificial Intelligence and Statistics (AISTATS), 2020.
[28] J. Shen, Z. Wu, D. Lei, C. Zhang, X. Ren, M. Vanni, B. Sadler, J. Han, HiExpan: Task-guided taxonomy construction by hierarchical tree expansion, 2018, ArXiv abs/1910.08194.
[29] J. Shen, Z. Wu, D. Lei, J. Shang, X. Ren, J. Han, SetExpan: Corpus-based set expansion via context feature selection and rank ensemble, in: ECML/PKDD, 2017.
[30] S. Dash, M. Chowdhury, A. Gliozzo, N. Mihindukulasooriya, N. Fauceglia, Hypernym detection using strict partial order networks, 2019.
[31] X. Shi, H. Lu, G. Jia, Adaptive overlapping community detection with bayesian nonnegative matrix factorization, in: DASFAA, Springer, 2017, pp. 339–353.
[32] T. He, L. Hu, K. Chan, P. Hu, Learning latent factors for community identification and summarization, IEEE Access 6 (2018) 30137–30148.
[33] J. Huang, Y. Xie, Y. Meng, Y. Zhang, J. Han, CoRel: Seed-guided topical taxonomy construction by concept learning and relation transferring, in: 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), 2020, pp. 1928–1936.
[34] F. Ye, C. Chen, Z. Zheng, R. Li, J. Yu, Discrete overlapping community detection with pseudo supervision, in: The 19th IEEE International Conference on Data Mining (ICDM), Beijing, China, 2019.
[35] H. Lim, Kim. H., Item recommendation using tag emotion in social cataloging services, Expert Syst. Appl. 89 (2017) 179–187.
[36] E. Cambria, R. Speer, C. Havasi, A. Hussain, Senticnet: A publicly available semantic resource for opinion mining, in: AAAI Fall Symposium: Commonsense Knowledge, 2010.
[37] Reza Zafarani, i Huan Liu, Users joining multiple sites: Friendship and popularity variations across sites, Inf. Fusion 28 (2016) 83–89.

[38] P. De Meo, E. Ferrara, F. Abel, L. Aroyo, G.J. Houben, Analyzing user behavior across social sharing environments, ACM Trans. Intell. Syst. Technol. (TIST) 5 (1) (2013) 14.
[39] R. Al-Dalky, K. Taha, D. Al Homouz, M. Qasaimeh, Applying Monte Carlo simulation to biomedical literature to approximate genetic network, IEEE/ACM Trans. Comput. Biol. Bioinform. 13 (3) (2016) 494–504.
[40] C. Aggarwal, Y. Xie, P. Yu, Towards community detection in locally heterogeneous networks, in: SDM, 2011, pp. 391–402.
[41] Y. Sun, C. Aggarwal, J. Han, Relation strength-aware clustering of heterogeneous information networks with incomplete attributes, in: VLDB, 2012.
[42] Qi, C. Aggarwal, T. Huang, On clustering heterogeneous social media objects with outlier links, in: WSDM, 2012, pp. 553–562.
[43] J. Cruz, C. Bothorel, F. Poulet, Integrating heterogeneous information within a social network for detecting communities, in: ASONAM, 2013.
[44] K. Taha, K. Salah, P. Yoo, Clustering the dominant defective patterns in semiconductor wafer maps, IEEE Transactions on Semiconductor Manufacturing 31 (1) (2018) 156–165.
[45] R. Bhopal, Glossary of terms relating to ethnicity and race for reflection and debate, J. Epidemiol. Community Health 58 (2004) 441–445.
[46] CoreNLP (accessed April 2019), Stanford University Available at: https://stanfordnlp.github.io/CoreNLP/.
[47] Protégé. Stanford Center for Biomedical Informatics Research, Stanford University: https://protege.stanford.edu/.
[48] L. Ni, W. Luo, W. Zhu, B. Hua, Local overlapping community detection, ACM Trans. Knowl. Discov. Data (TKDD) 14 (1) (2019).
[49] The Neo4j Database, The Neo4j manual v3.0, 2016, http://neo4j.com/docs/stable/. (2016).
[50] S. Guesmi, C. Trabelsi, C. Latiri, Community detection in multi-relational bibliographic networks, in: Database and Expert Systems Applications, in: Lecture Notes in Computer Science, vol. 9828, Springer, 2016, pp. 11–18.
[51] A. Sharma, R. Kuang, J. Srivastava, X. Feng, K. Singhal, Predicting small group accretion in social networks: A topology based incremental approach, in: IEEE/ACM International Conference on Advance in Social Networks Analysis and Mining (ASONAM), 2015, pp. 408–415.
[52] L. Katz, A new status index derived from sociometric analysis, Psychometrika 18 (1) (1953) 39–43.
[53] J. Tang, j. Zhang, L. Yao, J. Li, L. Zhang, X. Su, ArnetMiner: Extraction and mining of academic social networks, in: Proceedings of the Fourteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'2008), pp. 990–998.
[54] M. Zolkepli, F. Dong, K. Hirota, Visualizing fuzzy relationship in bibliographic big data using hybrid approach combining fuzzy c-means and Newman-Girvan algorithm, JACIII 18 (6) (2014) 896–907.
[55] S. Ganguly, V. Pudi, Paper2vec: Combining graph and text information for scientific paper representation, in: J. Jose, et al. (Eds.), Advances in Information Retrieval. ECIR 2017, in: Lecture Notes in Computer Science, vol. 10193, Springer, Cham, 2017.
[56] SNAP (accessed December 2020), Stanford University. Available at: http://snap.stanford.edu/data/.
[57] R. Warner, Applied Statistics: From Bivariate Through Multivariate Techniques, Sage Publications, 2007.
[58] S. Abkenar, M. Kashani, E. Mahdipour, S. Jameii, Big data analytics meets social media: A systematic review of techniques, open issues, and future directions, Telemat. Inform. (2020) 101517.
[59] K. Taha, P. Yoo, Using the spanning tree of a criminal network for identifying its leaders, IEEE Trans. Inf. Forensics Secur. 12 (2) (2016) 445–453.
[60] S. Abkenar, E. Mahdipour, S. Jameii, M. Kashani, A hybrid classification method for Twitter spam detection based on differential evolution and random forest, Concurr. Comput.: Pract. Exper. 2021 (2020).