A Novel Trust Model Based Overlapping Community Detection Algorithm for Social Networks

Shuai Ding, *Member, IEEE*, Zijie Yue, Shanlin Yang, Feng Niu, and Youtao Zhang, *Member, IEEE*

Abstract—With the fast advances in Internet technologies, social networks have become a major platform for social interaction, lifestyle demonstration, and message dissemination. Effective community detection in social networks helps to assess public sentiment, identify community leaders, and produce personalized recommendation. While different community detection approaches have been proposed in the literature, the trust model based detection schemes model user interactions as trust transfer, which helps to capture the implicit relation in the network. Unfortunately, trust model based detection schemes face a *cold start* problem, i.e., they cannot accurately model newly joined users as these users have few interactions for a duration after joining the network. In this paper, we propose TLCDA, a novel trust model based community detection algorithm. By enhancing the traditional trust computation with inter-node relation strength and similarity in social networks, TLCDA detects communities through coarse-grained K-Mediods clustering. Our evaluation on real social networks shows that the communities detected by TLCDA exhibit superior preference cohesion while satisfying the topology cohesion.

Index Terms—Community detection, social network, trust, coarse cluster

1 Introduction

With the fast advances in Internet technologies, social networks have become a major platform for social interaction, lifestyle demonstration, and message dissemination. Effective community detection reveals not only the behaviors of individuals but also their relationship in the network. By exposing the hidden structures in the network, community detection enables the analysis of their functionalities as well as the interactions with individuals, which support better assessment of public sentiment, identification of community leaders [1]. In addition, effective community detection helps to deliver advertisements precisely to potential clients, construct reliable E-business recommendation systems, and customize the results from searching engines [2].

The traditional community detection algorithms can be divided into three types.

- Node-link-based algorithms [3], [4], [5], [6], [7], [8], [9], [10]. By extracting nodes and links in the network, such algorithms convert community detection problem to graph problem. These algorithms ignore the contents and other characteristics of the nodes in the network.
- Node-content-based algorithms [11], [12], [13], [14], [15]. By studying the node contents as well as their other characteristics, such algorithms convert community detection problem to node clustering problem. The network topology is often ignored in these algorithms.
- Hybrid algorithms [16], [17]. These algorithms detect communities in a network twice based on the network topology and on the node contents, respectively. The final communities, merged from the ones detected in two steps, are both topology and content cohesive. Given these algorithms execute the detection twice, they often exhibit low performance for large networks.

Intuitively, the explicit connections among the users in a social network are due to the common or similar social activities that these users attend while the implicit connections are due to overlapped neighbors as well as similar shopping records, browsing history, and trusted users. In a social network that has large number of similar services, users are more likely to accept the suggestions and recommendations from their trusted users. To characterize the user relationship in social networks, researchers have started to adopt the trust model to weigh the importance of individual users and the strength of the connections. Evaluating trust transfer helps to model social connections and to describe their semantics, which can greatly improve the analysis accuracy of social networks [18], [19].

However, trust model based detection schemes, like other detection algorithms, face the *cold start* problem when

- *S. Ding, Z. Yue, S. Yang, and F. Niu are with the School of Management, and Key Laboratory of Process Optimization and Intelligent Decision-Making, Ministry of Education, Hefei University of Technology, Anhui, Hefei 23009, China.*
  E-mail: {dingshuai, yangsl, niufeng}@hfut.edu.cn, q164910798@gmail.com.
- *Y. Zhang is with the Computer Science Department, University of Pittsburgh, Pittsburgh, PA 15260. E-mail: zhangyt@cs.pitt.edu.*

Manuscript received 15 Jan. 2018; revised 5 Oct. 2018; accepted 22 Apr. 2019.
Date of publication 30 Apr. 2019; date of current version 6 Oct. 2020.
(Corresponding author: Youtao Zhang.)
Recommended for acceptance by X. Lin.
Digital Object Identifier no. 10.1109/TKDE.2019.2914201

modelling the trust in social networks, i.e., a new user often has few interactions with existing users for a duration after joining a social network. To solve this problem, Deng et al. [20] proposed a personalized recommendation algorithm that exploits diffusion process to integrate the networks of friends and user-product relations. Barjasteh et al. [21] proposed a matrix completion based approach to simultaneously exploit the similarity information among users and items to alleviate the cold-start problem. In existing trust-based models, the relation-strength and similarity between users are not exploited as the basis for model construction, leading to less accurate trust estimation with few dynamic interactions.

In this paper, we propose TLCDA, a novel trust model based community detection algorithm for detecting overlapping communities in social networks. TLVDA enhances traditional trust model with inter-node relation strength and similarity and then detects communities through coarse-grained K-Mediods clustering.

We summarize our main contributions as follows.

- We design TLCDA, a trust-based local overlapping community detection algorithm, that abstracts social network as data field, exploits trust potential to evaluate the local impact among nodes (i.e., users in the social networks), and then adopts the coarse-grained K-Medoids clustering to detect overlapped communities.
- We propose a trust evaluation model based on node similarity as well as their connections. We integrate the two types of trust and adjust the weights dynamically to mitigate the cold-start problem.
- We implement TLCDA with data extracted from real websites and compare it with the state-of-the-art community detection algorithms. Our results show that the communities detected by TLCDA exhibit superior preference cohesion while satisfying the topology cohesion.

In the rest of the paper, we briefly discuss the related work in Section 2. We present the trust model in Section 3 and elaborate the TLCDA algorithm in Section 4. We analyze the experiment results in Section 5 and conclude the paper in Section 6.

2 Related Work

2.1 Community Detection

Community detection for social networks has been extensively investigated in recent years. The detection algorithms can be divided into three types: node-link (network topology)-based community detection algorithms, node-content-based community detection algorithms and hybrid community detection algorithms.

The node-link-based community detection algorithms are the most widely adopted algorithms. In 1970, based on graph cut theory, Kernighan and Lin proposed the Kernighan-Lin (KL) algorithm [3], which identifies the community divisions through iterative decomposition of network sub-graphs for optimal gain function. In 2002, Girvan and Newman proposed the betweenness-based Girvan-Newman (GN) algorithm [4], which identifies the optimal network community division via iterative deletion of links with maximum betweenness. In 2004, Newman and Girvan proposed the concept of modularity [5] and designed the fast-Newman algorithm. In 2005, Guimera et al. proposed a simulation-degradation-based genetic algorithm (GA) algorithm [7], which exploits GA algorithm for identifying a global optimal solution. Subsequently, many researchers optimized the modularity function and proposed different algorithms for weighted network community detection, directed network community detection and overlapping community detection [22], [23], [24]. In general, node-link-based community detection algorithms do not consider node attributes and other features in a social network.

Node-content-based community detection algorithms extract node attributes, compute attribute similarity and discover network community based on node clustering. In 1999, Kleinberg et al. proposed a content-similarity-based webpage clustering algorithm named Hyperlink-Induced Topic Search (HITS) [11]. In 2004, based on latent Dirichlet allocation (LDA) research, Syeyvers et al. proposed an author-topic (AT) model [13] to identify relations among users, documents, subjects and keywords in a network. In 2007, based on the send-receive relation, McCallum et al. proposed the author-recipient-topic (ART) model to identify users with similar interests [14]. In 2010, Fang et al. [25] developed an active-learning privacy wizard that constructs a machine learning classifier to build privacy-preference models. In 2016, Misra et al. [26] proposed to analyze profile attributes and found that it is sufficient to calculate user similarity with a pair of profile attributes. The user similarity calculated by this method can also be utilized in the community detection area. Although node-content-based community detection algorithms consider network node attributes, they do not exploit important link topology information.

To address the deficiencies of the preceding two types of algorithms, hybrid community detection algorithms were proposed to take both nodes/links and content information into consideration. In 2010, Yan et al. [16] obtained a network community with similar interest patterns via content-based clustering and employed link information to expand obtained community topology. In 2012, Zhang et al. [17] performed link-based network community division and content-based similarity clustering via a non-negative matrix factorization (NMF) method and AT model to obtain comprehensive network community division. Leskovec et al. [27] defined a novel machine learning task for identifying social circles, which used not only the set of edges of social networks but also properties or traits of nodes to discover users' communities. These type of algorithms maintain network topology and detect network community with potential interest patterns. Since these algorithms execute the detection twice, they exhibit low performance for large networks.

2.2 Trust Modeling

Social networks contain explicit relations among users and implicit relations that are manifested in similar requirements for cloud services, similar spending or browsing records, shared neighbor networks, and similar trust preferences. In 2009, Gilbert et al. [28] presented a predictive model that maps social media data to tie strength and then computes the relation-strength using a linear combination of the predictive variables and terms for dimension interactions and network structure. In 2014, Fogus et al. [29] presented a BFF tool that automates the elicitation of tie strength and user communities to a large extent. In general, trust is a subjective feeling that is shared by users in a specific context and has been used to indicate the strength of user relationship. To analyze the patterns of user social relations in a social network, scholars measured user relation attributes such as individual weight, relation strength and probability distribution via trust and construct social network relation model and semantic description via trust transfer and trust evolution, which are practically useful for improving the accuracy in social network analysis.

The trust in social networks can be modeled using either a static trust model or a dynamic one. Golbeck et al. [30] summarized the relation between trust and user similarity in a social network, which laid the foundation for studying service recommendation in social networks. Bhuiyan et al. [31] showed that a higher level of preference similarity between two nodes in a social network indicates a higher level of inter-node trust, which can be exploited to calculate direct trust between two adjacent nodes and recommendation trust between two non-adjacent nodes. By leveraging the inter-node trust model in peer-to-peer (P2P) networks, researchers proposed feedback-based dynamic trust computational models to study the dynamics in social networks, e.g., the EigenTrust model [24] and the PeerTrust model [32].

Dynamic trust models, while performing better than static ones for stable networks, face the "cold start" problem when modelling the trust in social networks. To address this problem, Deng et al. [20] proposed a personalized recommendation algorithm that exploits diffusion process to integrate the networks of friends and user-product relations. Barjasteh et al. [21] proposed a matrix completion based approach to simultaneously exploit the similarity information among users and items to alleviate the cold-start problem. Chen et al. [33] proposed a cold start recommendation method that integrates a user model with trust and distrust networks to identify trustworthy users and then provide useful recommendations. Guo et al. [34] proposed a novel trust-based matrix factorization model TrustSVD for recommendation. Facing the "cold start", it considers both the explicit and the implicit influence of ratings and trust information when predicting ratings of unknown items. Deng et al. [35] showed a novel MF method for trust-aware recommendation by employing a deep learning technique, which exhibits good performance in solving the cold-start problem. In existing trust-based models, the relation-strength and similarity between users are not exploited as the basis for model construction, leading to less accurate trust estimation with few dynamic interactions.

3 Trust Relation Modeling

3.1 Hybrid Trust Model

In this section, we propose a hybrid trust model to integrate static and dynamic trust computation. The overall trust model is based on the node similarity and relation strength with different weights. In particular, we assign more weights on inter-node similarity for newly added nodes, and adjust the weights as node interactions increase such that the feedback-based dynamics help to estimate the trust more accurately, which helps to address the code-start problem.

Assume that u and v are two nodes in a social network G, the trust degree by node u for node v is calculated as

$$Trust(u,v) = \beta RTrust(u,v) + (1-\beta) STrust(u,v), \tag{1}$$

where $RTrust(u,v)$ and $STrust(u,v)$ represent relation-strength-based trust and similarity-based trust, respectively, and β is a weight coefficient.

3.2 The Relation-Strength-Based Trust

According to sociology and psychology studies, the trust is often reflected by subject behavior or action [36]. Thus, the trust relation in a social network is typically represented as social behaviors, such as active user communication, message forwarding or commenting.

Assume that W is the adjacent matrix of a social network G, $W_{u,v}$ = 1 and $W_{u,v}$ = 0 indicate whether nodes u and v are adjacent or not, respectively. The trust relation is defined as

$$RTrust(u,v) = \begin{cases} D\_RTrust(u,v) & \text{if } W_{u,v} = 1 \\ I\_RTrust(u,v) & \text{if } W_{u,v} = 0, \end{cases} \tag{2}$$

where *D_RTrust(u, v)* and *I_RTrust(u, v)* are the direct trust and indirect trust, respectively, of node u versus node v .

Definition 1 (Direct Trust). *Given two adjacent nodes u and v in a social network G, their direct trust can be expressed as follows.*

$$D\_RTrust(u,v) = \frac{w(u,v)}{w(u)}, \tag{3}$$

where w(u, v) indicates the relation strength between u and v; and w(u) is the sum of all relation strengths between node u and its adjacent nodes. Clearly, we have D_RTrust(u, v) ∈ (0, 1].

Definition 2 (Indirect Trust). *Given two non-adjacent nodes u_1 and u_n in a social network G, and p = (u_1, u_2, . . ., u_n) is one of the shortest inter-node paths, the indirect trust for the path can be expressed as follows.*

$$I\_RTrust_p(u,v) = \begin{cases} \prod_{i=1}^{n-1} D\_RTrust(u_i, u_{i+1}) & \text{if } d_p \le d_{max} \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where d_p and d_{max} represent the length of the shortest inter-node path and the maximum trust transfer distance in the network, respectively.

If there are more than one shortest paths p_1, . . ., p_t between nodes u_1 and u_n, the indirect trust can be expressed as follows.

$$I\_RTrust(u,v) = MAX_{i=1}^{t}(I\_RTrust_{p_i}(u,v)). \tag{5}$$

Non-adjacent nodes in social networks typically have indirect connections via intermediate nodes. While indirect trust may be propagated via intermediate nodes, existing studies showed that propagating information along a long path leads to degraded accuracy and integrity [37]. That is, the indirect trust propagation in a social network often results in inferior quality and severe loss.

3.3 The Node-Similarity-Based Trust

Recent studies showed that nodes in social networks often have clear homogeneity, i.e., similar nodes tend to be connected [36], [38]. While existing node similarity measurements for social networks could include many dimensions, social information and attribute-based similarities have proven effectiveness in application areas such as social network node relation analysis, community detection and customized recommendation. In this paper, we choose social similarity and attribute similarity to calculate the node similarity-based trust.

$$STrust(u,v) = \alpha STrust\_S(u,v) + (1-\alpha) STrust\_A(u,v), \tag{6}$$

where *STrust_S(u, v)* and *STrust_A(u, v)* represent the social-similarity-based trust and the attribute-similarity-based trust between node u and node v, respectively; and α is the weight.

The inter-node social similarity measures the similarity among local network topologies for different nodes. Given two nodes in the network, they tend to have a higher level of node similarity if they have more overlapped neighbors [37]. In this paper, we compute social similarity based on overlapped neighbors.

Definition 3 (Social-Similarity-based Trust). *Assume that N(u) and N(v) are the adjacent node sets of nodes u and v, respectively. The social similarity-based trust is defined as follows.*

$$STrust\_S(u,v) = \frac{\sum_{t \in N(u)N(v)} \frac{1}{D(t)}}{\sqrt{\sum_{t \in N(u)} \frac{1}{D(t)}} \sqrt{\sum_{t \in N(v)} \frac{1}{D(t)}}}, \tag{7}$$

where D(t) represents the degree of node t.

In addition social similarity, attributes such as age, gender, place of residence and tag play a critical role in characterizing the nodes in a social network. We therefore integrate these attributes in computing the node similarity.

Definition 4 (Attribute-Similarity-Based Trust). *Assume that u and v are two nodes in a social network, {$\alpha_m, m = 1, \ldots, M$} are the attributes of a node; and $S_m(u,v)$ is the similarity of attribute α_m. We define the attribute-similarity-based trust as follows.*

$$STrust\_A(u,v) = \frac{1}{|M|} \sum_{m=1}^{M} S_m(u,v), \tag{8}$$

where $|M|$ represents the number of attributes.

Based on the type of information, social network node attributes are classified into discrete attributes and text attributes. If α_m is a discrete attribute, when nodes u and v have the same attribute values, the attribute similarity $S_m(u,v) = 1$; otherwise, $S_m(u,v) = 0$. In a text analysis, the keyword frequency has a direct impact on information discrimination. If α_m is a text attribute, its similarity calculation should consider the keyword difference. Assume that {k =, k = 1, . . . ,K} is the keyword for attribute α_m, and $D_{in}(k)$ is the number of nodes whose attributes contain the keyword k. Then, $S_m(u,v) = \sum_{k=1}^{T} I_u(k) \times I_v(k) \times \frac{1}{lg\ D_{in}(k)}$. If the attribute of node u contains the keyword k, then $I_u(k) = 1$; otherwise, $I_u(k) = 0$.

4 TCLDA: A Novel Trust Based Community Detection Algorithm

In this section, we discuss our community detection algorithm TLCDA. We first present an overview and then elaborate the details and analyze the complexity.

4.1 Algorithm Overview

According to data field theory, every data point in data space is surrounded by an interaction field. An interaction field potential function describes how the data point state is affected by other nodes in the data space [39], [40]. Given that social networks are microcosms with no dimensions, they often exhibit strong localized characteristics. In this paper, we abstract a social network as a short-range data field in which each data point represents a user while trust manifests as the interaction with different nodes.

The trust-based local overlapping community detection algorithm (TLCDA) is proposed based on data field and coarse clustering theories. It exploits trust potential to describe the node interactions in local ranges, and detects overlapping communities via coarse K-medoids clustering. As shown in Fig. 1, the TLCDA algorithm consists of the following steps.

Step 1: We compute the inter-node trust and then the trust potential of each node in the network.

Step 2: We identify the network nodes that have high trust potentials. They are determined as the initial clustering centers.

Step 3: We then classify the nodes in the network based on their trust potentials and place the nodes into clustering upper approximation and the clustering lower approximation sets. TCLDA reselects the clustering center after computing clustering upper and lower approximation sets, and repeats classification until the clustering centers stabilize, which terminates the coarse K-medoids clustering.

Step 4: We repeatedly merge clusters that have the most significant node overlapping.

4.2 Trust Parameters

Before we elaborate the algorithm details, we first define a number of trust based parameters that are to be used in the algorithm.

4.2.1 Trust Potential

Given many characteristics exhibit strong locality in social networks and the inter-node trust decreases as the distance increases, we adopt the Gaussian potential-function to compute the trust potential, which can objectively measure the inter-node trust influence and evolution pattern in social networks.

Definition 5 (Trust Potential). *Given a network $G(V,E)$, node $v_i \in E$ is randomly selected as a field source. We use $U(v_i) = \{v_1, v_2, \ldots, v_n\}$ to denote the interaction field centered around node v_i. The trust potential of node v_i at node v_j is defined as follows.*

$$p(v_i, v_j) = m_{v_j} \times exp\left(\frac{Trust(v_j, v_i)^2}{2\sigma^2}\right), \tag{9}$$

where m_{v_j} represents the intrinsic attributes (characteristics, and activity) of node v_j, and $Trust(v_j, v_i)$ represents the trust degree by node v_j for the node v_i; the node interaction field range is controlled via the parameter σ. Studies have shown that the influence range of a node in Gaussian potential function is approximately $3\sigma/\sqrt{2}$ hops [41], where σ is determined by the network details.

The trust potential for node v_i is expressed as follows.

$$p(v_i) = \sum_{v_j \in U(v_i)} exp\left(\frac{Trust(v_j, v_i)^2}{2\sigma^2}\right). \tag{10}$$

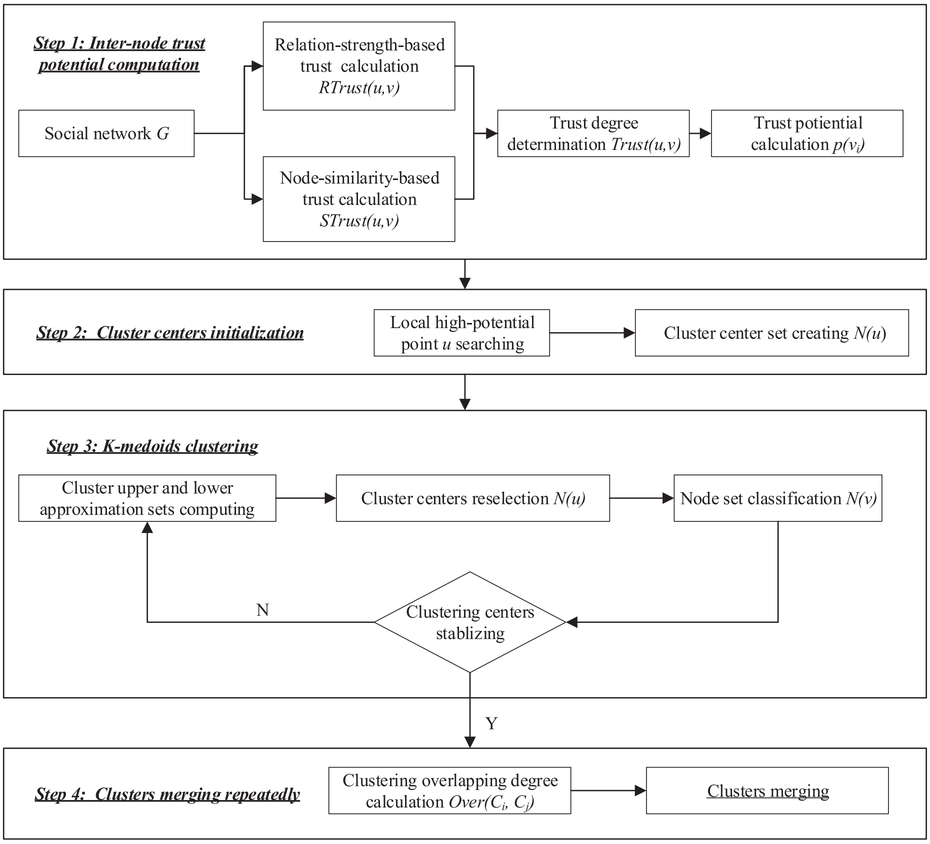

Step 1: Inter-node trust potential computation
Social network G
Relation-strength-based trust calculation RTrust(u,v)
Node-similarity-based trust calculation STrust(u,v)
Trust degree determination Trust(u,v)
Trust potiential calculation p(v_i)
Step 2: Cluster centers initialization
Local high-potential point u searching
Cluster center set creating N(u)
Step 3: K-medoids clustering
Cluster upper and lower approximation sets computing
Cluster centers reselection N(u)
Node set classification N(v)
Clustering centers stablizing
N
Y
Step 4: Clusters merging repeatedly
Clustering overlapping degree calculation Over(C_i, C_j)
Clusters merging

Fig. 1. The overview of proposed TLCDA.

4.2.2 Local High-Potential Node

Trust potential can be explored for detecting communities. For this purpose, we need to identify *local high-potential node* as follows. While a naive clustering algorithm could form a community from each high-potential node and its neighbors, the results are often sub-optimal. In this paper, we assign local high-potential nodes as initial clustering centers, and then adopt the hill climbing strategy to create an initial clustering center set.

Definition 6 (Local High-Potential Node). *Given a network $G(V, E)$, we denote the adjacent nodes of node v as $N(v) = \{u_1, u_2, \ldots, u_n\}$. Node v is a local high-potential point if it satisfies $p(v) \leq max\{p(v, u_1), p(v, u_2), \ldots, p(v, u_n)\}$.*

4.2.3 Trust Cohesion

Given a cluster that consists of a number of nodes, we compute trust cohesion as follows. In our algorithm, we use *trust cohesion* to determine if a cluster is of good structure.

Definition 7 (Trust Cohesion). *Given a clustering C_i and its center node $u_i \in C_i$, the trust cohesion of C_i is defined as*

$$CT(C_i, u_i) = \begin{cases} w_{low}C_{low} + w_{up}C_{up} & \text{if } \underline{C_i}, \overline{C_i} - \underline{C_i} \neq \phi \\ C_{low} & \text{if } \underline{C_i} \neq \phi, \overline{C_i} - \underline{C_i} = \phi \\ C_{up} & \text{if } \underline{C_i} = \phi, \overline{C_i} - \underline{C_i} \neq \phi \end{cases}$$
$$C_{low} = \sum_{v_i \in \underline{C_i}} p(u_i, v_i)$$
$$C_{up} = \sum_{v_i \in \overline{C_i} - \underline{C_i}} p(u_i, v_i), \tag{11}$$

where w_{low} and w_{up} represent the weights for the lower approximation set and the upper approximation set, respectively, of clustering C_i and $w_{low} + w_{up} = 1$; and $p(u_i, v_i)$ is the trust potential of center node u_i on node v_i. $\underline{C_i}$ and $\overline{C_i}$ denote the lower appropriation set and upper appropriation set of cluster C_i, respectively. We will elaborate the computation of these two sets when discussing the clustering algorithm.

Based on trust cohesion, we may compute a more appropriate cluster center as follows.

$$u_i = \{u | u \in C_i \wedge CT(C_i, u) = \max_{x \in C_i}\{CT(C_i, x)\}\}. \tag{12}$$

4.2.4 Overlapping Clusters

To enable the detection of overlapping communities, we define cluster overlapping as

Definition 8 (Clustering Overlapping). *Given two clusters C_i and C_j, their clustering overlapping degree is defined as follows.*

$$Over(C_i, C_j) = \frac{|C_i \cap C_j|}{min(|C_i|, |C_j|)}, \tag{13}$$

where $min(|C_i|, |C_j|)$ gives the size of the smaller cluster of C_i and C_j. When $|C_i \cap C_j| = \phi$, the overlapping between C_i and C_j is zero; when $C_i \subseteq C_j$ or $C_j \subseteq C_i$, $Over(C_i, C_j) = 1$. For all other cases, $Over(C_i, C_j)$ is a value in range (0, 1).

4.3 Coarse K-Medoids-Based Node Clustering

In this paper, we adopt coarse K-medoids clustering approach to detect communities. Intuitively, it exploits local high-potential nodes to assign initial clustering centers and adds non-center nodes to the lower and upper approximation sets. At the end of each round, it recalculate the center node based on trust cohesion. The above steps repeat until the center node of each cluster stabilizes.

Algorithm 1. Detection the Overlapping Community with Rough K-Mediods Clustering

```
Input :  The set of objects $V$;
         The initial cluster centers $U = u_1, u_2, \dots, u_k$;
         The weights of lower approximation set $w_{low}$;
         The weights of upper approximation set $w_{up}$
Output : The detected overlapping communities
         $C = \{C_1, C_2, \dots, C_n\}$
1  for $v_i \in V$ do
2     for $u_i \in V$ do
3        Calculate $p(v_i, u_i)$ using Equation (9);
4     end for
5     $p(v_i, C_l) = max\{p(u_1, v_i), p(u_2, v_i), \dots, p(u_k, v_i)\}$
6  end for
7  while $\underline{C_j}$ is not stable do
8     for $v_i \in V$ do
9        $\delta = p(v_i, C_l) - p(v_i, C_j)$
10       if $\delta \leq \alpha$ then
11          $v_i = \overline{C_j} \cap \overline{C_l}$
12       else
13          $v_i = \underline{C_j}$
14       end if
15       for $C_i, C_t \in C$ do
16          if $v_i \in (\overline{C_i} - \underline{C_i}) \cap (\overline{C_t} - \underline{C_t})$ then
17             $p(v_i, \overline{C_l}) = max\{p(v_i, C_i), p(v_i, C_t)\}$
18             $p(v_i, C_j) = min\{p(v_i, C_i), p(v_i, C_t)\}$
19          end if
20       end for
21    end for
22    Update the cluster center $u$ with Equation (12);
23 end while
24 return C;
```

The pseudo code of the algorithm is shown in Algorithm 1. In lines (1) through (6), it traverses all nodes and assigns non-center nodes to the clusters with the highest trust potential. In lines (8) through (14), for all $v_i \in V$, the algorithm computes its potential difference in C_i and C_l, i.e., $\delta = p(v_i, C_l) - p(v_i, C_j)$. If $\delta \leq \alpha$, i.e, the potentials of v_i in two clusters are similar, we assign v_i to the upper approximation set of the intersection of C_l and C_j; otherwise, to the lower approximation set of C_l. In lines (15) through (20), for any clusters $Ci, C_t \in C$, if there exists $v_i \in (\overline{C_i} - \underline{C_i}) \cap (\overline{C_t} - \underline{C_t})$, i.e., v_i is at the boundary of two clusters, we reassign the node. In line (22), after classifying all network nodes, we recalculate the center of each cluster using Equation (12). The algorithm terminates when a set of stable clusters are identified.

4.4 Complexity Analysis

We next analyze the time complexity of Algorithm 1. Given the network $G(V, E)$, $|E| = m$, and $|V| = n$, the trust potential computation complexity is primarily determined by the node trust influence range and inter-node trust. When the node trust influence range is one hop, the trust potential computation complexity is $O(m)$. When the node trust influence range is two hops, the trust potential computation complexity is $O(m + n^{3/\gamma})$, where $2 < \gamma < 3$ [42]. As the inter-node trust influence range increases, the trust potential computation complexity eventually reaches $O(n^2)$. The initial clustering centers are calculated via a hill climbing algorithm, whose computation complexity is $O(m)$. When the k initial clustering centers are determined, the initial overlapping community topology is obtained via coarse K-medoids clustering, whose complexity is $O(k(n-k)^2)$. The overlapping community topology optimization is approximately $O(k^2)$. Given that the number of communities $k << n$, the time complexity of the algorithm is $O(n^2 + k(n-k)^2) \sim O(kn^2)$.

5 Experimental Evaluation

5.1 Data Description

We implemented the proposed TLCDA scheme to evaluate the effectiveness and compared it to the state-of-the-art community detection schemes. We used a real social network with data extracted from Sina Microblog (www.weibo.com). We started with four users, extracted microblog data based on user follow relations, and added users incrementally. Due to the restriction of Sina Microblog, we could only collect the information of the first 200 followers of each user. The collected user data include the user relation list (fan, follow), personal attributes (user ID, nickname, location, gender, personal description and tag, and user type) and microblog information (microblog ID, user ID, publish time, and microblog content). We created four different networks via the mutual follow relations among users. Table 1 lists the basic parameters of these four networks. Fig. 2 illustrates their network topologies.

Fig. 2 shows that the networks do not contain any isolated nodes or node groups. The node degree distribution, as shown in Fig. 3, approximately follows a power-law distribution. That is, all four networks are standard dimensionless networks.

5.2 Evaluation Metrics

We evaluated the effectiveness of different community detection algorithms using topology cohesion and preference cohesion.

We employed the extended modularity index EQ to evaluate the topology cohesion. Assume that a network $G(V, E)$ has been partitioned into k communities $C_1, \dots, C_k$; $|E| = m$; D is the node degree function; and O_i is the total number of communities that the node has joined. In a real social network, this metric represents the closeness of nodes within each community. Note that interactions occur more frequently in communities with larger EQ values. The extended modularity is as follows.

$$EQ = \frac{1}{2m} \sum_k \sum_{i,j} \frac{1}{O_i O_j} \left(A_{i,j} - \frac{D(i)D(j)}{2m} \right), \tag{14}$$

where $A_{i,j}$ is the adjacent matrix of the network G.

We next quantify the preference similarity among the nodes in a community. Ideally, the preference similarity between two nodes of one community should be larger than that between two nodes from different communities. In traditional recommendation systems, the preference similarity is computed based on the property or type of goods purchased by a user. In this paper, we evaluated the user preference based on the topics of the microblogs published by the user.

Definition 9 (Preference Similarity). *For any two nodes $v_i, v_j \in V$ in a microblog network $G(V, E)$, the topic sets of the*

TABLE 1
Basic Parameters of Four Networks

network	number of nodes	number of links	average degree	average clustering coefficient	average path length	number of microblogs	number of shared microblogs
N1	5,731	43,549	15.198	0.148	3.649	14,602	3,133
N2	4,623	28,166	12.185	0.163	3.656	12,523	2,518
N3	3,815	18,535	9.717	0.167	3.605	11,279	2,064
N4	2,667	8,855	6.64	0.204	3.671	8,603	1,210

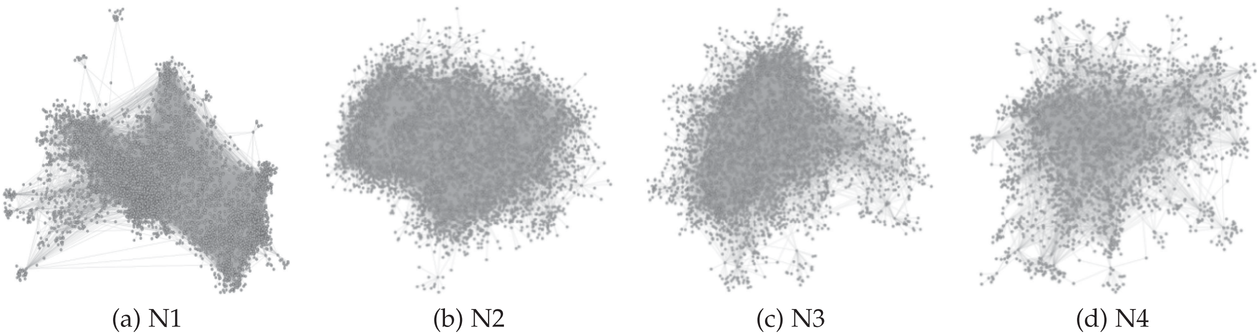

(a) N1 (b) N2 (c) N3 (d) N4

Fig. 2. The network topologies.

published microblogs are T_i and T_j. The preference similarity is defined as

$$pref(v_i, v_j) = \sum_{t_i \in T_i} \sum_{t_j \in T_j} exp(-dis(t_i, t_j)), \quad (15)$$

where $dist(t_i, t_j)$ represents the semantic distance between two microblog topics.

We computed the preference cohesion index and the average preference cohesion index (APCE) to quantify the network community preference cohesion.

Definition 10 (Preference Cohesion Index (PCE)). *Assume that the network $G(V, E)$ is divided into k communities $C_1, \ldots, C_k$. Then, the preference cohesion index of this network is expressed as*

$$PCE = \frac{\sum_{i=1}^{n} \sum_{u,v \in C_i} pref(u, v)}{\sum_{u,v \in G} pref(u, v)}, \quad (16)$$

where $PCE \in (0, 1]$, and $pref(u, v)$ represents the preference similarity between nodes u and v. In a real social network, this metric represents the similarity among nodes within each community. Note that, for communities with larger PCE values, their nodes have more similar attributes while there is a higher level of total preference cohesion for all communities in the network.

Definition 11 (Average Preference Cohesion Index). *For any community C_i, the average preference cohesion index is expressed as*

$$APCE = \frac{\sum_{u,v \in C_i} pref(u, v)}{|C_i|}, \quad (17)$$

where $|C_i|$ is the number of nodes in the community C_i. A larger APCE indicates a higher level of community preference cohesion.

5.3 Topology Cohesion Analysis

In evaluating the microblog based social network, we used microblog forwarding as direct node interaction to determine the inter-node trust relation. The average forwarding path lengths of four current networks are 3.649, 3.656, 3.605, and 3.671, respectively. Therefore, we set the maximal distance of the trust transfer d_{max} to four in calculating the relation trust. When calculating the trust potential, we set the parameter σ to 1.886 and the coarse clustering overlapping threshold to 0.75.

When calculating the inter-node attribute similarity, we selected two attributes, i.e., the numerical location information and the text tag information. For location attribute, we computed the similarity as follows: if both the province and city IDs are identical, then the similarity is 1; if the province IDs are identical but the city IDs differ, the similarity is $2/3$; otherwise, the similarity is 0. For tag attribute information, we preprocessed microblog data to create a user-tag bipartite network and calculate the tag attribute similarity.

5.3.1 Experimental Performance Analysis

Fig. 4 summarizes the effect of two parameters α and w_{up} on the community topology cohesion index EQ. α is the parameter for weighting the social similarity and attribute similarity; w_{up} is the upper approximation weight parameter w_{up} for coarse K-medoids clustering.

- When w_{up} is fixed, the EQ values increase along with increasing α, except for one abnormal point. This is because that, when α increases, the social-similarity-based trust has a larger weight such that the local network topology becomes more important in defining communities. This leads to larger overlapping communities and high EQ values. It also indicates that, in a real social network, if the detection scheme emphasizes more on the number of overlapped neighbors between nodes and on the closeness, the detected communities tend to have larger topology cohesion indices.
- When α is fixed, EQ decreases with increasing w_{up}. This is because a larger upper approximation weight produces a larger number of overlapping nodes in the detection and a smaller EQ. When w_{up} = 0, an upper

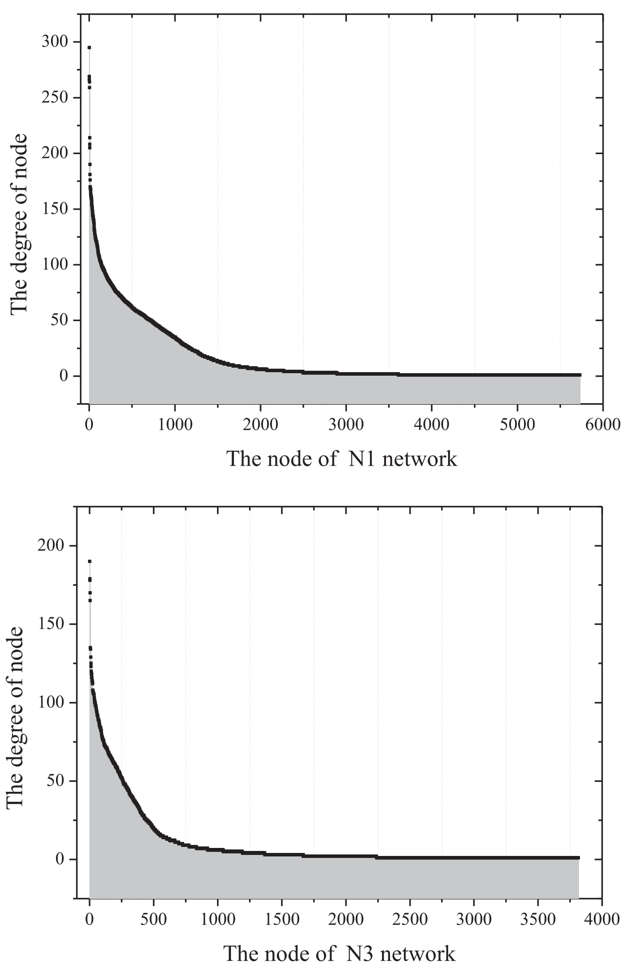


Fig. 3. The node degree distributions of four networks.

approximation set does not exist in coarse clustering, TLCDA converts to a non-overlapping community detection algorithm, and EQ attains the peak.

In summary, the results confirm that, the community topology produced by a non-overlapping community detection algorithm exhibits better modularity than that from an overlapping community detection algorithm.

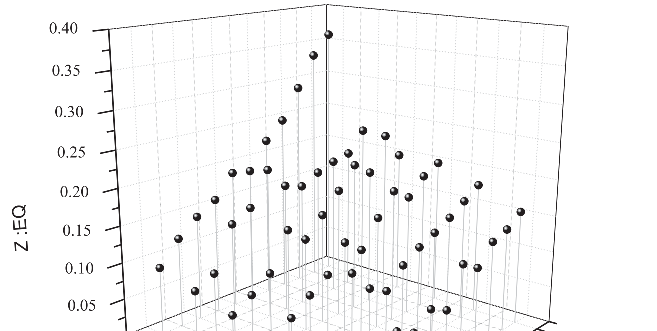


Fig. 4. The effect of weight parameter on EQ in N1 network.

5.3.2 Comparison to the State-of-the-Art

We next compared the effectiveness of TLCDA with the results from two widely adopted topology-optimization-based community detection algorithms, i.e., `Louvain` [43] and `LFM` (linear frequency modulation) [22], and a clique-percolation-based community detection algorithm, i.e., `CPM` [44], and three information-propagation-based community detection algorithms, i.e., `FluidC` [41], `GANXiSw` [45], [46] and `Infomap` [47].

Table 2 summarizes the comparison results. The fitness parameter α used in `LFM` is set in the range [0.6, 1.0]. The expected number of communities in `FluidC` is dynamically adjusted. The input parameter k of CPM algorithm is set to 4, i.e., the detection algorithms can produce communities of suitable size and structure sparsity. The parameter ov used in `GANXiSw` is set to 1.0; the tag filter threshold is set to 0.15; the maximum number of iterations is set to 10; and other parameters are set to the default values. The maximum EQ values of all algorithms are employed. The upper approximation weight parameter w_{up} is set to 0.1.

The preferable value range of community topology cohesion index EQ is [0.3, 0.7], as shown in recent community detection studies [48]. Table 2 reveals that the EQ values from TLCDA are lower than those from `Louvain`, `LFM` and `CPM`, higher than those from `FluidC` and `Infomap`, and are similar to those from `GANXiSw`. This is because network-topology-optimization-based community detection

TABLE 2
Comparing EQ Values from Different Algorithms in Four Networks

Algorithm		Louvain	LFM	FluidC	GANXiSw	Infomap	CPM	TLCDA
N1	# of communities	147	171	400	243	646	163	212
	EQ	0.359	0.287	0.091	0.184	0.116	0.273	0.208
N2	# of communities	112	99	325	174	406	103	159
	EQ	0.414	0.301	0.104	0.212	0.147	0.313	0.241
N3	# of communities	78	67	175	119	259	75	102
	EQ	0.352	0.283	0.107	0.193	0.186	0.275	0.201
N4	# of communities	44	42	100	63	141	51	52
	EQ	0.437	0.354	0.143	0.311	0.279	0.324	0.307

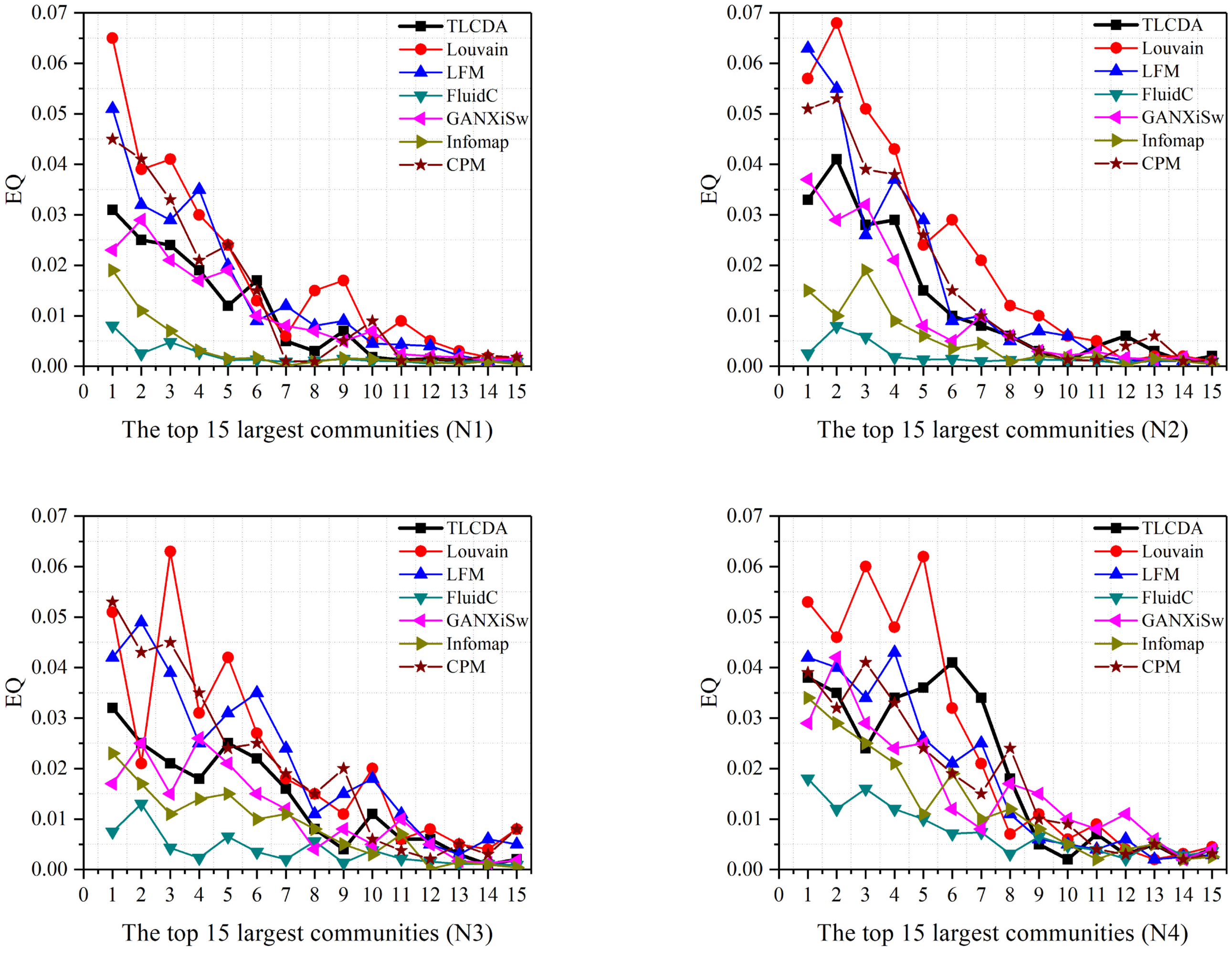


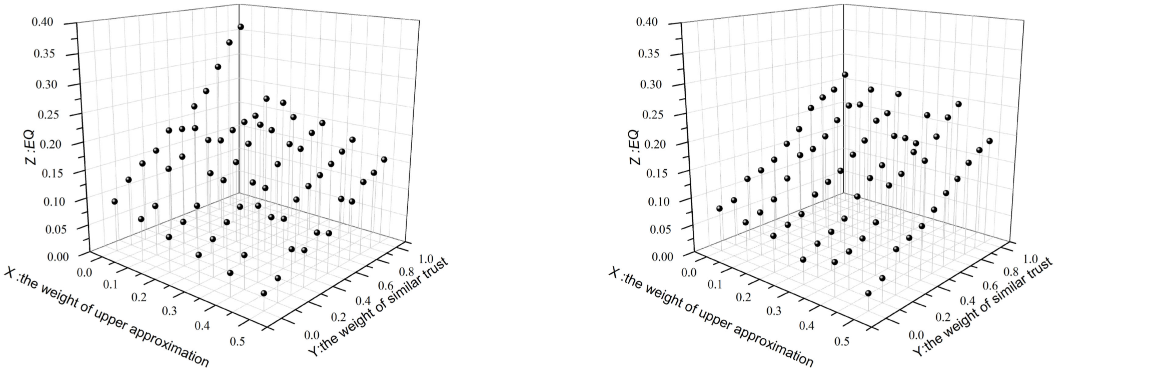
Fig. 5. The EQ distributions of Top-15 communities in four networks.

algorithms detect communities via the continuous optimization of the modularity function or self-defined fitness function, which produce higher EQ values. The clique-percolation-based community detection algorithm CPM exploits the interactions between nodes in the network and detects communities through maximum group filtering method, resulted in a higher EQ. `FluidC` has the lowest EQ values in all four networks, indicating that it is unsuitable for detecting communities from unknown networks, e.g., those used in the experiments. Table 2 shows that TLCDA produces significantly smaller numbers of communities than those from information-propagation-based algorithms. Merging communities with a higher level of overlapping helps to discover larger communities in the network.

Fig. 5 compares the EQ values of Top-15 communities from different algorithms. For modularity contribution, TLCDA is slightly worse than `Louvain`, `LFM` and `CPM`, slightly better than `GANXiSw`, while significantly outperforming `Infomap` and `FluidC`.

5.3.3 Coefficient Weights Analysis

To study the impact of the weight coefficient β on EQ, we varied the β value in $\{0.00, 0.25, 0.50, 0.75, 1.00\}$ on the N1 network and summarized the results in Fig. 6.



The effect of weight parameter on EQ (β=0.50) The effect of weight parameter on EQ (β=1.00)

Fig. 6. Comparing the EQ distributions of different β values.

- When β is set to 1.00, that is, only the relation-strength-based trust is considered, we got poor EQ values. This is because just considering the relationship between nodes in the current network, namely weibo forwarding, cannot effectively describe the local network structure. The potential attribute information of nodes is ignored in the calculation of the overall trust, which leads to poor topology cohesion for the detected communities.
- When changing β to other values, we observed better EQ values. The difference from adopting these β values is insignificant. In particular, β being 0.5 works well in the current network, which suggests that EQ is influenced by the duration and activeness of the current network. Giving that a different network may demand a different optimal β value in practice, we leave it as our future work to identify the optimal β value from the network settings.

In summary, in order to detect overlapping communities with stronger structures, it is necessary to adjust the value of β accordingly at runtime.

5.4 Preference Cohesion Analysis

5.4.1 Experimental Performance Analysis

We then studied the two parameters α and w_{up} on the community PCE. The results are summarized in Fig. 7.

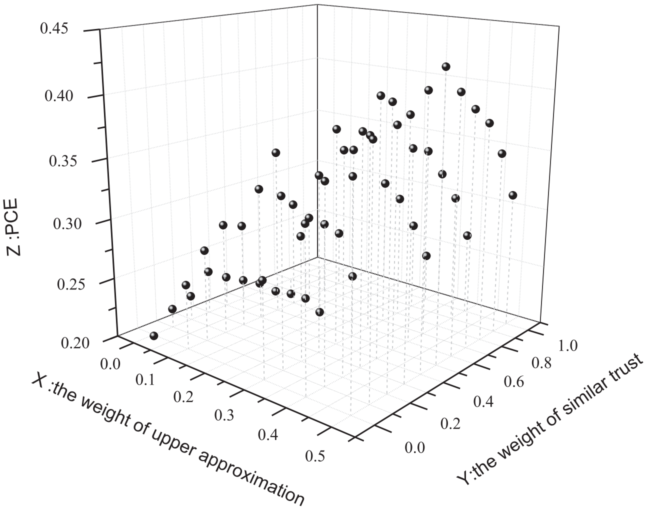


Fig. 7. The effect of the weight parameter on PCE in N1 network.

When the weight w_{up} is fixed, the PCE values from TLCDA increase initially and then decrease as the weight α increases. This is because the preference of microblog user is determined by both user groups and user attributes. Neither user attribute only ($\alpha = 0$) or user group only ($\alpha = 1$) can precisely reflect user preference in practice. This means that in a real social network, the social-similarity-based trust and the attribute-similarity-based trust should be considered simultaneously. To detect a community with a higher preference cohesion index, the weight coefficient α need to be set appropriately. When the weight α is fixed, the PCE values increase with increasing weight w_{up} values. This is because the number of overlapping nodes in a community shall increase significantly as the weight w_{up} increases, which leads to an increase in the sum of the preference similarities of all communities.

5.4.2 Comparison to the State-of-the-Art

We then compared the effectiveness of different algorithms on community preference cohesion in Table 3. TLCDA achieves superior preference cohesion over other algorithms. TLCDA creates a hybrid trust model via network topology, inter-node interaction information and attribute information; it then integrates these information in community detection, which significantly improves the preference cohesion of an identified community. The communities detected by `Louvain` or `LFM` have good preference cohesion because the node topology partially reflects node preference. The communities detected by `CPM` have better APCE values. This is because `CPM` exploits the network topology to capture the node preferences. While `Infomap` can identify more communities, the community preference cohesion is often sub-optimal.

Fig. 8 compares the APCE values from different algorithms for Top-15 communities. The top community identified by TLCDA has the best APCE value than those from other algorithms.

In summary, the topology cohesion and preference cohesion results showed that the proposed trust-based TLCDA scheme ensures community topology cohesion while enabling the detection of communities with better preference cohesion.

5.4.3 Coefficient Weights Analysis

To study the impact of the weight coefficient β on PCE, we varied the values in 0.00, 0.25, 0.50, 0.75, 1.00 on the N1 network and summarized the results in Fig. 9.

When the weight coefficient β becomes bigger, the PCE values of detected communities first increase and then decrease.

TABLE 3
Comparing PCE Values in Four Networks

network	parameters	Louvain	LFM	FluidC	GANXiSw	Infomap	CPM	TLCDA
N1	# of community preference	27714.342	29710.714	16323.278	33233.723	11978.232	32998.856	40044.875
	# of network preference				117433.651			
	PCE	0.236	0.253	0.139	0.283	0.102	0.281	**0.341**
N2	# of community preference	26275.248	23762.873	16958.527	32765.548	19261.536	25437.790	42605.681
	# of network preference				104682.263			
	PCE	0.251	0.227	0.162	0.313	0.184	0.243	**0.407**
N3	# of community preference	16760.595	14617.687	8724.693	16377.933	7500.175	17372.854	25332.223
	# of network preference				76532.395			
	PCE	0.219	0.191	0.114	0.214	0.098	0.227	**0.331**
N4	# of community preference	11375.182	12261.560	7189.509	15166.909	5958.427	15708.585	20386.670
	# of network preference				49243.212			
	PCE	0.231	0.249	0.146	0.308	0.121	0.319	**0.414**

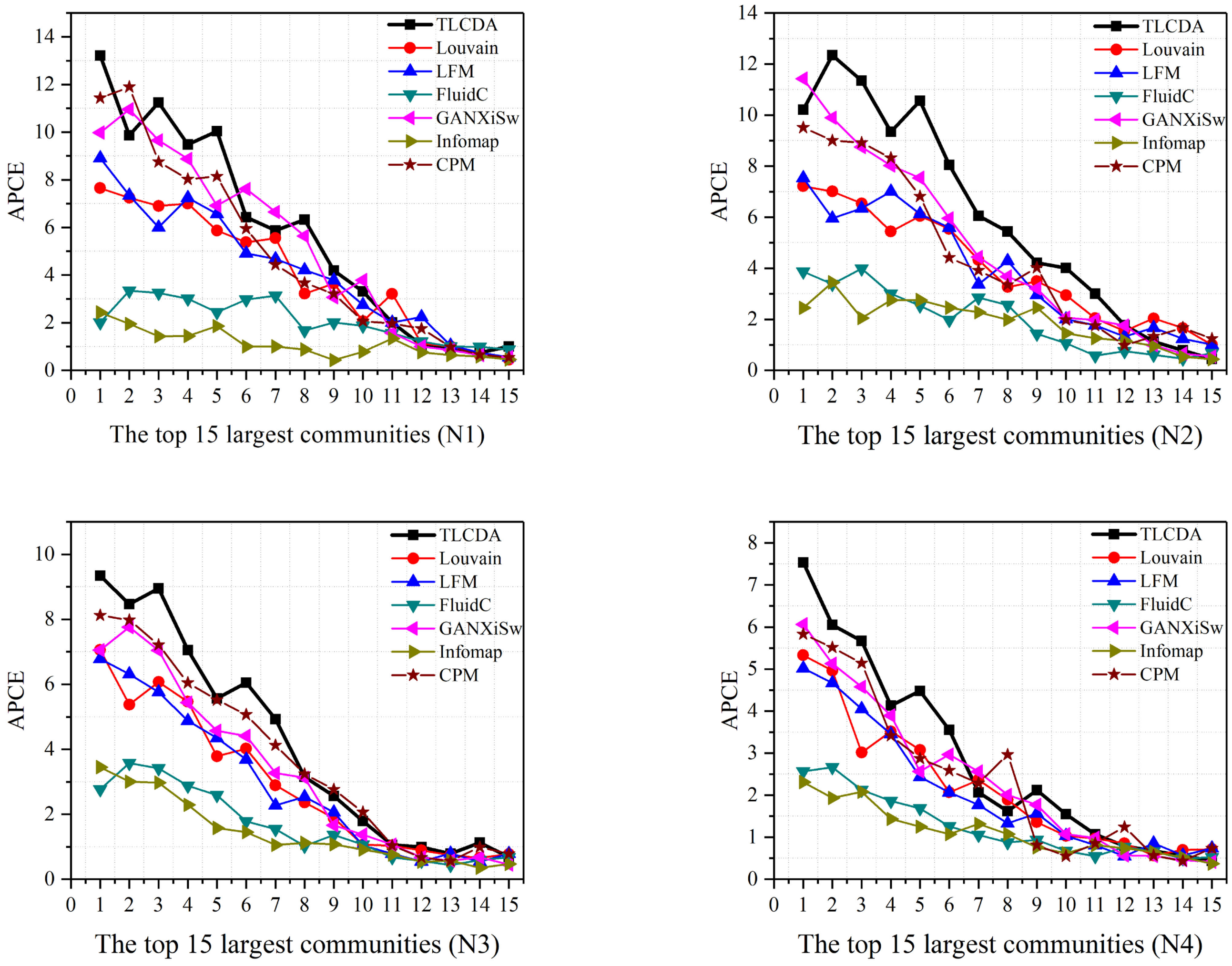


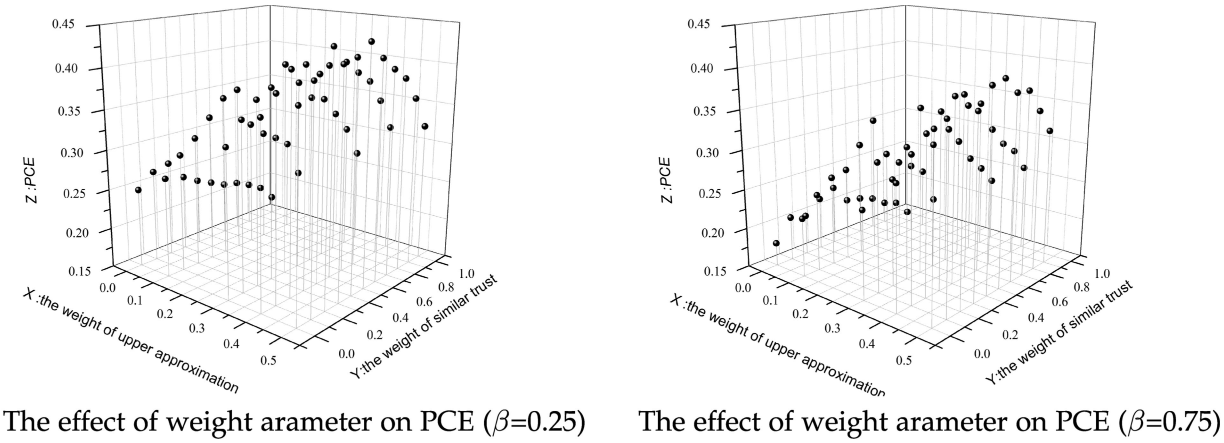
Fig. 8. Comparing the APCE distributions of Top-15 communities in four networks.

The best result appears when β is set to 0.25 in the current network. The is because, if we took more interaction information of nodes and less attribute information, a lot of nodes would inevitably have fewer interactions with neighbors, resulting in finding lots of fragmented communities. If the index of preference similarity across the network remains unchanged, the PCE values become worse as the nodes in the fragmented communities cannot be used in calculating the PCE.

In summary, assigning an appropriate β value can combine the node interaction, social information and attribute information effectively in the trust calculation.

5.5 Comparison with Existing Trust Models

We next compared our proposed model with existing well-known trust models based on the relation-strength [49] (referred as $Trust\_Rel$) and the node-similarity [50] (referred



The effect of weight arameter on PCE (β=0.25) The effect of weight arameter on PCE (β=0.75)

Fig. 9. Comparing the PCE distributions of different β values.

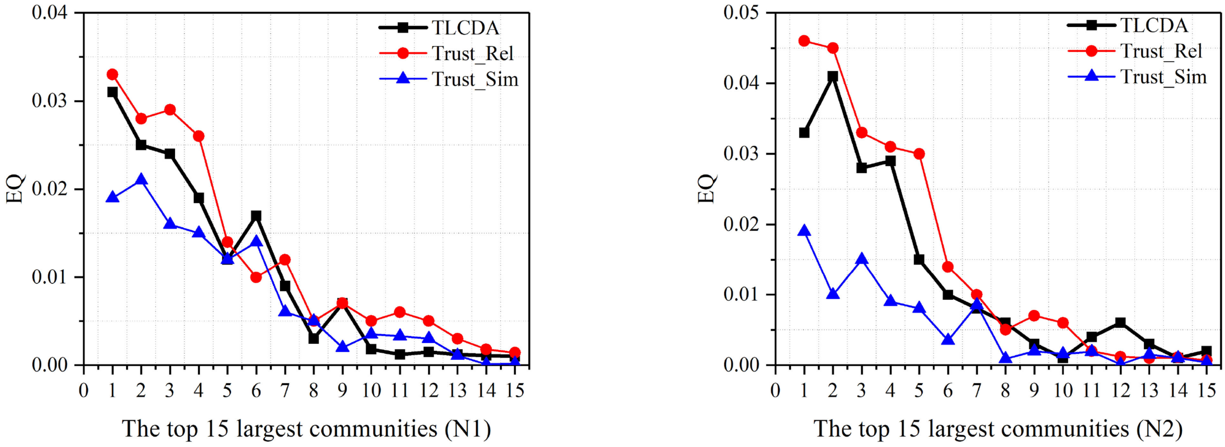


Fig. 10. Comparing the EQ distributions with different trust models.

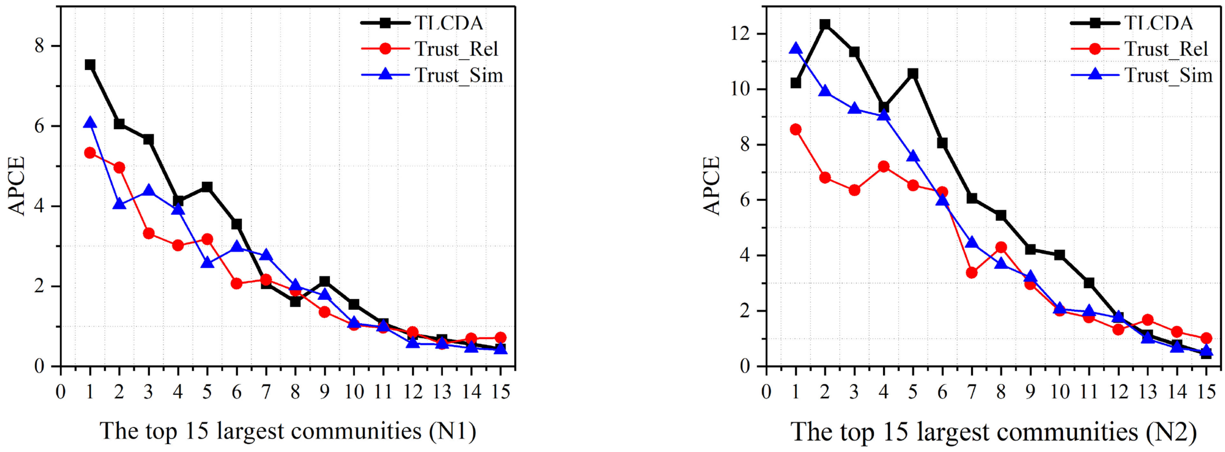


Fig. 11. Comparing the APEC distributions with different trust models.

as *Trust_Sim*). We evaluated the performance of different trust models on the N1 and N2 networks. The EQ and APEC values of the top 15 largest communities are shown in Figs. 10 and 11, respectively.

- Fig. 10 shows that the EQ values of *Trust_Rel* have better performance than those of TCLDA. This is because, in the current Weibo networks, the microblog forwarding (i.e., the interactions) occurs mainly between users that have fan relationship, the relation-strength based trust model can exploit the local network structures. Compared with TCLDA that considers the individual attributes of users, the topology of communities detected by the *Trust_Rel* based model are more compact and have higher EQ values.
- *Trust_Sim* has poor EQ values. This is because, it only considers user attributes and neglects the network structure information. The method is likely to place users that have similar attributes but no relationship in the same community. The network

topology of such communities is more divergent while their EQ values are lower.

- Fig. 11 shows that the APCE values of TCLDA are much better. This is because the users in social networks have multiple types of preferences, the partial activity information cannot fully capture the user's complete preferences. The hybrid trust model in TLCDA considers both the interaction and the individual attributes, which help to detect communities with high preference cohesion. From the figure, the APCE value improves when nodes that have similar preferences with neighbors are added into the community.

6 Conclusion

Community detection has become an important approach for studying social networks, in particular, for public sentiment monitoring, opinion leader detection and customized recommendation. To address the cold-start problem of the traditional approaches, this paper combines the inter-node relation strength and node attributes to propose a new hybrid trust computational model. A data field and coarse clustering method are employed to create the trust-modeling-based local overlapping community detection algorithm TLCDA. We evaluated the proposed algorithm and compared it with the state-of-the-art schemes. Our results showed that TLCDA achieves better topology cohesion and preference cohesion over existing schemes.

Modern social networks are transforming the manner in which people communicate and collaborate. It is important to develop adaptive community detection scheme to capture the changing user requirements and preferences. Trust model based community detection, while being proven effective for social networks, is still in its early stage. In our future work, we will integrate more network information, and study other types of network topologies.

Acknowledgments

The authors would like to thank the anonymous reviewers for their detailed and thoughtful feedback which improved the quality of this paper significantly. This work is fully supported by the National Natural Science Foundation of China Nos. 91846107, 71571058, and 71690235, Anhui Provincial Science and Technology Major Project Nos. 17030801001, and 18030801137, and the Fundamental Research Funds for the Central Universities No. PA2019GDQT0021.

References

[1] F. Zhang, Y. Zhang, L. Qin, W. Zhang, and X. Lin, "Finding critical users for social network engagement: The collapsed k-core problem," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 245–251.

[2] H. Djidjev and M. Onus, "Scalable and accurate graph clustering and community structure detection," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 5, pp. 1022–1029, May 2013.

[3] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *The Bell Syst. Tech. J.*, vol. 49, no. 2, pp. 291–307, 1970.

[4] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Nat. Acad. Sci. United States America*, vol. 99, no. 12, pp. 7821–7826, 2002.

[5] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Phys. Rev. E*, vol. 69, no. 6, 2002, Art. no. 66133.

[6] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Phys. Rev. E*, vol. 70, no. 6, 2004, Art. no. 66111.

[7] R. Guimera and L. A. Nunes Amaral, "Functional cartography of complex metabolic networks," *Nature*, vol. 433, no. 7028, pp. 895–900, 2004.

[8] M. E. J. Newman, "Analysis of weighted networks," *Phys. Rev. E*, vol. 70, no. 5, 2004, Art. no. 56131.

[9] E. A. Leicht and M. E. J. Newman, "Community structure in directed networks," *Phys. Rev. Lett.*, vol. 100, no. 11, 2008, Art. no. 118703.

[10] H. Shen, X. Cheng, K. Cai, and M.-B. Hu, "Detect overlapping and hierarchical community structure in networks," *Physica A: Statistical Mech. Appl.*, vol. 388, no. 8, pp. 1706–1712, 2009.

[11] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *J. ACM*, vol. 46, no. 5, pp. 604–632, 1999.

[12] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[13] M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. Griffiths, "Probabilistic author-topic models for information discovery," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2004, pp. 306–315.

[14] A. McCallum, X. Wang, and A. Corrada-Emmanuel, "Topic and role discovery in social networks with experiments on enron and academic email," *J. Artif. Intell. Res.*, vol. 30, no. 1, pp. 249–272, 2007.

[15] N. Pathak, C. DeLong, A. Banerjee, and K. Erickson, "Social topic models for community extraction," in *Proc. 2nd SNA-KDD Workshop*, 2008, pp. 1–10.

[16] F. Yan, J. Jiang, Y. Lu, Q. Luo, and M. Zhang, "Community discovery based on social actors' interests and social relationships," in *Proc. IEEE 4th Int. Conf. Semantics Knowl. Grid*, 2008, pp. 79–86.

[17] Z. Zhang, Q. Li, D. Zeng, and H. Gao, "User community discovery from multi-relational networks," *Decision Support Syst.*, vol. 54, no. 2, pp. 870–879, 2013.

[18] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proc. 4th ACM Conf. Recommender Syst.*, 2010, pp. 135–142.

[19] J. Wu and F. Chiclana, "A social network analysis trust-consensus based approach to group decision-making problems with interval-valued fuzzy reciprocal preference relations," *Knowl.-Based Syst.*, vol. 59, pp. 97–107, 2014.

[20] X. Deng, Y. Zhong, L. Lü, N. Xiong, and C. Yeung, "A general and effective diffusion-based recommendation scheme on coupled social networks," *Inf. Sci.*, vol. 417, no. 168, pp. 420–434, 2017.

[21] I. Barjasteh, R. Forsati, D. Ross, A. H. Esfahanian, and H. Radha, "Cold-start recommendation with provable guarantees: A decoupled approach," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 6, pp. 1462–1474, Jun. 2016.

[22] A. Lancichinetti and S. Fortunato, "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities," *Phys. Rev. E*, vol. 80, no. 1, 2009, Art. no. 16118.

[23] S. Gregory, "Finding overlapping communities in networks by label propagation," *New J. Phys.*, vol. 12, no. 10, 2010, Art. no. 103018.

[24] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in P2P networks," in *Proc. 12th Int. Conf. World Wide Web*, 2003, pp. 640–651.

[25] L. Fang and K. LeFevre, "Privacy wizards for social networking sites," in *Proc. 19th Int. Conf. World Wide Web*, 2010, Art. no. 351.

[26] G. Misra, J. M. Such, and H. Balogun, "IMPROVE - Identifying minimal PROfile VEctors for similarity based access control," in *Proc. IEEE Trustcom/BigDataSE/ISPA*, 2016, pp. 868–875.

[27] J. Leskovec and J. Mcauley, "Learning to discover social circles in ego networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 539–547.

[28] E. Gilbert and K. Karahalios, "Predicting tie strength with social media," in *Proc. 27th Int. Conf. Human Factors Comput. Syst.*, 2009, Art. no. 211.

[29] R. L. Fogués, J. M. Such, A. Espinosa, and A. Garcia-Fornes, "BFF: A tool for eliciting tie strength and user communities in social networking services," *Inf. Syst. Frontiers*, vol. 16, no. 2, pp. 225–237, 2014.

[30] J. Golbeck and J. Hendler, "Inferring binary trust relationships in web-based social network," *ACM Trans. Internet Technol.*, vol. 6, no. 4, pp. 497–529, 2006.

[31] T. Bhuiyan, Y. Xu, and A. Josang, "Integrating trust with public reputation in location-based social networks for recommendation making," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol.*, 2008, pp. 107–110.

[32] L. Xiong and L. Liu, "PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 7, pp. 843–857, Jul. 2004.

[33] C. C. Chen, Y. H. Wan, M. C. Chung, and Y. C. Sun, "An effective recommendation method for cold start new users using trust and distrust networks," *Inf. Sci.*, vol. 224, pp. 19–36, 2013.

[34] G. Guo, J. Zhang, and N. Yorke-Smith, "A novel recommendation model regularized with user trust and item ratings," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 7, pp. 1607–1620, Jul. 2016.

[35] S. Deng, L. Huang, G. Xu, X. Wu, and Z. Wu, "On deep learning for trust-aware recommendations in social networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 5, pp. 1164–1177, May 2017.

[36] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annu. Rev. Sociology*, vol. 27, no. 1, pp. 415–444, 2001.

[37] L. Lu and T. Zhou, "Link prediction in complex networks: A survey," *Physica A: Statistical Mech. Appl.*, vol. 390, no. 6, pp. 1150–1170, 2011.

[38] H. Bisgin, N. Agarwal, and X. Xu, "Investigating homophily in online social networks," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol.*, 2010, pp. 533–536.

[39] D. Li, S. Wang, W. Gan, and D. Li, "Data field for hierarchical clustering," *Int. J. Data Warehousing Mining*, vol. 7, no. 4, pp. 43–63, 2011.

[40] X. Yu, J. Yang, and Z.-Q. Xie, "A semantic overlapping community detection algorithm based on field sampling," *Expert Syst. Appl.*, vol. 42, no. 1, pp. 366–375, 2015.

[41] F. Parés, D. G. Gasulla, A. Vilalta, J. Moreno, E. Ayguade, J. Labarta, U. Cortes, and T. Suzumura, "Fluid communities: A Competitive, Scalable and Diverse Community, Detection Algorithm," in *Complex Networks & Their Applications VI*, 2018, pp. 229–240.

[42] M. E. J. Newman, S. H. Strogatz, and D. J. Watts, "Random graphs with arbitrary degree distributions and their applications," *Phys. Rev. E*, vol. 64, no. 2, 2001, Art. no. 26118.

[43] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Statistical Mech.: Theory Experiment*, vol. 2008, 2008, Art. no. P10008.

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

[45] J. Xie, B. K. Szymanski, and X. Liu, "SLPA: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process," in *Proc. IEEE 11th Int. Conf. Data Mining Workshops*, 2011, pp. 344–349.

[46] J. Xie and B. K. Szymanski, "Towards linear time overlapping community detection in social networks," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*, 2012, pp. 25–36.

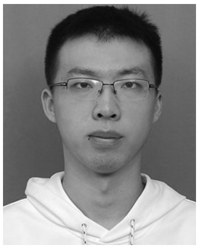
[47] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proc. Nat. Acad. Sci. United States America*, vol. 105, no. 4, pp. 1118–1123, 2008.

[48] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69, no. 2, 2004, Art. no. 26113.

[49] S. Nepal, W. Sherchan, and C. Paris, "STrust: A trust model for social networks," in *Proc. IEEE 10th Int. Conf. Trust Secur. Privacy Comput. Commun.*, 2011, pp. 841–846.

[50] X. Qian, H. Feng, G. Zhao, and T. Mei, "Personalized recommendation combining user interest and social circle," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 7, pp. 1763–1777, Jul. 2014.

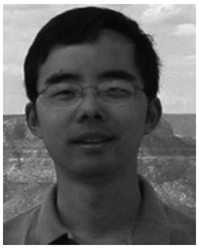
Shuai Ding received the PhD degree in MIS from the Hefei University of Technology, in 2011. He is an associate professor of information systems at the School of Management, Hefei University of Technology, China. He has been a visiting scholar with the University of Pittsburgh. His research interests include social networks, information systems, artificial intelligence, cloud computing, and business intelligence. He is a member of the IEEE.



Zijie Yue is working toward the PhD degree in the School of Management, Hefei University of Technology, China. His current research is in the area of social networks, data mining, and machine learning.

Shanlin Yang is a member of the Chinese Academy of Engineering, and the leading professor in management science and information system at the School of Management, Hefei University of Technology. He is the director of academic board of the Hefei University of Technology, and the director of the National-Local Joint Engineering Research Center of Intelligent Decision and Information System. He has won two second class prizes for State Scientific and Technological Progress Award, and six first class prizes for provincial and ministerial level science and technology award. His research interests include information systems, social networks, cloud computing, and artificial intelligence.

Feng Niu is working toward the master's degree in the School of Management, Hefei University of Technology, China. His current research is in the area of social networks, information systems, and cloud computing.



Youtao Zhang received the PhD degree in computer science from the University of Arizona, Tucson, AZ, in 2002. He is currently an associate professor of computer science, University of Pittsburgh, Pittsburgh, PA. His current research interests include memory systems and big data applications. He was the recipient of the US National Science Foundation Career Award, in 2005 and several best paper awards and best paper nominations.