Information Sciences 510 (2020) 70–88

Contents lists available at ScienceDirect

Information Sciences

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

User interaction-oriented community detection based on cascading analysis

Linbo Luo[a,*], Kexin Liu[a], Bin Guo[b], Jianfeng Ma[a]

[a] School of Cyber Engineering, Xidian University, Xi'an, China
[b] School of Computer Science, Northwestern Polytechnical University, Xi'an, China

ARTICLE INFO

Article history:
Received 2 March 2019
Revised 15 July 2019
Accepted 13 September 2019
Available online 14 September 2019

Keywords:
Community detection
Online social networks
Overlapping communities
Cascading analysis

ABSTRACT

Detecting user communities in online social networks (OSNs) is of great importance for understanding social dynamics. Compared to user connections, user interactions have been shown to be more meaningful for reflecting peer relationship in OSNs. To this end, we propose a user interaction-oriented community detection method based on cascading analysis. Specifically, user interactions are analyzed from a large collection of social object sharings (e.g., blog posts, photo shares). Both direct and indirect user interactions associated with each social object sharing are then extracted and the cascading relations among these interactions are captured using a graph representation. The proposed method makes use of such cascading relations to extract groups of actively interacting users and adopts a super graph approach to cluster these user groups for detecting communities. An extensive evaluation of our method was performed using three real OSN datasets and compared with three state-of-the-art overlapping community detection methods, namely two general methods applied to the interaction graph and an interaction-based method. Our method outperformed the compared methods, as demonstrated by several evaluation metrics, and produced more robust and stable detection results across different datasets.

1. Introduction

Online social networks (OSNs) are common venues in cyberspace where people make friends, share their thoughts and exchange ideas about daily life. OSNs have therefore become a valuable data source for researchers to understand social relationships and human behaviors. Among the various studies of OSNs, community detection is an important research area that aims to reveal the structure of user groups (i.e., communities) within a given OSN. The detection of such user communities is useful for many applications, such as targeted advertising, collaborative recommendations, public sentiment monitoring, influence analysis and network security.

Traditional community detection for complex networks relies mostly on linkage analysis of the network. In the context of online social networks, linkage analysis usually considers the connections of users (e.g., being friends or followers) as *social links* for quantifying peer relationships. However, many studies [15,22,26,40] have suggested that users' connections cannot fully reflect the network dynamics in OSNs. In fact, we often observe Milgram's "Familiar Stranger" phenomenon [28] in OSNs. That is, even though two users add each other as friends, they seldom interact. As pointed out in [40], users tend

[*] Corresponding author.
E-mail address: lbluo@xidian.edu.cn (L. Luo).

https://doi.org/10.1016/j.ins.2019.09.022
0020-0255/© 2019 Elsevier Inc. All rights reserved.

to interact mostly with a small subset of friends, and so user interactions (e.g., comments, likes and retweets) are more meaningful for reflecting peer relationships in OSNs. This work therefore focuses on the design of a user interaction-oriented approach to community detection for OSNs.

To effectively exploit information of user interactions for community detection, two observations regarding the characteristics of user interaction behaviors in OSNs are considered. Firstly, user interactions in OSNs are closely related to the sharing activities of users. In this work, we refer to activities including blog posts, photo shares, video shares, etc. as *social object sharings*. Users make interactions on top of these social object sharings and tend to generate more interactions if the topic of a given social object sharing is of interest to users. Therefore, we believe that analysis of the interactions associated with social object sharings can help to elicit the user community with the common interest. Secondly, user interactions in OSNs include direct interactions as well as indirect ones such as “comments on comments”, “comments on retweets” and “retweets of retweets”. Consequently, these interactions yield a cascading effect which may cause more users within the same community to interact with each other.

Given the aforementioned observations, we propose a user interaction-oriented community detection method based on cascading analysis. Specifically, a collection of social object sharings from a given OSN is first obtained and both the direct and indirect user interactions associated with each social object sharing are then analyzed. A graph representation, termed *event graph*, is introduced to capture cascading relations among all direct and indirect interactions. Within each event graph, clusters of small user groups, termed *sub-events*, are extracted with the aim of identifying the users from the same community who have actively interacted with each other in a given social object sharing. A super graph is then constructed with each node representing an extracted sub-event. Community detection is finally performed on this super graph with each detected community representing a group of users who have actively interacted with each other over multiple social object sharings. Overall, the proposed method aims to detect active communities in OSNs based on user interaction behaviors rather than one-time friendship establishments.

The main characteristics and contributions of this study are summarized as follows:

- To leverage user interaction data to analyze the community structure in OSNs, an event graph representation is presented and constructed to organize user interactions and associated cascading relations among these interactions.
- Densely-connected user groups termed sub-events are extracted within each constructed event graph to elicit users that potentially belong to the same community.
- A super graph approach is proposed, which treats sub-events from multiple event graphs as super nodes and clusters these nodes in a super graph to detect communities.
- Extensive evaluation using real world OSN datasets is conducted and the results show that our proposed method can outperform state-of-the-art overlapping community detection methods and produce more robust and stable detection results across different datasets.

The remainder of this paper is organized as follows. Section 2 reviews related work on the existing community detection methods and user interaction analysis for OSNs. Section 3 defines the problem of this work. Section 4 presents the overall framework of our user interaction-oriented community detection method and describes the detailed steps of the method. Section 5 presents our experimental design and the results of our evaluation analysis. Section 6 concludes the paper.

2. Related work

2.1. General community detection in networks

Over the past decade, substantial research effort has been devoted to community detection in networks [12,13]. An earlier body of research focused on identifying disjoint communities (i.e., a node can only fall within a single community). Popular approaches include graph partitioning and clustering [12,39], edge betweenness division [16,33], modularity-based optimization [6,30,31], label propagation [18,44] and nonnegative matrix factorization [27,41]. More recently, many studies [23,43] have revealed that overlapping communities are prevalent in real-world social networks (e.g., OSNs). Thus, a growing research trend is to focus on the design of overlapping community detection methods, such as clique percolation [11,32,37], link partitioning [2,10], extended label propagation [42,45] and local expansion and optimization [25,38]. Our work concentrates on the design of an overlapping community detection method for OSNs.

Among the existing overlapping community detection approaches, one branch adopts a general work flow where overlapping sets of user groups are first extracted from the original social graph and a super graph is then constructed with each node representing an extracted user group. Since a user can simultaneously be in multiple user groups, disjoint community detection algorithms can be applied to super graphs. In the clique percolation method (CPM) [11,32], the user groups are fully connected subgraphs (i.e., cliques) extracted from the original network. In link partitioning [2,10], two nodes connected by a link are considered to be a user group and the super graph is the line graph. In such super graph approaches, super nodes represent some high-level features of users and the links between super nodes reflect the relationships among users at the group level. Compared to traditional approaches that perform community detection on the original social network, super graph approaches have demonstrated superior flexibility and performance for overlapping community detection in several benchmark networks [43]. However, extraction of user groups in these approaches is based solely on the connectivity (e.g., friendship in OSNs) of a social network. Our work leverages knowledge of user interaction behaviors and investigates the cascading nature of user interactions to extract user groups. Thus, we believe that the results of our method can better reflect *active* communities in OSNs.

2.2. User interaction and interaction-based community detection

2.2.1. User interaction analysis in OSNs

User interactions have recently been shown to be an important feature of OSNs [15,22,40]. In [40], Wilson et al. conducted a social network analysis of Facebook based on *interaction graph* and showed that the network properties revealed from interaction-based analysis are greatly different than those revealed from connection-based analysis. They validated the usefulness of interaction-based analysis with several social network applications, such as a network security measure to prevent Sybil attacks. Jiang et al. [21] investigated *latent* interactions (e.g., profile browsing) which cannot be easily observed by traditional measurements, and constructed latent interaction graphs to analyze this type of interaction in the Renren online social network. However, these interaction-based social network analyses have not yet been applied for community detection.

2.2.2. Interaction-based community detection

Interaction-based community detection in OSNs is an emerging research area. Most of the existing methods leverage user interaction data to weight the edges in the social network graph for community detection. For instance, Dev et al. [9] considered the number of interactions between two users, the interaction types and the common neighbors interacting with both users to quantify edge weights and then applied hierarchical clustering on the constructed graph to detect non-overlapping communities. Chen et al. [7] also considered different types of interactions and the number of each type to measure the edge weight between two users, and proposed a heuristic community detection method based on an improved version of modularity to detect non-overlapping communities in dynamic networks. In contrast to these two works that focused on the detection of non-overlapping communities, Darmon et al. [8] proposed an interaction-based weighting scheme applied in an overlapping community detection algorithm.

The key difference between the aforementioned methods and the present work is that the former mainly considered pairwise interactions between users in a social network. Our work not only considers such pairwise interactions but also aims to capture how related interactions (e.g., retweets of a Twitter post) are propagated within the network. We believe that the propagation pattern of related interactions provides a strong indication of community structure. Thus, our work utilizes the proposed event graph to capture the propagation effect occurring within each social object sharing and gathers a collection of event graphs to form a super graph for community detection. Our super graph approach is also methodologically different from the aforementioned methods.

Some pervious works have aimed to combine user interaction analysis with topic-centric analysis for community detection. For example, Sachan et al. [34] utilized information about the discussed topics, types of user interactions and social connections to discover user communities. Lim and Datta [26] analyzed the celebrities that a user follows to determine each user's interests, making use of both user interests and interactions to design a community detection method. While these existing methods can identify topically meaningful communities, a certain level of semantic analysis (e.g., identification of topics discussed in user interactions) is needed. This may require some domain knowledge and introduces an additional computational cost. Our proposed method does not rely on semantic analysis. How to integrate a topic-centric method with our user interaction-oriented method is beyond the scope of the present work.

3. Problem definition

In this section, the definitions relevant to the design of our method are first introduced and our problem of community detection based on user interaction data in OSNs is then formulated. Notations frequently used in this paper are listed in Table 1.

In an online social network, we refer to the information shared (e.g., a blog post, a photo share) by a user as social object sharing s. An interaction that occurs in the given OSN is quantified by a four-tuple $r = (u^{\mathrm{a}}, u^{\mathrm{d}}, s, p)$, where u^{a} is the user who initiates the interaction, u^{d} is the user that u^{a} interacts with, s is the social object sharing that interaction r is associated with and p indicates the interaction type (i.e., direct or indirect).

For direct interaction, u^{d} of the interaction is the user who created the social object sharing. That is, the interaction is performed directly on the social object sharing (e.g., a "like" or comment on a photo share). To account for the cascading nature of user interactions, indirect interactions are also considered where u^{d} is not the user who creates the social object sharing but the user who is u^{a} of a previous direct or indirect interaction associated with the same social object sharing. That is, the interaction is performed on top of a previous interaction (e.g., a comment on a "comment on a photo share", a comment on a retweet of a post). Our community detection method is based on analysis of both direct and indirect user interaction data, as we assume that the cascading nature of the information shared among different users provides a strong implication of community structure.

In a given OSN, a set of social object sharings $S = \{s_1, s_2, \ldots, s_n\}$ generated by different users is first randomly selected. For each selected social object sharing s_i in S, all of the interactions (i.e., both direct and indirect) associated with s_i are

Table 1
The frequently used notations.

Name	Summary
s	A social object sharing extracted from the crawled OSN data
r	A user interaction
u^a	A user that initiates an interaction
u^d	A user that an interaction is directed to
CE	The cascading event
U_i^a	The set of users who act as u^a in the cascading event i
U_i^d	The set of users who act as u^d in the cascading event i
$c_{i,j,k}$	The total number of interactions initiated by u_j^a and directed to u_k^d in the cascading event i
EG_i	The event graph corresponding to the cascading event i
V_i	The node set of the event graph i
E_i	The edge set of the event graph i
$w_{I_{uv}}^i$	The interaction weight between users u and v in the event graph i
$w_{G_{uv}}^i$	The group behavior weight between users u and v in the event graph i
α	The balancing parameter
w_{uv}^i	The overall weight of the edge between users u and v in the event graph i
K_u^i	The total sum of the weights of all of the edges connected with user node u in the event graph i
W^i	The total sum of weights of all of the edges in the event graph i
se	The sub-event
SG	The super graph
ϵ	The cut-off threshold parameter
l	The level parameter

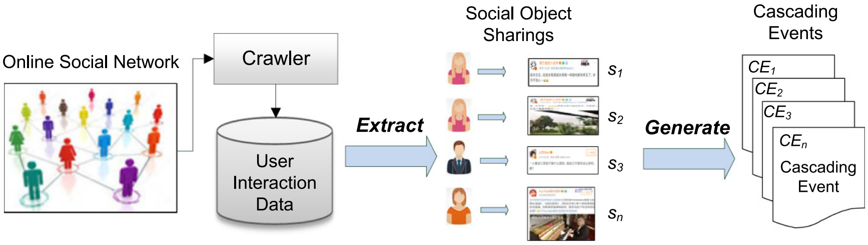


Fig. 1. The work flow of data preparation.

crawled. To quantify these interactions, *cascading event* is defined as the following matrix:

$$CE_i = \begin{bmatrix} c_{i,1,1} & c_{i,1,2} & \cdots & c_{i,1,W} \\ \cdots & \cdots & \cdots & \cdots \\ c_{i,H,1} & c_{i,H,2} & \cdots & c_{i,H,W} \end{bmatrix}, \tag{1}$$

where $c_{i,j,k}$ is the number of interactions initiated by a user u_j^a and directed to u_k^d in a given sharing s_i, $W = |U_i^a|$ is the number of users who act as u^a in s_i and $H = |U_i^d|$ is the number of users who act as u^d in s_i. Note that $c_{i,j,k} = 0$ in the case that $j = k$ (i.e., the positive diagonal of matrix CE_i). The work flow of the data preparation process for our interaction-oriented community detection method is shown in Fig. 1.

As shown in Fig. 1, the final output after data preparation is a series of cascading events each of which describes the interacting relationship of different users and counts the number of interactions occurring between the users in a given social object sharing. In this work, we assume that communities have already been formed in a given OSN dataset. These cascading events thus serve as the essential component of our method to reveal the existing community structure. The rationale is that if two or more users in the same community happen to come across a social object sharing of common interest, they are more likely to interact with each other around this social object sharing. Thus, we believe that users who frequently interact within the cascading event of a social object sharing may potentially belong to the same community.

Our problem: For all social object sharings, a set of cascading events $\Theta = \{CE_1, CE_2, \ldots, CE_n\}$ is formed. Our interaction-based community detection method takes the crawled interaction data associated with a set of social object sharings S and the cascading event list L as the main sources of input. The problem is to find h overlapping communities $C_i (1 \le i \le h)$ where each C_i is a subset of the union set $\{U_1^a \bigcup U_1^d \bigcup \cdots \bigcup U_n^a \bigcup U_n^d\}$.

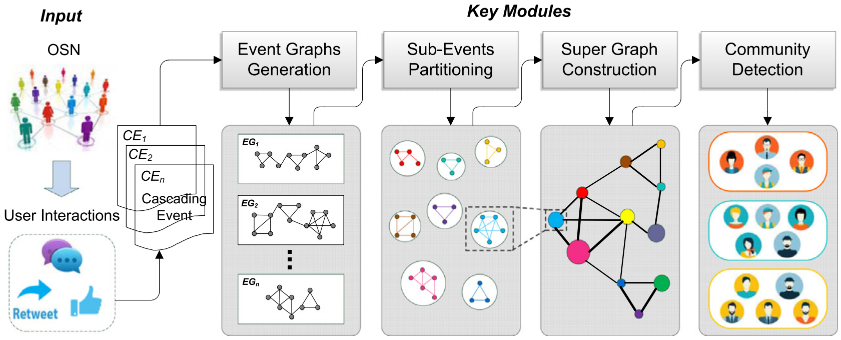


Fig. 2. The overall framework of ICDCA.

4. User interaction-oriented community detection based on cascading analysis

4.1. Framework

The overall framework of the user interaction-oriented community detection based on cascading analysis (ICDCA) method is shown in Fig. 2. The inputs of the system are the series of cascading events extracted from the user interaction data of an OSN. The overall community discovery process is divided into four key steps: *event graphs generation, sub-events clustering, super graph construction* and *community detection*.

(1) Event graphs generation: The first step is to generate so-called event graphs for all cascading events crawled from an OSN. For each cascading event, an undirected and weighted event graph is generated where the nodes represent all of the users who have interacted with one or more users and an edge is established if two users have interaction(s). The weight of an edge is used to quantify the strength of ties between users in terms of their interaction behaviors. The function of this step is thus to use the graph-based representation, namely the event graph, to depict the interaction relationships as well as the degrees of these interactions between different users within each cascading event.

(2) Sub-events partitioning: The second step is to partition the nodes in each event graph into a set of so-called sub-events, each of which is a subgraph with densely-connected nodes. Each sub-event therefore contains a group of users who interact more with each other and less with other users in a given event graph. Such sub-events represent the set of users involved in a social object sharing who may potentially belong to the same user community. To find such sub-events, the multistep greedy algorithm (MSG) [35] is applied on each event graph and a collection of sub-events generated from all event graphs is obtained. The function of this step is thus to identify groups of users, namely sub-events, in a given event graph that potentially belong to the same communities.

(3) Super graph construction: The third step is to construct a super graph in which each node represents a sub-event extracted from the previous step. The edge between two nodes in the super graph is established by checking the Jaccard similarity of the corresponding pair of sub-events. To reduce the required number of pairwise similarity checks, the locality sensitive hashing (LSH) method [3] is adopted for edge establishment in the super graph. The function of this step is thus to use the constructed super graph to examine the similarities between the sub-events partitioned over all event graphs.

(4) Community detection: In the last step, an existing community detection algorithm, namely the Louvain method [6], is applied to the constructed super graph to detect user communities based on the cascading events of their interaction behaviors. Each detected user community depicts a group of users who actively interacted with each other in multiple social object sharings. We believe that communities based on user interaction behaviors reflect the active communities that exist in OSNs. The function of this step is thus to merge similar nodes (i.e., sub-events) in the super graph to detect the user communities in a given OSN dataset.

4.2. Step 1: Event graphs generation

In our interaction-oriented community detection method, event graph is utilized to capture the users involved in a cascading event as well as the strength of ties between these users based on their interaction behaviors within that cascading event. Unlike existing user interaction behavior analyses [8,9,40] that typically construct a single large interaction graph based on all crawled user interaction data, separate small event graphs for the different cascading events are constructed in our method. The rationale is that the interactions that occur within a cascading event are relevant to each other (i.e., all are related to a social object sharing). Constructing an event graph per cascading event can capture this relevance, whereas the existing single graph approaches merge all interactions between two users regardless of the relevance of these interactions. We believe that relevant interactions bear some semantic correlations (e.g., all related to a sports topic) that are important for community detection.

Each event graph for a given cascading CE_i can be represented as $EG_i = (V_i, E_i)$ where $V_i = U_i^{\mathrm{a}} \bigcup U_i^{\mathrm{d}}$, that is, all of the users who are u^{a} or u^{d} of the interactions within cascading event CE_i, and E_i is the set of weighted undirected edges established between the nodes in V_i. To establish the edge between users in an event graph, two conditions are considered: (1) whether interaction(s) exist between the two users, and (2) whether the two users frequently interact with "common neighbor(s)" even if there is no interaction between the two users. An edge will be established between the two users if either of these two conditions is true. Correspondingly, the weight w^i_{uv} of an edge between two users u and v in event graph EG_i is determined by two parts: the interaction weight w^i_{Iuv} and the group behavior weight w^i_{Guv}.

The interaction weight w^i_{Iuv} is determined by the number of interactions between two users. Since user interactions are directional, $c_{i,\ u,\ v}$ and $c_{i,\ v,\ u}$ are used to denote the number of interactions from user u to v and from user v to u respectively in a given cascading event CE_i. The total number of interactions from both directions (i.e., u to v and v to u) is therefore obtained as: $c^*_{i,uv} = (c_{i,u,v} + c_{i,v,u})$. If $c^*_{i,uv}$ is greater than zero, an edge between user u and v is established in EG_i and the interaction weight w^i_{Iuv} is determined as:

$$w^i_{Iuv} = s\left(\left(\frac{c^*_{i,uv} - c^{\mathrm{min}}_i}{c^{\mathrm{max}}_i - c^{\mathrm{min}}_i} - \frac{1}{2}\right) \cdot 2\omega\right), \tag{2}$$

where $s(\cdot)$ represents the Sigmoid function, c^{min}_i and c^{max}_i are the minimum and maximum number of interactions, respectively, of two users among all of the user pairs in the given CE_i and ω is the scaling parameter, which controls the slope of the Sigmoid function and is empirically set to 5 in our experiments. Here, non-linear mapping (i.e., the Sigmoid function) is adopted to derive w^i_{Iuv}. That is, our formulation assigns higher weights to user pairs whose $c^*_{i,uv}$ is close to c^{max}_i and lower weights to user pairs whose $c^*_{i,uv}$ is close to c^{min}_i. By doing so, "active user pairs" that frequently interact with each other are given a higher weight while "inactive user pairs" who rarely interact even though they might be "friends" are given a lower weight.

The group behavior weight between two users is determined by exploring the "common neighbor" of these two users similar to Dev et al. [9]. A user is referred to as a "common neighbor" of users u and v if she/he interact with both u and v. According to Dev et al. [9], it is likely that two users belong to the same community if they have "common neighbor(s)". The group behavior weight of user u and v is therefore determined as:

$$w^i_{Guv} = \sum_{j=1}^{d} min(w^i_{Iug_j}, w^i_{Ivg_j})/d, \tag{3}$$

where $G_{uv} = \{g_1, g_2, \ldots, g_d\}$ represents the common neighbors of users u and v. The group behavior weight depends on the interaction weight of the two users with the common neighbor (i.e, the minimum of the two interaction weights) and takes the average across all common neighbors. If no common neighbor of u and v exists, w^i_{Guv} is set to zero.

To incorporate the impact of both the number of interactions and group behaviors, the overall weight of an edge between the two users u and v in event graph EG_i is determined as:

$$w^i_{uv} = \alpha * w^i_{I_{uv}} + (1 - \alpha) * w^i_{G_{uv}}, \tag{4}$$

where α is referred to as the balancing parameter and $0 \le \alpha \le 1$, $w^i_{I_{uv}}$ and $w^i_{G_{uv}}$ are the interaction weight and group behavior weight derived according to Eqs. (2) and (3) respectively. α is usually set to a value close to 1 (e.g., 0.7–0.8) such that the interaction weight constitutes the major part of the overall weight. Note that if both $w^i_{I_{uv}}$ and $w^i_{G_{uv}}$ are zero, there will be no edge between u and v.

For each cascading event, an event graph is generated where the edge weights are determined by the above equations. Consequently, a set of event graphs is obtained for all cascading events crawled from an OSN. These generated event graphs are then used as the inputs in step 2 of our method.

4.3. Step 2: Sub-events partitioning

Once the event graphs for different cascading events have been generated, each event graph is partitioned into a set of subgraphs, namely sub-events, in which nodes are densely connected. The nodes within each sub-event thus represent a group of users that strongly interacted with each other. Here, we assume that members in the same community usually have some common interests (e.g., sports) and that the social object sharing (e.g., a sports news item) associated with the event graph may be related to the common interest of the community. Therefore, when a social object sharing is of interest to the community, the users in that community are more likely to interact with the given social object sharing. The sub-event is thus used to represent users who potentially belong to the same community within the event graph.

To partition each event graph into such sub-events, the mutlistep greedy algorithm (MSG) [35], which is an efficient modularity optimization method that can support simultaneous searching of partitions that maximize modularity in a given network, is adopted. Due to its low complexity requirement and parallel nature, the MSG algorithm is used to identify sub-events from the set of event graphs generated in the previous step. A simple description of MSG for sub-event partitioning is shown in Algorithm 1. For further details about this algorithm, please refer to [35].

Algorithm 1 Multistep greedy algorithm for sub-event partitioning.

```
Input: Event graphs $\{EG_1, EG_2, \ldots, EG_n\}$, level parameter $l$.
Output: A set of sub-events $SUB = \{se_1, se_2, \ldots, se_m\}$.
$SUB \leftarrow \emptyset$;
for $i = 1$; $i \leq n$; $i++$ do
    Make each node in $EG_i$ as a cluster;
    Calculate modularity change $\Delta Q^i_{j,k}$ for each pair $(j, k)$ of connected clusters in $EG_i$;
    while pair$(j, k)$ with $\Delta Q^i_{j,k} > 0$ exists do
        Sort $\Delta Q^i_{j,k}$ for all pairs $(j, k)$ in a decreasing order;
        if $\Delta Q^i_{j,k}$ is within the top $l$ values of modularity changes and clusters $j$ and $k$ have not been merged yet then
            Merge cluster $j$ and $k$;
        end if
    end while
    $SUB = SUB \bigcup$ {all of the final merged clusters from $EG_i$};
end for
```

The Algorithm 1 takes inputs of the event graphs generated in the previous step and the level parameter l and produces a set of sub-events. For each event graph, the algorithm adopts a bottom-up approach where each node in the event graph is initially treated as a cluster (i.e., a sub-event). In every iteration, the modularity change (i.e., $\Delta Q^i_{j,k}$) is evaluated for a pair of connected clusters j and k. $\Delta Q^i_{j,k} > 0$ is calculated for all such pairs and sorted in decreasing order. If $\Delta Q^i_{j,k}$ is among the top l values in the sorted list and clusters j and k have not yet been merged, they are merged in the current iteration. This iterative process continues until all pairwise merges cause a decrease in the modularity change. Note that the level parameter l is set to always be smaller than the number of edges in the graph.

As noted above, partitioning of sub-events is based on maximization of modularity, which is the most widely used metric to evaluate the quality of a particular partitioning of a network into clusters. Since our event graph is weighted and undirected, the following classical modularity evaluation metric Q^i [29] is used for the event graph EG_i:

$$Q^i = \frac{1}{2W^i} \sum_{u,v \in V} (w^i_{uv} - \frac{K^i_u \cdot K^i_v}{2W^i}) \delta(c^i_u, c^i_v), \tag{5}$$

where W^i is the total sum of weights of all of the edges in EG_i, $w^i_{i,j}$ is the weight of the edge between user nodes u and v in EG_i, K^i_u (K^i_v) is the total sum of weights of all of the edges connected with user node u (v), and $\delta(c^i_u, c^i_v) = 1$ if the clusters of node u and v (i.e., c^i_u and c^i_v) are the same, and 0 otherwise.

It should be noted that each sub-event contains the users that potentially belong to the same community and participate in the interactions (direct or indirect) associated with a given social object sharing. Thus, each sub-event is not a complete community for a given dataset; rather, it is a subset of users in a community. Members of a community may be involved in user interactions over multiple social object sharings. Thus, it is necessary to cluster sub-events from multiple event graphs to merge members of a community scattered in different sub-events. This will be achieved in steps 3 and 4 of our method.

4.4. Step 3: Super graph construction

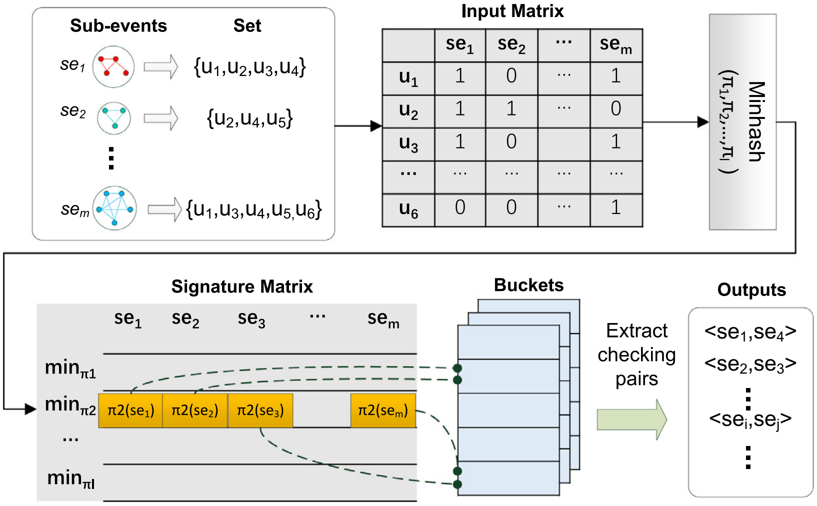
In this step, a super graph is constructed in which the nodes are the sub-events extracted from all event graphs. In the super graph, the establishment of an edge between two nodes is based on the similarity between the two nodes, which is measured based on the Jaccard similarity as follows:

$$sim(se_i, se_j) = \frac{\{u|u \in se_i\} \cap \{v|v \in se_j\}}{\{u|u \in se_i\} \cup \{v|v \in se_j\}}, \tag{6}$$

where $\{u|u \in se_i\}$ represents the set of all of the users contained in sub-event se_i, and $\cap$ and $\cup$ are the set operators of intersection and union, respectively. The Jaccard similarity measure yields a higher value (i.e., closer to 1) when there are more common users within two sub-events. In our super graph, an edge between two nodes se_i and se_j is established when $sim(se_i,\ se_j)$ is greater than the cut-off threshold ϵ.

Constructing such a super graph requires a pairwise similarity check between every sub-event pair. The time complexity is therefore $O(m^2)$ where m is the total number of sub-events. This brings a substantial computation overhead when the number of sub-events is large (e.g., thousands of sub-events). Therefore, the locality sensitive hashing (LSH) method [3] is adopted to reduce the number of pairwise similarity checks required to construct the super graph. LSH is an efficient algorithm for finding similar items in high-dimensional space. In our case, the LSH algorithm is used to generate "checking pairs" of sub-events and Jaccard similarity checks among all pairs of sub-events are performed.

Fig. 3 shows the general workflow for applying LSH. As shown in Fig. 3, each sub-event is first converted into a set representation in which the elements are the users in the sub-event. Next, an input matrix is constructed where the rows



	se_1	se_2	…	se_m
u_1	1	0	…	1
u_2	1	1	…	0
u_3	1	0		1
…	…	…	…	…
u_6	0	0	…	1

Fig. 3. The LSH workflow for generating checking pairs.

represent all users in all sub-events and the columns represent sub-events (note that in the actual implementation, it is not necessary to construct an input matrix because random hash functions are typically used for min-hashing). A min-hashing strategy containing the set of permutation functions $\pi_1, \ldots, \pi_k$, is then applied to the input matrix, which transforms the input matrix into a signature matrix. According to Gionis et al. [3], if the signature of two sub-events $\pi(se_i)$ and $\pi(se_j)$ are the same, $sim(se_i, se_j)$ must be greater than zero. We thus consider two sub-events with the same signature value to be a “checking pair” (i.e., a pair of sub-events necessary to calculate the Jaccard similarity for edge establishment). To extract all such pairs, different buckets (i.e., hash table) are created for different permutation functions, and sub-events with the same signature value are put into the same bucket. The output of LSH is the checking pairs extracted from these buckets. Detailed implementation of MinHash LSH can be found in [1]. After the checking pairs are obtained, it is only necessary to establish the edges in the super graph between the sub-events in each checking pair.

4.5. Step 4: Community detection

The last step of our method is to identify the OSN user communities from the constructed super graph. Recall that each node in our super graph represents a sub-event containing a set of users that densely interacted with each other about a social object sharing. Therefore, an existing community detection algorithm is applied to our super graph to search for clusters of sub-events with high similarity. Each cluster contains common users who densely interacted in multiple social object sharings and we believe that such clusters represent a user community in an OSN from a user interaction perspective. Note that because different sub-events (i.e., the nodes in our super graph) may contain the same users, overlapping communities can be detected even though a non-overlapping community detection algorithm is applied to our super graph.

Since our super graph is relatively large in size compared to our event graph, the Louvain method proposed by Blondel et al. [6] is adopted. The Louvain method is a highly-efficient community detection algorithm that supports finding communities in large graphs (e.g., graphs with up to 10^9 edges). As shown in Algorithm 2, the Louvain method takes the input of super graph SG_0 constructed in the previous step. All nodes of super graph SG_0 are initially treated as different communities. The algorithm then iteratively repeats two phases for community detection.

The first phase reassigns a different community to each node of the graph. For each node i, the algorithm selects the community of a neighboring node j that achieves the largest modularity gain (i.e., ΔQ_l^{max}). Note that the same modularity measure as in Eq. (5) is used, except that the graph here is the super graph and not the event graph. The first phase ends when further node reassignment to another community cannot improve the modularity. The second phase aggregates the communities found in the first phase as super nodes and constructs another super graph (i.e., SG_{l+1}) on top of the existing super graph SG_l at iteration l. The weight of the edges between super nodes in SG_{l+1} is the sum of the weights of the edges between the represented communities in SG_l. The two phases are repeated until no further modularity gain can be obtained. Given the super graph obtained in the last iteration, the function *RetrieveCommunities*() in Algorithm 2 disaggregates the nodes in the super graph into a union set of all users contained in each detected community.

4.6. Summary and analysis of our ICDCA method

To summarize, Algorithm 3 shows our interaction-oriented community detection based on cascading analysis that includes the four key steps previously described. The time complexity of our method is analyzed by considering three major computational steps: the MSG algorithm for sub-event partitioning (line 5), construction of a super graph (line 9) and the

Algorithm 2 Louvian method for community detection.

```
Input: The super graph $SG_0$ obtained from our previous step.
Output: A set of detected user communities.
$l \leftarrow 0$
for each iteration $l$ do
    Make each node in $SG_l$ as a community;
    $isModGain \leftarrow false$;
    while reassignment of node $i$ can increase the modularity of $SG_l$ do
        $\Delta Q_l^{\max} \leftarrow 0, j^{\max} \leftarrow i$;
        for each neighbor $j$ of node $i$ do
            if reassign $i$ to community of $j$ with $\Delta Q_l^{i,j} > \Delta Q_l^{\max}$ then
                $j^{\max} \leftarrow j$;
                $isModGain \leftarrow true$
                $\Delta Q_l^{\max} \leftarrow \Delta Q_l^{i,j}$;
            end if
        end for
        Move $i$ to the community of $j^{\max}$ in $SG_l$;
    end while
    if $isModGain == true$ then
        Aggregate the updated communities in $SG_l$ as super nodes and construct super graph $SG_{l+1}$;
        $l \leftarrow l + 1$;
    else
        return $RetrieveCommunities(SG_l)$;
    end if
end for
```

Algorithm 3 Interaction-oriented community detection based on cascading analysis.

```
1: Input: A set $\Theta$ of cascading events extracted from an OSN dataset
2: Output: user communities detected
3: Generate event graph $EG_i$ for each cascading event $CE_i$.
4: for each event graph $EG_i$ in $\{EG_1, EG_2, \ldots, EG_n\}$ do
5:     Call MSG algorithm (Algorithm 1) to partition $EG_i$ into sub-events.
6:     $SUB = SUB \cup$ {all of the sub-events from $EG_i$}.
7: end for
8: Apply LSH method to generate checking pairs for constructing super graph from $SUB$.
9: Construct super graph $SG_0$.
10: Call Louvian method (Algorithm 2) to detect user communities from $SG_0$.
11: return the communities detected.
```

Louvain method for community detection from the super graph (line 10). The MSG algorithm converges in a finite number of iterations. The time complexity of the algorithm is $O(hdlog(v))$ where h is the total number of iterations, d is the number of edges in a given event graph and v is the number of nodes in a given event graph. Given that there are n event graphs, the total computational complexity of obtaining all sub-events is $O(nhdlog(v))$. To construct the super graph, the time complexity is $O(m^2)$ if the LSH method is not applied, where m is the total number of sub-events. By applying LSH, the time complexity is reduced to $O(p)$, where p is the total number of checking pairs extracted. As for the Louvain method for community detection on super graph, the time complexity is proportional to the number of edges in the super graph (i.e., $O(p)$). Therefore, the time complexity of our method is $O(nhdlog(v)) + p)$.

5. Experiment and analysis

5.1. Experimental datasets

Most publicly available datasets of online social networks contain only users' connection data or have limited interaction data, making them insufficient for use with our ICDCA method. Therefore, we collected real data of three OSNs that involve user interactions. The three datasets are as follows.

LiveJournal dataset: LiveJournal (LiveJ for short) is a popular social network platform that supports the services of blogging and forums. In LiveJournal, each user maintains a homepage where they can post blog, journal or diary entries. Other users can interact with the user by commenting on the user's post or commenting on their "comments". One unique feature of LiveJournal is that it provides a "community" functionality to collect relevant blogs from different users. Users can then join

Table 2
Statistics of the three datasets after pre-processing (social.obj, and int. are the short forms of the social object sharings and interactions respectively).

Dataset	No. of social.Obj	No. of user	No. of int.	No. of direct int.	No. of indirect int.	Avg. no. of int. per user
LiveJ	1130	3987	49,113	10,237	38,876	31.8
Weibo	237	6216	25,296	1564	23,732	8.79
Tieba	637	8444	77,196	66,260	10,936	18.3

such “communities” based on their interests. Thus, “community” information can be considered as the ground truth in this dataset.

The collected LiveJournal dataset includes 1130 social object sharings (i.e., blog and journal posts) created by 352 users. Both direct and indirect interaction data associated with these social object sharings were crawled. As a result, 49,113 interactions and 3987 users (as either u^a or u^d) who participated in these interactions were obtained. We also examined the “communities” that these users joined, which we consider to be the ground truth.

Sina Weibo dataset: Sina Weibo (Weibo for short) is a popular Chinese microblogging network where users can share short texts, photos, video, etc. In constrast to OSNs like Facebook and WeChat, Weibo users can perform interactions such as comments, likes and reposts on other users’ social object sharings or interactions even if they do not have a connection (i.e., a “friend”). Thus, this OSN is more centered on user interactions than user connections.

The collected Weibo dataset includes 265 social object sharings (i.e., microblog posts) created by 195 users (note that only a small Weibo dataset could be obtained due to constraints set by the service provider). Both direct and indirect interaction data associated with these social object sharings were crawled. As a result, 30,149 interactions and 7825 users who participated in these interactions were obtained. Even though the Weibo dataset has less social object sharings, there are more users participating in the corresponding interactions.

Baidu Tieba dataset: Baidu Tieba (Tieba for short) is a forum-like OSN hosted by the Chinese Internet company Baidu. Unlike traditional forum, Tieba is organized into different forums known as “bars” based on users’ common interests. A user can explore the bars associated with their interests and post blogs and comments on others’ blogs in these bars. Compared to LiveJournal and Weibo, the user interactions on Tieba are more interest-driven.

The collected Tieba dataset includes 637 social object sharings (i.e., blog posts) created by 472 users. Both direct and indirect interaction data associated with these social object sharings were crawled. As a result, 77,196 interactions and 8444 users involved in these interactions were obtained.

After obtaining the above datasets, some pre-processing operations were performed prior to organizing the user interaction data into cascading events. Firstly, social object sharings with no interaction were removed. This small set of social object sharings is not useful since our method relies on interaction data analysis. Secondly, it is necessary to remove social object sharings that involve too many interactions in order to avoid biased seeds (e.g., social object sharings posted by famous celebrities). To do so, the statistical method based on interquartile range (IQR) was used. Specifically, our IQR is the difference between the first quartile and third quartile of the number of user interactions per social object sharing across the crawled dataset. The threshold for removing social object sharings with too many interactions was thus set as the third quartile plus 1.5 times the IQR. As a result, 2.3%, 3.8% and 5.2% of social object sharings were removed from the original LiveJournal, Weibo and Tieba datasets respectively. After pre-processing, each social object sharing and its associated user interaction data are organized into the form of a cascading event. Table 2 summarizes the statistics of the three datasets after pre-processing.

For each social object sharing, we generated the corresponding cascading event as defined in Section 3. To investigate how the generated cascading events are related to user communities, we analyzed the users involved in each cascading event belonging to the ground truth communities in the LiveJournal dataset. On average, 73.8% of users in each cascading event belonged to one or more communities and 54.3% of user interactions in each cascading event were performed between users belonging to the same community. This supports our assumption that users in the same community are more likely to participate in cascading events and have more interactions compared to users not belonging to that community.

For comparison with general community detection algorithms and evaluation of the results, the interaction graph proposed in [40] was constructed for each crawled dataset. The interaction graph is undirected and unweighted, where a node represents a user and an edge is established if and only if the number of interactions between two users (i.e., nodes) exceeds the threshold value. Without loss of generality, the threshold value was set to 1 to preserve all interaction information between different users. Table 3 summarizes the statistics of the three interaction graphs constructed from the three datasets.

5.2. Compared methods and parameter settings

As discussed in Section 2.2, the literature on interaction-based community detection is still limited especially existing methods to support overlapping community detection. Thus, three state-of-the-art methods are selected for comparison with our proposed ICDCA: SLPA (aka GANXiS) [45] and NISE [38], which are general overlapping community detection methods that can be applied to different types of networks, and iOSLOM [8], an interaction-based method that can detect overlapping communities based on user interactions in OSNs. In order to use SLPA and NIST for user interaction-based community de-

Table 3
Statistics of the interaction graphs constructed from the three datasets (diam. is the short form of the diameter which is the longest path length among all pairs of nodes in a given interaction graph. Path.len is the short form of path length which is the length of the shortest path from one node to another).

Dataset	No. of nodes	No. of edges	Avg. node degree	Diam.	Avg.Path.Len.
LiveJ	3987	26,268	13.7	10	3.87
Weibo	6216	13,924	4.48	20	6.76
Tieba	8444	25,190	5.96	7	3.91

tection, the interaction graph (see Section 5.1) was constructed from our dataset and used as the underlying graph structure in these two methods. A brief description of these three methods is as follows.

- SLPA [45]: This method is an efficient overlapping community detection algorithm based on the label propagation strategy in which a node in a network dynamically updates its label based on its neighbors' majority labels and the node is assigned to a community based its label(s) when the algorithm converges. The key advantage of SLPA is that it scales linearly with the number of edges in the network. In [43], SLPA was shown to achieve a superior and stable performance for both LFR synthetic network and real-world networks compared to 13 overlapping community detection algorithms.
- NISE [38]: This method is an efficient overlapping community detection algorithm that uses a seed expansion approach to find some seed nodes in a network and then greedily expand these seeds to detect communities. NISE can be applied to very large-scale networks with millions of edges. It has been shown to outperform other state-of-the-art methods in terms of cohesiveness of communities and ground-truth accuracy.
- iOSLOM [8]: In [8], an interaction-based approach is proposed to construct a weighted and directed network graph based on user interaction data, and the OSLOM [25] algorithm is used to detect overlapping communities from such a graph. Therefore, we refer to this method as iOSLOM (i.e., i stands for interaction) in this paper. To construct an interaction-based network graph, iOSLOM weights the edges in the graph with a measure proportional to the number of interactions of different types between users. The measure considers the relative importance of interactions by normalizing the number of interactions made by user u to f over the total number of interactions of that type made by user u to all users. The normalized values of all interaction types are then averaged to obtain the final weight. According to Darmon et al. [8], iOSLOM can distinguish users who actively interact from the inactive ones and shows very different community characteristics from those detected from structural networks.

For the parameter settings for these three methods, a similar strategy as in [43] was adopted where a range of values for each tunable parameter was explored. The settings with the best results were used. For SLPA, the parameter r, which defines the threshold for deleting the label of a node, varies from 0.05 to 0.5 with an interval of 0.05. Since SLPA is nondeterministic, 10 runs of each method were performed on each dataset. For NISE, one of two seeding strategies, “Graclus centers” or “Spread hubs”, must be selected to run the algorithm. Both strategies were used for each dataset, and the one giving the best results was chosen. NISE also requires specification of the number of communities k as an algorithm input. The numbers of communities returned by ICDCA, SLPA and iOSLOM were used to estimate the lower and upper bounds of k and the experiments were repeated to obtain the best results within the estimated bounds. For iOSLOM, two levels of hierarchy were considered.

For our ICDCA method, three key parameters must be set in steps 1, 2 and 3: the balancing parameter α, the level parameter l and the cut-off threshold ϵ. The values of these parameters were set as follows:

The parameter α controls the contribution of interaction weight and group behavior weight when determining the overall weight of an edge in the event graph, as shown in Eq. (4). In principle, if a given OSN dataset contains mostly direct interactions, α can be set to a small value to facilitate exploitation of indirect relationships between users via group behaviors. Based on analysis of the three datasets, the LiveJournal dataset contained the most direct interactions. Therefore, α was set as 0.3 for the LiveJournal dataset and 0.7 and 0.6 for the Weibo and Tieba datasets, respectively.

For the parameter l, according to Schuetz and Caflisch [35], the MSG algorithm typically achieves optimal performance when $l < \sqrt{d}$ where d is the number of edges in the graph that the algorithm is applied to. In our experiments, we thus set $l = 0.25\sqrt{d^{EG}}$ according to the number of edges (i.e., d^{EG}) of a given event graph.

The parameter ϵ controls the number of edges established in the super graph (i.e., the smaller ϵ is, the more edges there will be in the super graph). Having more edges in the super graph helps retain more relationships between sub-events, which may in turn affect the community detection results. We thus investigated how different values of ϵ affected the performance of our method (see Section 5.4.4). Based on our investigation, ϵ was set as 0.01, 0.03 and 0.01 for the LiveJournal, Weibo and Tieba datasets, respectively.

5.3. Evaluation metrics

5.3.1. Metrics without ground truth

To evaluate all three datasets without ground truth data, the detected communities are evaluated from two aspects: modularity and interaction degree. Modularity is the most commonly used metric to evaluate the “goodness” of communities and is based on the idea that nodes within a community should have more connections with each other than with nodes outside the community. To support the evaluation of overlapping communities, the extended modularity *EQ* proposed in [36] is adopted as follows:

$$EQ = \frac{1}{2m}\sum_{i}\sum_{u,v\in C_i}\left[A_{uv} - \frac{k_u k_v}{2m}\right]\frac{1}{O_u O_v}, \tag{7}$$

where m is the total number of edges in a given graph, u, v denotes any two nodes in the same community C_i as node i, $A_{uv} = 1$ if there is an edge between u and v, and is 0 otherwise, k_u is the degree of node u, and O_u is the number of communities to which node u belongs. It should be noted that *EQ* is evaluated on the interaction graph as described in Section 5.1 where the edges represent the existence of interactions between users.

Since the interaction graph is unweighted, the modularity *EQ* only reflects the structural cohesiveness of users in the detected communities. It is thus also necessary to evaluate the frequency of interactions between users. In principle, users in the same community usually have more frequent interactions with each other compared to interactions with users outside the community. The interaction degree *ID* is thus defined as follows:

$$ID = \sum_{C}\frac{N_C}{N_{\text{all}}}\cdot\frac{I_C^{\text{in}}}{I_C^{\text{in}} + I_C^{\text{out}}}, \tag{8}$$

where C is a community in all detected communities, N_C is the number of users in community C, N_{all} is the sum of the number of users in all communities, I_C^{in} is the total number of interactions involving two users both in C, and I_C^{out} is the total number of interactions involving one user in C and one user not in C.

To account for the modularity and interaction degree using one metric, an F-score like evaluation metric, *MI-score*, is proposed as follows:

$$\textit{MI-score}_{\beta} = (1+\beta^2)\cdot(EQ\cdot ID)/(\beta^2\cdot EQ + ID), \tag{9}$$

where $\beta \in [0, \infty]$ is a parameter to adjust the weight of modularity *EQ* and interaction degree *ID*. When $\beta = 1$, the *MI-score* is considered as the harmonic average of *EQ* and *ID*. When $\beta > 1$, the metric puts greater emphasis on *ID* than *EQ*, and vice versa.

5.3.2. Metrics with ground truth

To evaluate the LiveJournal dataset based on ground truth data, two existing evaluation metrics are used to measure the detection accuracy: the overlapping Normalized Mutual Information (*NMI*) and Omega index (*OI*).

NMI is a metric based on information theory that assesses the similarity of two clusters. It has been widely used to compare the detected communities with the ground truth communities for non-overlapping community detection. To apply *NMI* for evaluating overlapping communities, the extended NMI_{op} metric proposed in [24] is adopted. Let $C = \{C_1, C_2, \ldots, C_k\}$ denote the communities returned by a community detection method and $C' = \{C'_1, C'_2, \ldots, C'_l\}$ be the ground truth communities. Whether a node i belongs to a community C_k can be modeled as a random variable X_k with a probability distribution given by $P(X_k = 1) = N_k/N$, $P(X_k = 0) = 1 - P(X_k = 1)$, where $X_k = 1$ indicates that node i belongs to C_k, N_i is the number of nodes in C_i and N is the total number of nodes. The same holds for the random variable Y_j associated with C'_j in C'. Vectors of random variables X and Y can be formed for C and C', respectively. Entropy $H(X)$ and conditional entropy $H(X|Y)$ can then be derived from the probability distributions of X and Y (for detailed derivations refer to [24]). The overlapping NMI for C and C' is then given by:

$$NMI_{\text{op}}(X|Y) = 1 - [H(X|Y) + H(Y|X)]/2, \tag{10}$$

where $NMI_{\text{op}} \in [0, 1]$ with $NMI_{\text{op}} = 1$ indicates perfect matching.

The Omega index is the overlapping version of the Adjusted Rand Index [20]. It evaluates pairs of nodes that are partitioned into the same number of communities in C and C'; in other words, the Omega index counts how many pairs of nodes are not placed in any community, how many are placed in exactly one community, how many are placed in exactly two communities and so on. Given the set of communities C detected by a community detection method and the set of ground truth communities C', the Omega index is calculated as follows:

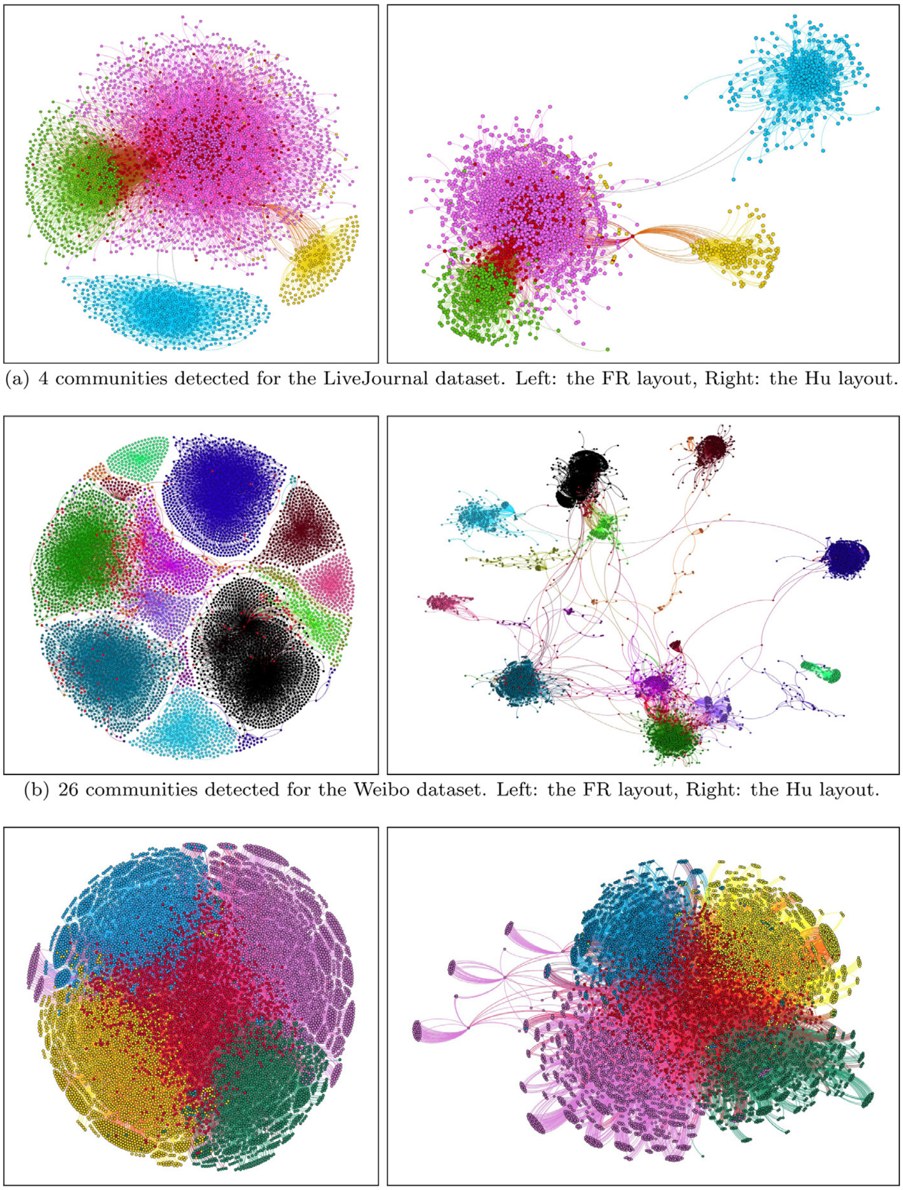
$$\omega(C, C') = \frac{\omega_u(C, C') - \omega_e(C, C')}{1 - \omega_e(C, C')}, \tag{11}$$

where ω_u is the unadjusted Omega index that denotes the fraction of pairs that occur together in the same number of communities in C and C', and ω_e is the expected *Omega* index that represents the expected value of the same fraction in the null model (please refer to [17] for the calculation of ω_u and ω_e).

5.4. Results and analysis

5.4.1. Detection results

Having applied our ICDCA method to the three datasets, the visualizations of the detected communities using the Gephi tool [5] are first presented. Two force-directed layouts, namely the “Fruchterman and Reigold” layout [14] (FR layout for



(a) 4 communities detected for the LiveJournal dataset. Left: the FR layout, Right: the Hu layout.

(b) 26 communities detected for the Weibo dataset. Left: the FR layout, Right: the Hu layout.

(c) 4 communities detected for the Tieba dataset. Left: the FR layout, Right: the Hu layout.

Fig. 4. Visualization results of the communities detected by ICDCA. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

short) and "Yifan Hu" layout [19] (Hu layout for short), are used to visualize the results of each dataset as shown in Fig. 4. The FR layout is a classical force-directed layout based on a system of mass particles. It promotes even distribution of nodes and has a few edge crossings, which allow close observation of node interactions within each community. The Hu layout combines a force-directed model with a graph coarsening technique for efficient visualization of large graphs. Compared to the FR layout, visualization with the Hu layout provides better separation of the different communities. In the visualization graphs, different colors are used to represent users in different communities; red dots represent the overlapping users that belong to multiple communities.

For the LiveJournal dataset, 4 communities are detected by our method as shown in Fig. 4(a). It can be observed in the FR layout that these communities have a small number of overlapping users, except for the communities shown in green and magenta. It can be seen in the Hu layout that the two communities can be linked through one hop of overlapping users (e.g., the communities in yellow and magenta). This is consistent with the characteristics of the LiveJournal dataset, as the interaction graph constructed from this dataset has a small diameter and average path length, as shown in Table 3. Overlapping users thus serve as hub nodes that reduce the distance between users in different communities. For the communities

Table 4
Statistical characteristics of the detected communities (Diam., Avg.Path.Len., and C.Coef are the short forms of diameter, average path length, and clustering coefficient, respectively).

Dataset	Set of nodes	Diam.	Avg.Path.Len.	C.Coef
LiveJ	All nodes	10	3.87	0.28
	Avg. of comm.	4	2.24	0.37
Weibo	All nodes	20	6.76	0.35
	Avg. of comm.	5.37	3.55	0.43
Tieba	All nodes	7	3.91	0.05
	Avg. of comm.	4.5	2.13	0.17

in green and magenta, we further analyzed the extracted dataset and found that the users in these two communities are fans of two celebrities. Thus, the overlapping users are most likely those who are interested in both celebrities.

For the Weibo dataset, 26 communities are detected by our method, as shown in Fig. 4(b). These communities have clear boundaries (see the FR layout), indicating fewer interactions between different communities, and the two communities are usually linked through multiple hops of overlapping users (see the Hu layout). The clear boundary phenomenon is probably due to the relative closeness of the social object sharings mechanism. In Sina Weibo, a user usually views social object sharings (i.e., micro-blogs) posted by users that they already follow, unless the user switches to other viewing tabs. Therefore, it is easier to have interactions with users who have connections. In addition, Sina Weibo supports retweet functionality, which makes indirect interactions more frequent than direct interactions, as shown in Table 2. This results in a large diameter and path length, as shown in Table 3, which explain the multiple-hop linkages between the detected communities.

For the Tieba dataset, 4 communities are detected by our method, as shown in Fig. 4(c). Compared to the results of the LiveJournal and Weibo datasets, there are more overlapping users and interactions between different communities in the Tieba dataset. This may be because Baidu Tieba is a more “open” OSN where users can view all social object sharings (i.e., posts) in a forum-like user interface and interact with other users based on their interests. In addition, there are many more direct interactions than indirect interactions, as shown in Table 2. This indicates the presence of many hub nodes that many other users directly interact with. Users interacting with these hub nodes may be from different communities. The distance between users from different communities is therefore short and communities are less separated from each other, as shown in both layouts in Fig. 4(c).

Table 4 shows the statistical characteristics of the detected communities from each dataset in terms of diameter, average path length and clustering coefficient. Specifically, we first calculate the values of these metrics for the entire interaction graph (i.e., all nodes) and then compare the average values of the same metrics across all of the communities in each dataset. As shown in Table 4, the detected communities generally have lower diameters and average path lengths but higher clustering coefficients than those derived from the whole graph. Low diameters and average path lengths within a detected community imply that users in that community can reach each other over a short path in the interaction graph, whereas high clustering coefficients indicate that users in a community are more densely connected compared to users outside the community. This generally reflects the intrinsic nature of communities in OSNs.

Fig. 5 provides a step-by-step example of the four steps of our ICDCA method using the Weibo dataset. For this dataset, step 1 of our ICDCA generates 237 event graphs, which are graph-based representations of the cascading events associated with 237 social object sharings (see Table 2). Fig. 5(a) shows one such event graph, where nodes represent users and an edge is established if two users interact. The numbers along the edges are the weights representing the strengths of ties based on our proposed weighting scheme (see Section 4.2). Note that the weights of some edges are not shown in Fig. 5(a) to allow clearer display of the event graph.

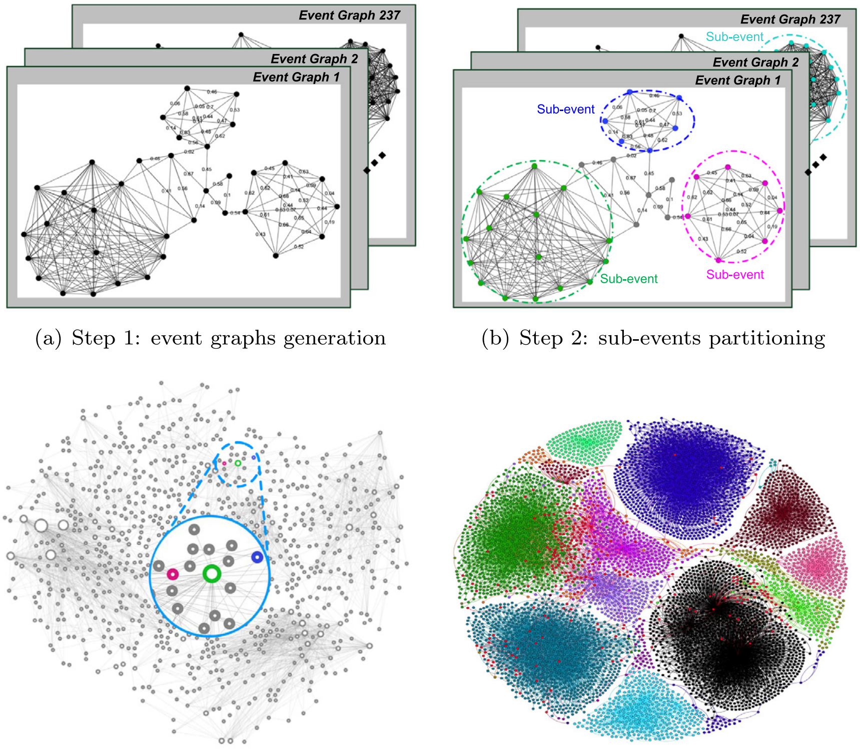
In step 2 of our ICDCA, each generated event graph is partitioned into a set of sub-events, each of which is a densely connected group of users as shown in Fig. 5(b). The users in a sub-event potentially belong to the same community. For all of the event graphs in the Weibo dataset, a total of 1166 sub-events are obtained, each containing an average of 8–10 users. Note that many users participate in multiple social object sharings and are thus involved in multiple sub-events.

In step 3 of our ICDCA, the sub-events obtained over all event graphs are treated as super nodes and a super graph is constructed where an edge between two super nodes is established based on the similarity measure described in Section 4.4. Fig. 5(c) provides a zoomed-in view of the super graph constructed for the Weibo dataset where the nodes in green, blue and magenta correspond to the three sub-events in Fig. 5(b). The size of a super node is proportional to the number of users in the corresponding sub-event.

In step 4 of our ICDCA, community detection is performed on the constructed super graph. This step essentially merges similar super nodes (i.e., sub-events) into a community, and produces 26 such communities for the Weibo dataset. Fig. 5(d) shows the final community detection results where each node represents a user and the colors of the nodes represent different communities. Red nodes indicate overlapping users belonging to multiple communities.

5.4.2. Evaluation based on MI-score

Based on the *MI-score* metric proposed in Section 5.3.1, the performance of our ICDCA method is compared with that of SLAP, NISE and iOSLOM. For the evaluation, β was set to 0.5, 1.0 and 1.5, which represent different strengths for the



(a) Step 1: event graphs generation (b) Step 2: sub-events partitioning

(c) Step 3: super graph construction (d) Step 4: community detection

Fig. 5. A step-by-step example of ICDCA using the Weibo dataset.

modularity and interaction degree. The corresponding results are shown in Fig. 6. It can be seen that our method achieves a higher *MI-score* in all cases. Furthermore, our method yields stable and consistent results, whereas some compared methods (i.e., SLPA) perform well for one dataset, but have a degraded performance for others.

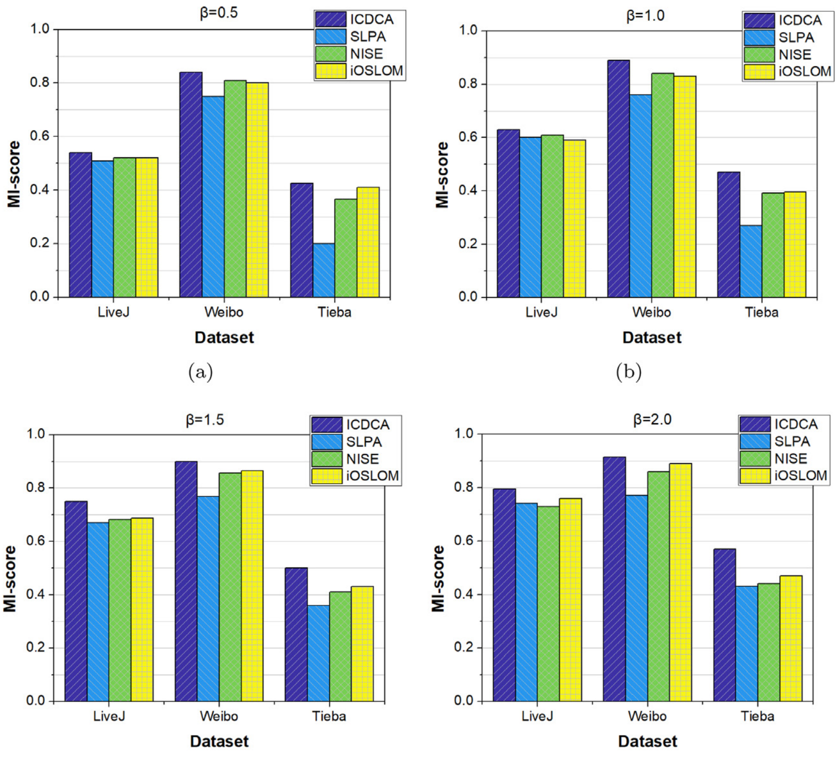
For the LiveJournal dataset, Fig. 6 shows that both ICDCA and the compared methods achieve comparable results in the cases of $\beta = 0.5$ and $\beta = 1.0$. ICDCA outperforms the other three methods when $\beta = 1.5$ and $\beta = 2.0$ (i.e., the interaction degree is emphasized more). On LiveJournal, users' interactions are largely influenced by the "community" functionality provided by the OSN. That is, a user tends to interact more with users who join the same community. Thus, the interaction behaviors of users naturally exhibit the internal community structure of the OSN. This is likely why all methods, including ICDCA, can achieve relatively good results across all β settings.

For the Weibo dataset, ICDCA has a very good performance in terms of *MI-score*, as shown in Fig. 6. One salient feature of the Weibo dataset is that it contains many more indirect interactions than direct interactions. This implies that most user interactions are performed on "retweets" of original social object sharing or as "comments on comments". Such a behavior pattern intrinsically requires certain cascading analysis, which is the core idea of ICDCA. It can also be observed that iOSLOM achieves a good performance for the Weibo dataset, especially for the cases of $\beta = 1.5$ and $\beta = 2.0$. One reason is that iOSLOM is also an interaction-based method and therefore tends to yield better results when the interaction degree is emphasized. Another reason may be that the Weibo dataset does not have too many overlapping users because of how users view social object sharings as explained previously. iOSLOM is based on a two-step method that first uses a non-overlapping community detection method (i.e., the Louvain method) to perform coarse-grained partitioning and then makes refinements in the second step. It may therefore handle such datasets well. NISE also archives good results for the Weibo dataset, especially for the cases of $\beta = 0.5$ and $\beta = 1.0$. NISE tends to yield a higher value in the modularity part compared to the interaction degree part of *MI-score*.

For the Tieba dataset, it is challenging to achieve a good *MI-score*. This is primarily due to the "openness" of sharing and viewing social object sharings in Baidu Tieba, which may lead to many overlapping users between different communities. Nevertheless, ICDCA still obtains a higher *MI-score* compared to the other three methods. For the cases of $\beta = 0.5$, the performances of ICDCA and iOSLOM's are comparable. For the other cases, ICDCA achieves much higher *MI-scores*. iOSLOM also achieves relatively high *MI-scores* compared to the other two methods.

5.4.3. Evaluation based on NMI and Omega index

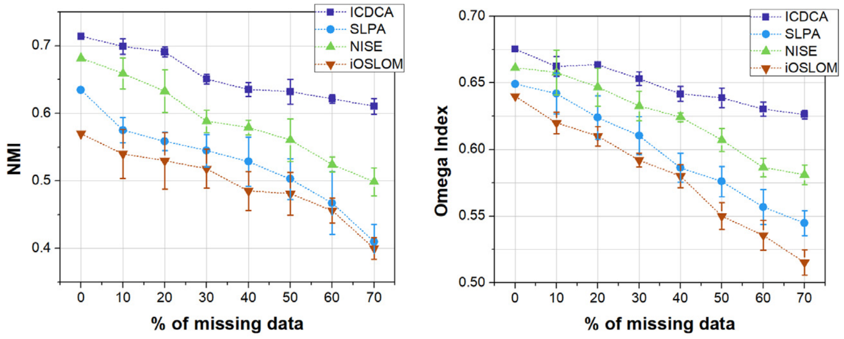
For the LiveJournal dataset in which the ground truth is known, the performances of ICDCA, SLPA, NISE and iOSLOM are evaluated based on the NMI and Omega index. To testify the robustness of these methods, evaluations with the absence



(a) (b)

(c) (d)

Fig. 6. MI-score evaluations of different methods for LiveJournal, Weibo and Tieba datasets.



(a) (b)

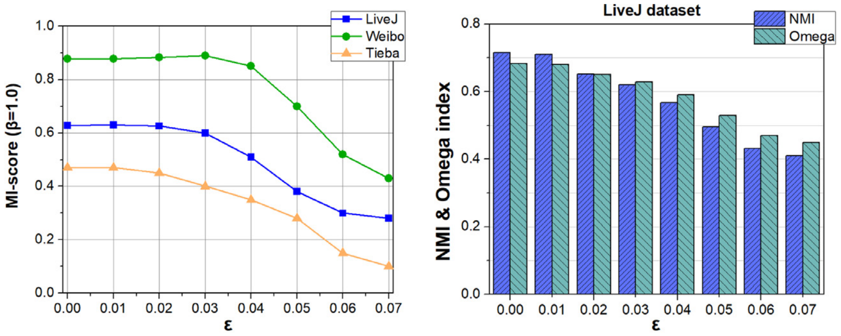
Fig. 7. NMI and Omega index evaluations under different removal rates of user interactions data.

of a certain amount of user interaction data are conducted. To do so, we removed 0%–70% of user interactions from the LiveJournal dataset. To avoid any bias in the data removal process, the removed user interactions are randomly chosen. If all of the interactions associated with a certain user are removed, this user is also excluded from the dataset and the ground truth data. For each removal rate r (e.g., $r = 20\%$), we repeated the data removal process 5 times, resulting in 5 different datasets for a given removal rate (except for $r = 0\%$). We then measure the average NMI and Omega index across the datasets for each removal rate. Fig. 7 shows the decreasing trend of average NMI and Omega index with an increasing removal rate for ICDCA and the compared methods. Table 5 shows the average NMI and Omega index values and corresponding standard deviations at four representative removal rates: 0%, 20%, 40% and 60%.

Fig. 7 shows that our ICDCA achieves the highest average NMI and Omega index values under all removal rates. For the tested methods, the NMI and Omega index generally decreases with an increasing removal rate. However, ICDCA drops more slowly compared to the other methods. From Table 5, it can be seen that ICDCA also yields smaller standard deviations for the NMI and Omega index values across multiple runs of the data removal process for each removal rate. NISE achieves the second best NMI and Omega index values. This is reasonable since NISE is known to be good at achieving high ground

Table 5
Average NMI and Omega index values and corresponding standard deviations (in brackets) at four representative removal rates.

Removal rate		$r = 0\%$	$r = 20\%$	$r = 40\%$	$r = 60\%$
ICDCA	NMI	0.71(0)	0.69(0.008)	0.64(0.010)	0.62(0.006)
	Omega	0.68(0)	0.66(0.002)	0.64(0.006)	0.63(0.005)
SLPA	NMI	0.63(0)	0.56(0.014)	0.53(0.036)	0.57(0.047)
	Omega	0.65(0)	0.62(0.016)	0.59(0.011)	0.56(0.013)
NISE	NMI	0.68(0)	0.63(0.032)	0.58(0.011)	0.52(0.011)
	Omega	0.66(0)	0.65(0.014)	0.62(0.003)	0.59(0.007)
iOSLOM	NMI	0.57(0)	0.53(0.041)	0.48(0.029)	0.46(0.019)
	Omega	0.64(0)	0.61(0.007)	0.58(0.009)	0.54(0.010)



(a) (b)

Fig. 8. Impact of parameter ϵ.

truth accuracy, as in [38]. Unlike in the evaluation of *MI-score*, iOSLOM does not perform very well in the evaluation of NMI and Omega index. This is likely because iOSLOM tends to generate some orphan nodes as singleton communities [8], which causes the number of detected communities to deviate from the number of the ground truth communities.

The reasons why our ICDCA can yield robust and stable results with the absence of data are two-fold. First, our method does not only analyze the explicit relationship between two users in terms of their interactions, it also explores the implicit relationship that two users' interactions are associated with the same social object sharing. Such implicit relationships are retained unless all of the interactions of a user associated with a social object sharing are removed. Second, in the event graphs generation step of our method, the group behavior weight is used to complement the direct interaction weight, which helps to explore the implicit relationship that two users both interact with a common user. Such analysis of group behavior is less affected by the removal of data.

5.4.4. Impact of the parameter ϵ

The parameter ϵ is a threshold value that affects the number of edges established in the super graph in step 3 of our method. A smaller ϵ value indicates that more edges are established in the super graph. The community detection results based on the super graph are therefore affected by the ϵ value. In this section, we investigate how different values of ϵ affect the performances of our method. To this end, the value of ϵ is varied from 0 to 0.07 in steps of 0.01, and our ICDCA is run with each value. For all three datasets, the corresponding *MI-score*s are obtained. For the LiveJournal dataset, the corresponding NMI and Omega index values are also obtained. The results are shown in Fig. 8.

As shown in Fig. 8(a), the *MI-score* generally decreases with an increasing ϵ value. Compared to the other two datasets, the Weibo dataset maintains a good *MI-score* when the ϵ value is set to 0.04. This may be because the Weibo dataset contains more indirect interactions (see Table 2), which are likely performed by users within the same community. Since our method can more easily extract these users as sub-events (i.e., nodes in our super graph), the results are less susceptible to an increased ϵ value. In principle, ϵ can be set to a relatively large value if the dataset for a given OSN contains more indirect interactions than direct ones. From Fig. 8(b), it can be observed that the NMI and Omega index also drop with an increasing ϵ value. It can also be seen that NMI drops more quickly than the Omega index. This is because, with the increasing ϵ value, the number of detected communities also increases. Evaluation of NMI especially the calculation of conditional entropy $H(X|Y)$ is more easily affected by the number of communities than evaluation of the Omega index.

6. Conclusions

In this paper, we advocate the use of social object sharings and associated user interactions to analyze the potential community structure within a given OSN. To this end, a user interaction-oriented community detection method based on cascading analysis is proposed. The method comprises four key steps. The first step is to use a graph-based representation (i.e., event graph) to capture the cascading relations among user interactions associated with social object sharings and generate an event graph for each social object sharing. The second step is to cluster groups of actively interacting users in each event graph as sub-events. The third and four steps adopt a super graph approach that treats all extracted sub-events as super nodes to construct a super graph on which community detection is performed. Extensive evaluation of the proposed ICDCA using three real OSN datasets was conducted, and the results were compared with those of general and interaction-based community detection methods. In this comparison, our ICDCA achieved the best results based on several evaluation metrics. Our ICDCA produced more robust and stable detection results across the different datasets.

This work is expected to call greater attention to the use of user interaction data to detect and analyze community structure in OSNs. There are several future directions. One promising direction is how to extend the proposed method to deal with dynamic social networks where the community structure evolves over time. It would also be interesting to perform an in-depth analysis of the crowd behaviors of the detected communities and compare the differences in behaviors with those of communities detected by connection-based methods. Lastly, content-based community detection methods (e.g., the description-oriented method in [4]) could be integrated with the proposed method to enhance the interpretability of the detected communities.

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.

Acknowledgment

This work is supported by National Natural Science Foundation of China (Grant No. 61872282), Natural Science Basic Research Plan in Shaanxi Province of China (Program No. 2019JM-031), Key R&D Program of Shaanxi Province of China (Grant No. 2019ZDLGY12-06) and the Open Project Program of State Key Laboratory of Virtual Reality Technology and Systems, Beihang University (No. VRLAB2019C04).

References

[1] MinHash LSH implementation, (https://ekzhu.github.io/datasketch/).
[2] Y.-Y. Ahn, J.P. Bagrow, S. Lehmann, Link communities reveal multiscale complexity in networks, Nature 466 (2009) 761–764.
[3] A. Gionis, P. Indyk, R. Motwani, Similarity search in high dimensions via hashing, in: Proceedings of the 25th International Conference on Very Large Data Bases, 1999, pp. 518–529.
[4] M. Atzmueller, S. Doerfel, F. Mitzlaff, Description-oriented community detection using exhaustive subgroup discovery, Inf. Sci. 329 (2016) 965–984.
[5] M. Bastian, S. Heymann, M. Jacomy, Gephi: an open source software for exploring and manipulating networks, in: Proceedings of the International AAAI Conference on Weblogs and Social Media, 2009, pp. 361–362.
[6] V.D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech. Theory Exp. 2008 (10) (2008) P10008:1–P10008:12.
[7] D. Chen, Y. Dong, X. Huang, H. Chen, D. Wang, A community finding method for weighted dynamic online social network based on user behavior, Int. J. Distrib. Sens. Netw. 11 (6) (2015) 306160:1–306160:10.
[8] D. Darmon, E. Omodei, J. Garland, Followers are not enough: a multifaceted approach to community detection in online social networks, PloS One 10 (8) (2015) e0134860:1–e0134860:20.
[9] H. Dev, M.E. Ali, T. Hashem, User interaction based community detection in online social networks, in: Proceedings of the 19th International Conference of Database Systems for Advanced Applications, 2014, pp. 296–310.
[10] T. Evans, R. Lambiotte, Line graphs, link partitions, and overlapping communities, Phys. Rev. E 80 (1) (2009) 016105:1–016105:8.
[11] I. Farkas, D. Ábel, G. Palla, T. Vicsek, Weighted network modules, New J. Phys. 9 (6) (2007) 180:1–180:18.
[12] S. Fortunato, Community detection in graphs, Phys. Rep. 486 (3) (2010) 75–174.
[13] S. Fortunato, D. Hric, Community detection in networks: a user guide, Phys. Rep. 659 (2016) 1–44.
[14] T.M. Fruchterman, E.M. Reingold, Graph drawing by force-directed placement, Softw. Pract. Exp. 21 (11) (1991) 1129–1164.
[15] R. Fuentes-Fernndez, J.J. Gmez-Sanz, J. Pavn, User-oriented analysis of interactions in online social networks, IEEE Intell. Syst. 27 (4) (2012) 18–25.
[16] M. Girvan, M.E. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. 99 (12) (2002) 7821–7826.
[17] S. Gregory, Fuzzy overlapping communities in networks, J. Stat. Mech. Theory Exp. 2011 (02) (2011) P02017.
[18] K. Guo, W. Guo, Y. Chen, Q. Qiu, Q. Zhang, Community discovery by propagating local and global information based on the mapreduce model, Inf. Sci. 323 (2015) 73–93.
[19] Y. Hu, Efficient, high-quality force-directed graph drawing, Math. J. 10 (1) (2005) 37–71.
[20] L. Hubert, P. Arabie, Comparing partitions, J. Classif. 2 (1) (1985) 193–218.
[21] J. Jiang, C. Wilson, X. Wang, W. Sha, P. Huang, Y. Dai, B.Y. Zhao, Understanding latent interactions in online social networks, ACM Trans. Web 7 (4) (2013) 18:1–18:39.
[22] L. Jin, Y. Chen, T. Wang, P. Hui, A.V. Vasilakos, Understanding user behavior in online social networks: a survey, IEEE Commun. Mag. 51 (9) (2013) 144–150.
[23] S. Kelley, M. Goldberg, M. Magdon-Ismail, K. Mertsalov, A. Wallace, Defining and discovering communities in social networks, in: Handbook of Optimization in Complex Networks, Springer, 2012, pp. 139–168.
[24] A. Lancichinetti, S. Fortunato, J. Kertész, Detecting the overlapping and hierarchical community structure in complex networks, New J. Phys. 11 (3) (2009) 033015.
[25] A. Lancichinetti, F. Radicchi, J.J. Ramasco, S. Fortunato, Finding statistically significant communities in networks, PloS One 6 (4) (2011) e18961:1–e18961:18.
[26] K.H. Lim, A. Datta, An interaction-based approach to detecting highly interactive twitter communities using tweeting links, Web Intell. 14 (1) (2016) 1–15.
[27] X. Liu, W. Wang, D. He, P. Jiao, D. Jin, C.V. Cannistraci, Semi-supervised community detection based on non-negative matrix factorization with node popularity, Inf. Sci. 381 (2017) 304–321.

[28] S. Milgram, The Familiar Stranger: An Aspect of Urban Anonymity, Addison-Wesley, 1977.
[29] M.E. Newman, Analysis of weighted networks, Phys. Rev. E 70 (5) (2004) 056131:1–056131:9.
[30] M.E. Newman, Fast algorithm for detecting community structure in networks, Phys. Rev. E 69 (6) (2004) 066133:1–066133:5.
[31] M.E. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2) (2004) 026113:1–026113:15.
[32] G. Palla, I. Derényi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, Nature 435 (2005) 814–818.
[33] M.J. Rattigan, M. Maier, D. Jensen, Graph clustering with network structure indices, in: Proceedings of the 24th International Conference on Machine Learning, 2007, pp. 783–790.
[34] M. Sachan, D. Contractor, T.A. Faruquie, L.V. Subramaniam, Using content and interactions for discovering communities in social networks, in: Proceedings of the 21st International Conference on World Wide Web, 2012, pp. 331–340.
[35] P. Schuetz, A. Caflisch, Efficient modularity optimization by multistep greedy algorithm and vertex mover refinement, Phys. Rev. E 77 (4) (2008) 046112:1–046112:7.
[36] H. Shen, X. Cheng, K. Cai, M.-B. Hu, Detect overlapping and hierarchical community structure in networks, Phys. A Stat. Mech. Appl. 388 (8) (2009) 1706–1712.
[37] X. Wen, W.-N. Chen, Y. Lin, T. Gu, H. Zhang, Y. Li, Y. Yin, J. Zhang, A maximal clique based multiobjective evolutionary algorithm for overlapping community detection, IEEE Trans. Evol. Comput. 21 (3) (2017) 363–377.
[38] J.J. Whang, D.F. Gleich, I.S. Dhillon, Overlapping community detection using neighborhood-inflated seed expansion, IEEE Trans. Knowl. Data Eng. 28 (5) (2016) 1272–1284.
[39] S. White, P. Smyth, A spectral clustering approach to finding communities in graphs, in: Proceedings of the SIAM International Conference on Data Mining, 2005, pp. 274–285.
[40] C. Wilson, A. Sala, K.P. Puttaswamy, B.Y. Zhao, Beyond social graphs: user interactions in online social networks and their implications, ACM Trans. Web 6 (4) (2012) 17:1–17:31.
[41] W. Wu, S. Kwong, Y. Zhou, Y. Jia, W. Gao, Nonnegative matrix factorization with mixed hypergraph regularization for community detection, Inf. Sci. 435 (2018) 263–281.
[42] Z.-H. Wu, Y.-F. Lin, S. Gregory, H.-Y. Wan, S.-F. Tian, Balanced multi-label propagation for overlapping community detection in social networks, J. Comput. Sci. Technol. 27 (3) (2012) 468–479.
[43] J. Xie, S. Kelley, B.K. Szymanski, Overlapping community detection in networks: The state-of-the-art and comparative study, ACM Comput. Surv. 45 (4) (2013) 43:1–43:35.
[44] J. Xie, B.K. Szymanski, Community detection using a neighborhood strength driven label propagation algorithm, in: Proceedings of the IEEE Network Science Workshop, 2011, pp. 188–195.
[45] J. Xie, B.K. Szymanski, Towards linear time overlapping community detection in social networks, in: Proceedings of the 16th Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2012, pp. 25–36.