Information Sciences 318 (2015) 123–143

Contents lists available at ScienceDirect

Information Sciences

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

κ-FuzzyTrust: Efficient trust computation for large-scale mobile social networks using a fuzzy implicit social graph

Shuhong Chen [a,b], Guojun Wang [a,*], Weijia Jia [c]

[a] School of Information Science and Engineering, Central South University, Changsha 410083, China
[b] School of Computer and Communication, Hunan Institute of Engineering, China
[c] Department of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai 200240, China

ARTICLE INFO

Article history:
Received 8 February 2014
Received in revised form 15 September 2014
Accepted 29 September 2014
Available online 13 October 2014

Keywords:
Trust inference
Fuzzy community
Mobile social network
Fuzzy implicit social graph

ABSTRACT

Large-scale mobile social networks (MSNs) facilitate connections between mobile devices and provide an effective mobile computing environment in which users can access, share, and distribute information. In MSNs, users may belong to more than one community or cluster, and overlapping users may play a special role in complex MSNs. For such MSNs, a key problem is how to evaluate or explain user trustworthiness. In this context, trust inference plays a critical role in establishing trusted social links between mobile users. To infer fuzzy trust relations between users in MSNs with overlapping communities, we propose an efficient trust inference mechanism based on fuzzy communities, which we call κ-FuzzyTrust. We propose an algorithm for detection of community structure in complex networks under fuzzy degree κ and construct a *fuzzy implicit social graph*. We then construct a mobile social context including static attributes (such as user profile and prestige) and dynamic behavioural characteristics(such as user interaction partners, interaction familiarity, communication location and time) based on the fuzzy implicit social graph. We infer the trust value between two mobile users using this mobile social context. We discuss the aggregation and propagation of trust values for overlapping users and indirect connected users. Finally, we evaluate the performance of κ-FuzzyTrust in simulations. The results show the validity of our fuzzy inference mechanism for behavioural trust relationships in MSNs. They also demonstrate that κ-FuzzyTrust can infer trust values with high precision.

1. Introduction

Large-scale mobile social networks (MSNs) integrate online social computing services and mobile devices, and allow mobile social users to discover and interact with friends and use distributed network services. MSNs represent relationships among sets of entities, and public accessibility via various mobile terminals makes such platforms ubiquitous [25,58]. For example, when searching for a subway station for a destination in a foreign country, a traveller can use her mobile phone to obtain advice provided by a friend's friend. For proper action in such cases, a trust value is required to establish a social link with this new friend through MSNs. The development of trust-based collaboration is an effective solution to reduce

* Corresponding author.
E-mail addresses: shchenannabell@csu.edu.cn (S. Chen), csgjwang@csu.edu.cn (G. Wang), jia-wj@cs.sjtu.edu.cn, wei.jia@cityu.edu.hk (W. Jia).

http://dx.doi.org/10.1016/j.ins.2014.09.058
0020-0255/© 2014 Elsevier Inc. All rights reserved.

vulnerability and exploit the potential of spontaneous social networking. The success of such attempts relies on trust levels among users and service providers [23,24,26].

1.1. Motivation

Unlike conventional mobile networks, unstable connectivity can lead to uncertain data, which makes trust inference a challenging issue in MSN scenarios. Despite strong security architectures, mobile devices can still be infected with malware that compromises individual privacy in ways that an ordinary computer cannot. Therefore, MSNs suffer from security and privacy issues when mobile users try to interact with others at any time and anywhere [13,42,27]. To balance the open nature of social networks and safeguard user privacy, it is important to build trust communities. These communities create an environment in which members can access, share, and distribute information in an open and honest way without concerns about privacy and fear of being judged. MSN users do not have any previous interactions, so it is important to establish an acceptable level of trust among participating users [36,15]. Therefore, trust inference plays a critical role in establishing trusted social links between mobile users [35,39,51].

However, it is not known how many communities there are in a given MSN. Moreover, some MSN users may belong to more than one community, so MSNs have an overlapping community structure [53]. In such MSNs, many people communicate mainly with their friends (such as family members and coworkers) through social network services. Mobile users share their roles within their MSNs via interactive behaviours, which increases the overall trustworthiness of the relationship between interacting users. Thus, a method for measuring user trustworthiness is necessary [10,54]. Online social relationships always depend on physical world relationships. Hence, we can infer a level of user trustworthiness underpinning the online community in which they exist according to some real-world attributes [6,11,17,22,38,40,59,62].

A key MSN problem is how to evaluate or explain user trustworthiness in terms of fuzzy relationships. There have been a number of studies on trust computation and inference in social networks. This research has established a foundation for our work. However, existing trust models commonly evaluate entities using predefined, static trust values, which have three disadvantages. First, it is difficult to determine the proper number of trust classes. Fine-grained definition may not be necessary and may even affect system efficiency, whereas coarse-grained definition may negatively affect system security. Moreover, different users may assign diverse judgments to the same object. Thus, a dynamic mechanism for trust class definition is required. More importantly, social network analysis in this type of research usually starts with a hard (i.e., not fuzzy, probabilistic, or possibilistic) graph-theory representation of the social network. Most previous work does not address this issue well [1,2,12,16,21,19,28,31,29,30,32,34,43,44,49,58,60,61], which may lead to inaccurate or unfair outcomes for trust decision-making.

1.2. Contributions of the present study

Here we use a fuzzy clustering algorithm to detect relations among users and present an efficient trust computation approach, which we call κ-FuzzyTrust, to determine the membership grade of these relations for large-scale MSNs using fuzzy community relationships. In our approach, a cluster is a subset of vertices of a social graph that is highly connected. In particular, we assume that the density of edges within a cluster (intra-cluster edges) is greater than the density of edges connecting vertices inside the cluster to vertices outside the cluster. On the basis of our findings, we suggest a method for quantifying trusting relations. The proposed approach helps not only in deciding communication paths via trustworthy users in a mobile environment but also in addressing trustworthiness security issues by ranking trustworthy relationships among users.

In this way, a communication path via trustworthy users under a mobile environment is suggested. With enhanced trustworthiness, the issue of security can also be addressed. This trust, along with the socially corrective mechanisms inherent in social networks, can also be applied to other domains. Social networking platforms provide a multitude of integrated applications that deliver particular functionality to users, including credential authentication in many diverse domains. For example, many sites support Facebook Connect as a trusted authentication mechanism [8,62]. The key innovations of our trust scheme go beyond the features of existing schemes in terms of the following aspects:

- To detect fuzzy relations for overlapping communities, we propose a fuzzy clustering algorithm for preprocessing for large-scale MSNs with overlapping communities according to fuzzy relations in group decision theory. Since different values of the fuzzy degree κ correspond to different community relations, we call this a *fuzzy implicit social graph* (or *fuzzy cluster*), where $\kappa \in \mathbb{R}$ and $0 \leqslant \kappa \leqslant 1$.
- Using this *fuzzy implicit social graph*, we provide an efficient method for computation of local trust to infer trust values between users in MSNs, which we call *κ-FuzzyTrust*. Moreover, we differentiate between direct and indirect trust, and explore stable and objective information for evaluating trust, which can weaken the effect of vicious nodes.
- To compute global trust values between indirectly connected nodes, we consider the propagation and aggregation of local trust values in MSNs by computing the gain and decay of trust.

Together, these features make our approach an accurate and efficient solution that can be used in MSN environments. Experiments using data from the Reality Mining set (http://realitycommons.media.mit.edu) confirm the effectiveness of our approach and reveal many interesting and useful findings.

The remainder of the paper is organised as follows. Related work is described in Section 2. Section 3 introduces our system model and Section 4 presents the concept of a fuzzy implicit social graph and an algorithm for its detection. κ-FuzzyTrust is presented in Section 5 and Section 6 discusses how to aggregate and transit local trust values. We analyse the performance of our trust model via simulations in Section 7. Section 8 concludes.

2. Related work

Our study builds on a large body of prior work that we broadly classify into two categories: (i) trust computing for social networks; and (ii) identification and clustering technology for fuzzy communities.

2.1. Trust computing for social networks

Without trust relationships among users in a mobile social environment, the reliability of the total network would decrease. Hence, many studies have attempted to discover relationships between communication entities using social trust models. Grandison and Sloman surveyed several existing trust models and defined trust as a "firm belief in the capability of an entity to act consistently, securely and reliably within a specified context" [22]. Kuada and Olesen proposed a provisioning and management approach based on a collaborative strategy for social relationships in mobile computing services [34]. Golbeck and Hendler [19] and Kim and Han [28] proposed methods for quantitative inference of trust between users for a recommendation system in a Web-based social network.

In MSNs, privacy concerns are attracting increasing attention [13,27]. Damopoulos et al. studied the privacy level of two services, Tethering and Siri, for the iPhone [13]. They implemented a DNS poisoning malware that redirects all or a subset of DNS requests to a DNS resolver under the control of the attacker. Anonymity is an integral part of a user's right to privacy. Kambourakis highlighted different issues related to anonymity and argued that it is a multifaceted and contextual concept [27]. To plan, schedule, and reflect on group activities, Kikin-Gil [32] and Counts [12] proposed mobile users in a mobile network to create privately shared group spaces on mobile devices whereby each group can communicate and collaborate. To enhance trustworthiness in social networks, Pezzi defined a social network as a means of cultivating collective intelligence and facilitating the development of self-organising communities [44]. According to this perspective, a social network and its services are provided by network nodes owned by members of the network rather than by centralised servers owned by the social network. Traditionally, social network platforms provide only marginal functionality for enhanced communication on mobile devices [21]. However, a truly mobile social network will offer functionality to improve services by considering the mobile behaviour patterns of users.

To support mobile awareness and collaboration, Oulasvirta et al. designed an approach called *ContextContact* [43]. Farnham and Keyani provided smart convergence through mobile group text messaging (*Swarm*) [16]. However, *ContextContact* and *Swarm* are designed to enhance communication within a large group including all of a user's contacts. Kim et al. [31,29] and Wang et al. [26,58,61] proposed a trust model using user profiles that is appropriate for online communities. Interest in analysis and utilisation of data obtained from smartphones has increased as the use of these devices has become more widespread. Balasubramaniyan et al. proposed a method to filter out spam voice calls on IP telephony systems that recognises relationships between users by analysing sustained call behaviour patterns (e.g., duration, frequency, recent history) extracted from call detail records (CDRs) [2].

To support mobile social relations, Ankolekar et al. extracted the behavioural pattern for smartphone users according to their contact lists and phone call histories [1]. Recommended user lists are helpful when a user wants to make decisions. Roth et al. [49] and Chen et al. [10] described an implicit social graph formed by users' contacts with contacts and groups of contacts, which is distinct from explicit social graphs in which users explicitly add other individuals as their "friends". The authors also presented a novel friend suggestion algorithm that utilises a user's implicit social graph to generate a friend group, given a small seed set of contacts who the user has already labelled as friends. However, Roth et al. did not consider the relative importance of different contact types and communication group homogeneity for inferring trust in determining the social relationships between users. Here we propose a κ-FuzzyTrust model for mobile network computing using a fuzzy implicit social graph that considers these issues.

2.2. Identification and clustering technology for fuzzy communities

Although the notion of community structure is easy to understand, construction of an efficient algorithm for detection of community structure is highly nontrivial [18,20,24,37,45,63]. The earliest work on the use of fuzzy relations for social network analysis was by Blin [5], who introduced the idea of using fuzzy relations in group decision theory. This work highlighted another property found in many networks, community structure, whereby network nodes are joined together in tightly-knit groups between which there are only looser connections. Blin proposed a new method for detecting such communities, built around the idea of using centrality indices to find community boundaries. The method was tested on computer-generated and real-world graphs whose community structures were already known, which revealed that this approach detects structure with high sensitivity and reliability.

Zadeh was the first to discuss a fuzzy similarity relation on pairs of nodes in a social network [64]. He defined the notion of "similarity" as essentially a generalisation of the notion of equivalence. In the same vein, a fuzzy ordering is a generalisation of the concept of ordering. For example, the relation $x \gg y$ (x is much larger than y) is a fuzzy linear ordering in the set of real numbers. Zadeh pointed out that a fuzzy ordering is a fuzzy relation that is transitive. In particular, a fuzzy partial ordering P is a fuzzy ordering that is reflexive and antisymmetric. Zadeh investigated various properties of similarity relations and fuzzy orderings proved an extended version of Szpilrajn's theorem as an illustration [64].

Bezdek et al. collected data from small groups of students in communications classes, and developed models based on reciprocal fuzzy relations that quantified notions such as distance to consensus [4,52]. They defined reciprocal relations over n alternatives and investigated the efficacy of this model of the group decision process. The authors also defined and analysed several measures of individual preference and group consensus, and used these to generate associated measures of distance to consensus. They identified various decision-oriented goals, and characterised these goals geometrically as subsets of a convex subset of a hyperplane in R^n. The authors described an application of this model for assessing the degree of consensus.

In another study, Bezdek et al. used a model to find fuzzy communities via multiple spectral clustering [3]. Clustering can be performed using either hard or fuzzy c-means [65] and Bezdek et al. performed validation using an index they called fuzzy modularity [3]. They presented a comprehensive introduction to the use of fuzzy models in pattern recognition and selected topics in image processing and computer vision. A single notation, presentation style, and purpose were used throughout. This results in an extensive unified treatment of many fuzzy models for pattern recognition. In the present study, the main topics are clustering and classifier design, with consideration of feature analysis relational clustering, image processing, and computer vision.

Broadly speaking, the above studies have improved our understanding of trust computing for MSNs with overlapping communities.

3. System model and problem formulation

For convenience, Table 1 lists symbols that are introduced later in the paper.

3.1. System model

Consider a mobile social network with overlapping communities. According to Reichardt and Bornholdt, communities (clusters) are groups of densely interconnected nodes that are only loosely connected to the rest of the network [47]. Instead, overlapping communities are seen visually as off-diagonal content in co-appearance images of the connection data. Let a graph $G = G(V, E, W)$ denote a mobile social network, where the set $V = \{1, 2, \ldots, n\}$ of vertices represents users (nodes), the set $E = \{e_1, e_2, \ldots, e_m\}$ of edges denotes relationships between these users, and $W = [w_{ij}]$ is the set of edge weights. w_{ij} refers to the weights for edges, that is, the ratio of the interaction counts between users i and j to total interactions. If user i trusts user j, then there exists a directional edge from user i to user j, and vice versa. In this paper, we ignore the direction of edges, so each edge is bidirectional. We assume that G can be divided into multiple subgraphs, where each subgraph G_i is a

Table 1
Main symbols for modelling and analysis.

Symbol	Description
G_i	Cluster including user i and his/her direct neighbours
w_{ij}	Weights for edges between users $i \rightarrow j$
κ	Fuzzy degree coefficient
$tr(i,j)$	Trust relationship between users i and j
$Ex(i,j)$	Expected trust value
$WEn(i,j)$	Weighted entropy
$\hat{tr}(i,j)$	Normalised value of $tr(i,j)$
$LOCTrust(i,j)$	Local trust between users i and j
$S_{a_k}(i,j)$	Similarity values for users i and j
$PTrust^{\kappa}(i,j)$	Profile trust value
$BTrust^{\kappa}(i)$	Basic trust value for user i
$StaticTrust^{\kappa}(i,j)$	Static trust value under fuzzy degree κ
$LTrust^{\kappa}(i,j)_l$	Location identity trust value between users i and j
$TTrust^{\kappa}(i,j)_l$	Intimacy of users i and j in fuzzy implicit social graph l
$\mathcal{IF}^{\kappa}(i,j)$	Interaction familiarity between users i and j
$\mathcal{PIR}^{\kappa}(i)_l$	Persistent interaction ratio for user i
φ_{ij}	Gain coefficient of trust from user i to j
$\varsigma_{i,j}$	Decay coefficient of trust from user i to j

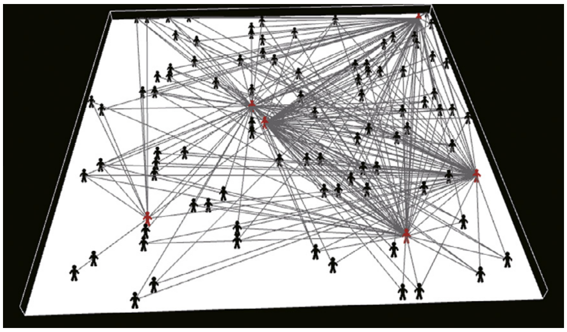


Fig. 1. System model for the Reality Mining data set [14].

cluster comprising user i and his/her direct neighbours, as shown in Fig. 1. The red users in Fig. 1 have very dense interconnections with others, and these users are often overlapping nodes. We suppose that there are N_i nodes in G_i. Each edge in G_i is formed by the sending and receiving of interactions. Any node in an MSN can belong to more than one community, which results in a complex overlapping community structure. If an individual is added to cluster G_i as a "friend", this implies that user i has at least some degree of knowledge about the individual being added. Such connectivity between individuals can be used to infer that a trust relationship exists between them. However, it does not describe the level of trust or the context of the relationship. Therefore, it is important to provide a quantitative method to describe the trust relationship.

According to the critical properties of MSNs, we construct a mobile context for our system model that includes the intersection of the profile and prestige of users, location, time, interaction partners, and interaction familiarity. The proposed mobile context is robust as it considers MSN features. This provides a basis for establishing a mobile context-aware trust-inferring model.

3.2. Problem formulation

According to the above discussion, most real-world networks comprise overlapping communities or a fuzzy similarity relation on pairs of nodes in MSNs. For such MSNs, a key issue is evaluation of trust relationships between users in overlapping communities. According to Zadeh, any weighted graph can be thought of as a fuzzy graph [64]. Therefore, we can consider the equivalence classes above as the skeletons of overlapping communities, and extend these skeletons by adding vertices when detecting overlapping communities. In our method, we first use a fuzzy relation to describe the trust level between users. Then an algorithm detects these fuzzy community structures or overlapping communities. We use our social trust context to determine the membership grade of the relation and infer trust values between users according to the community structures obtained by our algorithm. Finally, we propagate and aggregate local trust values to obtain the global trust value. We analyse the effectiveness and precision of our method using data from the Reality Mining data set (http://realitycommons.media.mit.edu) in experiments.

4. Fuzzy implicit social graph

4.1. Concept of a fuzzy implicit social graph

In MSNs, participants often organise and coordinate activities among multiple individuals using their mobile phones. The participants are divided into different communities (i.e., clusters). Cluster communication is often performed with members of a community existing in real life, such as members of a sports team, classmates, or family [21]. In general, global knowledge of network topology allows very efficient inference and propagation of trust values. However, collection and exchange of topology information in MSNs are cumbersome because of their intermittent connectivity and unpredictable mobility. Therefore, trust inference schemes for such networks typically rely on partial knowledge and the social trust context. Data collected from mobile phones have the potential to provide insight into the relational dynamics of individuals. In such an MSN service, users who link to each other usually indicate relationships, which leads to a social behavioural graph in which related users are connected through ties. To capture the strength of the relationship between two users in overlapping communities, we introduce the *fuzzy implicit social graph* (*fuzzy implicit cluster*). Similar to an ego-graph (or egocentric graph) [57], a fuzzy implicit social graph is a special form of social graph. It is a subset of vertices of an MSN that is highly connected and consists of a user of interest (ego) and his/her direct neighbours, who depend on fuzzy community relations. The structure of a *fuzzy implicit social graph* relies on a fuzzy parameter κ, which denotes the fuzzy degree of users' community relations, as discussed in detail in the next section. κ reflects the number of users in a communication community and whether a user belongs to this community or not. Edges in the fuzzy implicit social graph have both direction and weight. The direction of an edge is determined by whether it was formed by an outgoing contact initiated by the user, or an incoming communication received by the user. We ignore the direction of edges for simplicity. The weight for each edge between a user and his/her direct neighbours is determined by the contact behavioural patterns. The fuzzy implicit social graph for user i (the node with the largest degree) corresponding to κ is denoted by κ-Fuzzy G_i. An example of κ-Fuzzy G_i is illustrated in Fig. 2, where the edge weights are ignored for simplicity. Suppose the value of κ is 0.25 in Fig. 2; then user 4 belongs to 0.25-Fuzzy G_2 and 0.25-Fuzzy G_{10}, and user 9 belongs to 0.25-Fuzzy G_6 and 0.25-Fuzzy G_{10}.

4.2. FuzzyDetecting: detection of the fuzzy implicit social graph

We use the *FuzzyDetecting* algorithm to detect the fuzzy implicit social graph structure $G = G(V, E, W)$. Detecting the implicit social graph involves assigning labels to the users in V. Fuzzy clustering methods do not use hard assignment, and only assign a membership degree $fr(i, k)$ to every node i with respect to cluster j. For convenience, we provide some basic definitions for the *FuzzyDetecting* algorithm. In social networks, users who share more friends are more likely to be friends with each other and more likely to be members of the same community. Therefore, we introduce the d-distance neighbourhood and then present a similarity function based on shared neighbours.

4.2.1. Basic definitions for fuzzy detection

For any positive integer d, the d-distance neighbourhood set $Neigh_d(i)$ for node i contains every node for which the distance from node i is not greater than d:

$$Neigh_d(i) = \{j \in V | d(i,j) \leqslant d\}, \tag{1}$$

where $d(i,j)$ denotes the shortest distance between nodes i and j. Eq. (1) shows that $Neigh_d(i)$ is the set comprising node i and nodes that are adjacent to it. When i is an isolated node, there is only node i in $Neigh_d(i)$. If $d = 1$, these neighbours can be called the direct neighbours or the direct neighbourhood of node i. If $d > 1$, we can call these neighbours indirect neighbours or the indirect neighbourhood of node i. We now define d-similarity.

Definition 1. d-similarity is defined as follows:

$$Sim_d(i,j) = \frac{|Neigh_d(i) \cap Neigh_d(j)|}{\sqrt{|Neigh_d(i)| \cdot |Neigh_d(j)|}}, \tag{2}$$

where $| Neigh_d(*) |$ denotes the node count for $Neigh_d(*)$.

Because $0 \leqslant | Neigh_d(i) \cap Neigh_d(j) | \leqslant \min(| Neigh_d(i) |, | Neigh_d(j) |) \leqslant \sqrt{| Neigh_d(i) | \cdot | Neigh_d(j) |}$, we have $0 \leqslant Sim_d(i,j) \leqslant 1$.

Definition 2. For nonempty vertex set V of $G = G(V, E, W)$, a fuzzy relation $\forall (i,j) \in V \times V$. $fr(i,j)$ can be interpreted as the grade of membership of the ordered pair (i,j). We say that fr is a fuzzy relation in V, where $fr \in F(V \times V)$, and $F(V \times V)$ is the set of all the fuzzy relations of $V \times V$.

For later discussion, we introduce the max–min composition operation for fuzzy relations [48].

Definition 3. Let $S \in F(U \times V)$ and $T \in F(V \times W)$. Then the max–min composition operation for fuzzy relations can be defined as follows:

$$S \circ T \in F(U \times W)(u, w) = \vee_{v \in V}(S(u, v) \wedge T(v, w)), \tag{3}$$

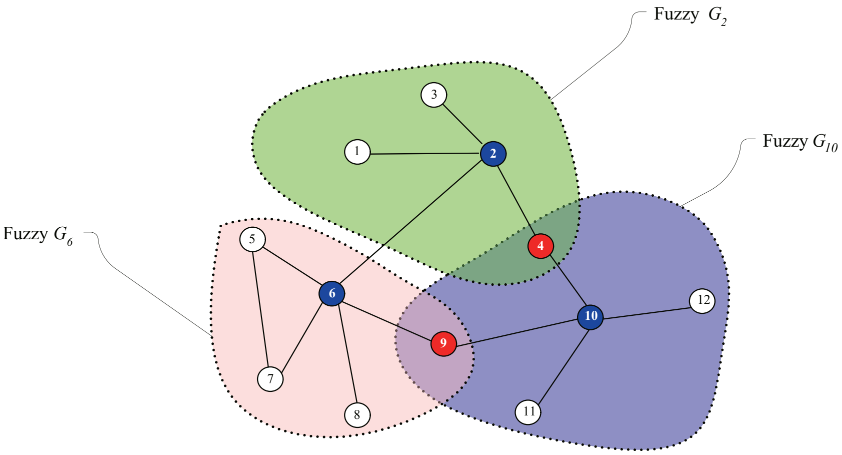


Fig. 2. A sample *fuzzy implicit social graph*.

where $(u, w) \in U \times W$.

Let $FR = [fr(i,j)]$ be a matrix of relational values on $V \times V$. Because $FR \in (V \times V), FR^n = FR^{n-1} \circ FR, n = 1, 2, \cdots, FR^0 = \mathbf{I}$, and $\mathbf{I}$ is the identity relation.

According to Ross, FR is a fuzzy equivalence relation if it satisfies *Reflexivity*, *Symmetricity*, and *Transitivity* conditions [48]. The transitive closure of a fuzzy relation FR is the minimal transitive relation $t(FR)$, which is obtained as [48]

$$t(FR) = FR^{2k}, \tag{4}$$

where $k = \lfloor \log_2^n \rfloor + 1$ and $n = \mid V \mid$.

Detection of clusters in the MSN $G = G(V, E, W)$ involves finding partitions of V. Hence, the objective of the *FuzzyDetecting* algorithm is to partition the fuzzy relation set FR of MSN G into C distinct subsets (C-partition) in a way that puts densely connected groups of vertices in the same community, where C is an integer and $1 \leqslant C \leqslant n$. Then a C-partition can be defined as follows [4].

Definition 4. A C-partition of FR is a set of values $fr(i,k)$ arrayed as a $C \times n$ matrix $FR^C = [fr(i,k)]$, where $fr(i,k)$ is the membership of user k in cluster i.

According to [3], the set of non-degenerate possibilistic C-partitions of FR can be defined as follows:

$$P_{possi} = \{FR^C \in \Re^{C \times n} : 0 \leqslant fr(i,k) \leqslant 1, \forall i, k\}, \tag{5}$$

where $0 \leqslant \sum_{i=1}^{C} fr(i,k) \leqslant C, \forall i$, and $\sum_{k=1}^{n} fr(i,k) < n, \forall k$.

Then the set of the fuzzy C-partition of FR is defined as follows:

$$P_{fuzzy} = \{FR^C \in P_{possi} : \sum_{i=1}^{C} fr(i,k) = 1, \forall k\}. \tag{6}$$

In the next subsection, we introduce how to determine a fuzzy C-partition and a fuzzy membership function.

4.2.2. Fuzzy detection method and fuzzy membership function

Our fuzzy detection method can be described as follows:

(1) We first transform the adjacency matrix $J = [J_{ij}]$ of the interaction graph $G = G(V, E, W)$ into a matrix of fuzzy relations, FR. According to Definition 1, FR is reflexive and symmetric. For the example in Fig. 2, the adjacency matrix and the matrix of fuzzy relation are shown in Fig. 3.

(2) We transform the fuzzy relation FR into the fuzzy equivalence relation $t(FR)$ using the max–min composition operation. For example, we let $k = 4$ because $n = \mid V \mid = 12$ in Fig. 3. As a result, we obtain $t(FR) = FR^8$ from Eq. (4).

(3) We select a fuzzy degree $\kappa \in [0, 1]$. Each value $\overline{fr}(i,j)$ of $t(FR)$ is replaced by a 1 or 0 according to the following rule:

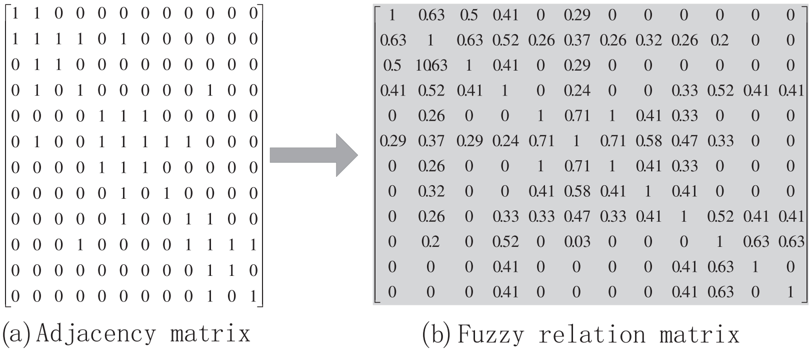
$$\overline{fr}(i,j) = \begin{cases} 1 & \text{if } \overline{fr}(i,j) \geqslant \kappa; \\ 0 & \text{if } \overline{fr}(i,j) < \kappa. \end{cases} \tag{7}$$

Iteration of this procedure for all values results in the conversion $t(FR) \rightarrow t_\kappa(FR)$.

(4) We compute the highest-order nonzero subdeterminant A of $t_\kappa(FR)$. Let $C = R(A)$, where $R(*)$ is the rank of matrix $*$.

(5) We construct a corresponding $n \times C$ assignment matrix $A_C = [a_1, \cdots, a_C]$ with $a_{ij} = \frac{Sim_d(i,j)}{\sum_{k=1}^{C} Sim_d(k,j)}$ for each $j = 1, 2, \cdots, C$ and $\sum_{j=1}^{C} a_{ij} = 1$ for each $i = 1, \cdots, n$.

(6) The membership of each community is defined as follows:



$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

(a) Adjacency matrix

$$\begin{bmatrix} 1 & 0.63 & 0.5 & 0.41 & 0 & 0.29 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.63 & 1 & 0.63 & 0.52 & 0.26 & 0.37 & 0.26 & 0.32 & 0.26 & 0.2 & 0 & 0 \\ 0.5 & 1063 & 1 & 0.41 & 0 & 0.29 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.41 & 0.52 & 0.41 & 1 & 0 & 0.24 & 0 & 0 & 0.33 & 0.52 & 0.41 & 0.41 \\ 0 & 0.26 & 0 & 0 & 1 & 0.71 & 1 & 0.41 & 0.33 & 0 & 0 & 0 \\ 0.29 & 0.37 & 0.29 & 0.24 & 0.71 & 1 & 0.71 & 0.58 & 0.47 & 0.33 & 0 & 0 \\ 0 & 0.26 & 0 & 0 & 1 & 0.71 & 1 & 0.41 & 0.33 & 0 & 0 & 0 \\ 0 & 0.32 & 0 & 0 & 0.41 & 0.58 & 0.41 & 1 & 0.41 & 0 & 0 & 0 \\ 0 & 0.26 & 0 & 0.33 & 0.33 & 0.47 & 0.33 & 0.41 & 1 & 0.52 & 0.41 & 0.41 \\ 0 & 0.2 & 0 & 0.52 & 0 & 0.03 & 0 & 0 & 0 & 1 & 0.63 & 0.63 \\ 0 & 0 & 0 & 0.41 & 0 & 0 & 0 & 0 & 0.41 & 0.63 & 1 & 0 \\ 0 & 0 & 0 & 0.41 & 0 & 0 & 0 & 0 & 0.41 & 0.63 & 0 & 1 \end{bmatrix}$$

(b) Fuzzy relation matrix

Fig. 3. Adjacency matrix and fuzzy relation matrix for the example in Fig. 2.

$$V_k = \{i | a_{ik} \geqslant \kappa; 1 \leqslant i \leqslant C\}, \tag{8}$$

where κ is a fuzzy degree that can convert an assignment into final clustering for $k = 1, 2, \cdots, n$. Using V_k, we define the fuzzy membership function $Q_{fuzzy}(FR^C)$ as

$$Q_{fuzzy}(FR^C) = \sum_{k=1}^{C} \left[\frac{S_f(V_k, V_k)}{S_f(V, V)} - \left(\frac{S_f(V_k, V)}{S_f(V, V)} \right)^2 \right], \tag{9}$$

where $FR^C \in P_{fuzzy}$, and

$$S_f(V_k, V_k) = \sum_{i,j \in V_k} \frac{fr(i,k) + fr(j,k)}{2} w_{ij}, \tag{10}$$

$$S_f(V_k, V) = S_f(V_k, V_k) + \sum_{i \in V_k, j \in V \setminus V_k} \frac{fr(i,k) - fr(j,k) + 1}{2} w_{ij}, \tag{11}$$

$$S_f(V, V) = \sum_{i,j \in V} w_{ij} \tag{12}$$

for $1 \leqslant i, j \leqslant C$. Our objective is to compute an assignment matrix by maximising the $Q_{fuzzy}(FR^C)$ function with appropriate k.

(7) Let the diagonal matrix be $D = [d_{ii}]$, where $d_{ii} = \sum_k J_{ik}$. Verma and Meilă used the top K eigenvectors of the generalised eigensystem $Jx = tDx$ to form a matrix whose rows correspond to original data points, and showed that after Euclidean normalisation of the rows, the eigenvectors are mathematically identical [55]. Their method is stable and appropriate for network clustering. Hence, we compute the top $C - 1$ eigenvectors of the eigensystem to obtain assignment matrix U_C, where C is the expected number of clusters.

(8) Choose an appropriate C to maximise the new $Q_{fuzzy}(FR^C)$ function in Eq. (9).

4.2.3. *FuzzyDetecting algorithm and complexity*

We are now in a position to present our *FuzzyDetecting* algorithm as Algorithm 1.

Algorithm 1. The *FuzzyDetecting* algorithm framework

```
Input:
    Input the adjacency matrix of $G, A = [J_{ij}], i,j = 1,2,3,\ldots,C$;
    The upper bound of the number of clusters, $C$;
    The fuzzy degree $\kappa$;
Output:
    The fuzzy implicit social graph, i.e., $\kappa$-Fuzzy $G_i, i = 1,2,3,\ldots,C$;
for Each vertex $i \in V$ do
    Calculate $Neigh_d(i)$ according to Eq. (1);
    Compute the diagonal matrix $D = [d_{ii}]$, where $d_{ii} = \sum_k J_{ik}$;
end for
for Each node pair $(i,j) \in V \times V$ do
    Calculate $Sim_d(i,j)$ by Eq. (3);
end for
Form the matrix of fuzzy relation, $FR$;
Transform $FR$ into the fuzzy equivalence relation, $t(FR)$;
$t(FR) \rightarrow t_\kappa(FR)$ using the formula ();
Compute the highest order nonzero subdeterminant, $A$, of $t_\kappa(FR)$;
$K = R(A)$;// $R(*)$ is the rank of matrix $*$;
Compute the $C$ eigenvectors of the generalised eigensystem $Jx = tDx$;
Construct the $C$ eigenvectors matrix $E_C = [e_1, e_2, \ldots, e_C]$;
for Each $i, 2 \leqslant i \leqslant C$ do
    Extract the sub matrix $E_k = [e_1, e_2, \ldots, e_i]$ from $E_C$;
    Compute $Q_{fuzzy}(FR^C)$ by Eq. (9);
end for
Pick up the maximal $Q_{fuzzy}(FR^C)$ and the corresponding $i$;
return $\kappa$-Fuzzy implicit social graph $G_i$;
```

Now we analyse the time complexity of the algorithm. Let n be the number of vertices and C the number of maximal clusters in the initial state of the algorithm. In steps 1–3, we apply the Dijkstra algorithm to calculate the shortest distance between nodes i and other nodes, and $\mathcal{O}(n^2)$ operations are needed to calculate $Neigh_d(i)$. In steps 5–7, $\mathcal{O}(n^2)$ operations are needed to calculate the similarity between each pair of initial communities. In steps 8–13, we take $\mathcal{O}(n^2)$ and $\mathcal{O}(n^3)$ operations to form the matrix of FR and to compute the eigenvectors, respectively. $\mathcal{O}(C)$ operations are needed to construct the C eigenvectors matrix in step 14. In steps 15–18, $\mathcal{O}(C * n^2)$ operations are used. Thus, the algorithm takes at most $\mathcal{O}(3n^2 + n^3 + C + C * n^2)$ operations. Because $n \gg C$, the algorithm takes at most $\mathcal{O}(n^3)s$ operations. In addition, we need to find all the maximal clusters in the network, which is a nonpolynomial problem. However, real-world social networks are often sparse and it is easy to find all the maximal clusters.

5. κ-FuzzyTrust: trust inference in MSNs according to a fuzzy implicit social graph

5.1. Local trust

As discussed above, our system contains multiple divisive communication clusters, which approximate real-life communities such as classmates, work colleagues, and family members. Each user can belong to multiple communication clusters according to fuzzy degree κ. We calculate the trust value within a communication cluster. Trust inference in the following involves the premise of fuzzy degree κ. We call this *κ-FuzzyTrust*.

Let $\mathbf{U}$ be the universe set of discourse, and let f and h be random functions with a stable tendency $f : \mathbf{U} \to [0,1]$ and $h : \mathbf{U} \to [0,1]$, respectively. For convenience, we give the fuzzy trust relationship R between user i and user j under the *κ-FuzzyTrust* model as a tuple of $\langle f, h \rangle$.

Definition 5. For any two users i and j in MSN G, the trust relationship R indicates the degree of trust and the trust value between them, and can be expressed as

$$tr(i,j) \triangleq R\langle Ex(i,j), WEn(i,j)\rangle, \tag{13}$$

where $Ex(i,j)$ is an expected trust value and $WEn(i,j)$ is a weighted entropy.

In this paper, $tr(i,j)$ is the basic element of trust space. According to Definition 5, $Ex(i,j)$ indicates the basic degree of trust between users i and j. $WEn(i,j)$ reflects the importance of the trust relationship between users i and j for user i. To derive $tr(i,j)$ under the *κ-FuzzyTrust* model, the trust computation includes static factors and dynamic factors. The static factors include user profile attributes and user prestige. The dynamic factors include user interaction partners, interaction familiarity, communication location, and time. We call these *static trust* and *dynamic trust*.

Definition 6. For user i and any other user j in the same fuzzy implicit social graph, the local trust between users i and j under degree κ is defined as

$$LOCTrust(i,j) \triangleq \langle \hat{tr}(i,j), \kappa\rangle, \tag{14}$$

where $\hat{tr}(i,j)$ is the normalised value of $tr(i,j)$, that is, $\hat{tr}(i,j) \triangleq R\langle \widehat{Ex}(i,j), \widehat{WEn}(i,j)\rangle$, and κ is a fuzzy degree coefficient.

In the following, we show how to infer these trust values.

5.2. Static trust computation

As discussed above, static trust includes two aspects: profile trust and prestige trust.

5.2.1. Profile trust

Users have attributes such as occupation, affiliations, age, address, and nationality. Identifiable attributes can distinguish between users. For example, two users can have relation such as "colleagues in the same university" and "members of the same research institute". If we assume that user i has h types of attributes $(a_1, a_2, \ldots, a_h)$, then user i is denoted as a set of h types of attributes according to

$$user(i) = \{a_1, a_2, \ldots, a_h\}. \tag{15}$$

Let $user(i).a_k$ denote attribute a_k of user i and let $\mathbf{R}^C_{a_k} = user(i) \bigcap user(j) = \{\langle user_i, user_j\rangle \in E : s.t.user(i).a_k = user(j) \cdot a_k\}$ denote the relationships between users i and j who share attribute a_k. To calculate the similarity between users, we first determine the similarity values between users i and j who share attribute a_k as

$$S_{a_k}(i,j) = \frac{|\mathbf{R}^C_{a_k}|}{h}, \tag{16}$$

where h is the attribute count for users.

To infer the profile trust value between users i and j, we evaluate their coincident attributes as follows:

$$\delta_{a_k}(i,j) = \begin{cases} 1, & \text{if } user(i) \cdot a_k = user(j) \cdot a_k; \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

Thus, the profile trust value between users i and j under fuzzy degree κ is calculated according to

$$PTrust^{\kappa}(i,j) = \sum_{k=1}^{h} S_{a_k}(i,j)\delta_{a_k}(i,j). \tag{18}$$

Eq. (18) shows that the greater the number of attributes shared by users i and j, the greater is the trust value.

Remark. For a given κ, users i and j can belong to multiple clusters. In each fuzzy cluster, the profile trust value $PTust^{\kappa}(i,j)$ remains the same. Although users i and j are not in the same fuzzy implicit social graph, we let $PTust^{\kappa}(i,j) = 0$.

5.2.2. Prestige trust

In general, the greater the number of neighbouring users that user i has, the higher is his/her prestige. Various measurement approaches can be used to calculate the prestige of mobile users, such as proximity-based [41], rank-based [57], and degree-based prestige computation [7]. In particular, the degree-based approach is an efficient and quick method for evaluating basic trust and it has been widely used. Therefore, we used degree-based prestige evaluation to obtain the basic trust value for each MSN user. Let $BTrust^{\kappa}(i)$ be the prestige trust value for user i. Then we have

$$BTrust^{\kappa}(i) = \frac{Neigh(i)}{n-1}, \tag{19}$$

where $Neigh(i)$ is the number of direct neighbours of user i and n is the total number of users in the social network.

Remark. As discussed above, since user i can belong to multiple clusters according to the value of κ, we compute the average as the final prestige trust. Assume that user i belongs to C clusters under fuzzy degree κ and that $BTrust^{\kappa}(i)_l$ is the prestige trust for cluster l. Then we have

$$BTrust^{\kappa}(i) = \frac{1}{C}\sum_{l=1}^{C} BTrust^{\kappa}(i)_l. \tag{20}$$

Hence, we obtain the static trust value under fuzzy degree κ according to

$$StaticTrust^{\kappa}(i,j) = PTrust^{\kappa}(i,j) + BTrust^{\kappa}(i). \tag{21}$$

5.3. Dynamic trust computation

We compute dynamic trust according to dynamic behavioural characteristics, such as user communication location and time, interaction partners, and interaction familiarity. For convenience, we define the interaction identity as follows.

Definition 7. Interaction identity represents the degree of homogeneity of interactions between users in MSNs with respect to a particular attribute.

Calculation of the interaction identity depends on the interaction location and time. Hence, a trust value based on identity includes *location identity trust* and *time identify trust*.

5.3.1. Location identity trust computation

Assume users i and j of a κ-Fuzzy implicit social graph in a social network have several expected locations, such as home, apartment, and office. Let $Loc(i)$ be the communication location of user i. To obtain the location identity trust value between users i and j in fuzzy implicit social graph l, we define the following functions:

$$LTrust^{\kappa}(i,j)_l = \begin{cases} 1 & \text{if } Loc(i) = Loc(j) = Home; \\ 0.8 & \text{if } Loc(i) = Loc(j) = Apartment; \\ 0.6 & \text{if } Loc(i) = Loc(j) = Office; \\ 0.4 & \text{if } Loc(i) = Loc(j) = Others; \\ 0 & \text{if } Loc(i) \neq Loc(j). \end{cases} \tag{22}$$

$LTrust^{\kappa}(i,j)$ is experimentally set according to the user's location. For example, consider four mobile users U_1, U_2, U_3, and U_4. Assume users U_1 and U_2 engage in social activities at home. "Home" is a very private and trustworthy place, in contrast to public places such as "Railway station" and "Office". Therefore, users who engage in social activities at home have higher trust values than users at a railway station or in an office. Therefore, the weight for "home" is set to 1 and a greater trust value between users U_1 and U_2 is assigned. Users U_1 and U_3 engage in social activities both in the office and in an apartment, which are assigned lower weights compared to "home". Users U_1 and U_3 probably have a colleague relationship, so the trust value between them is less than that between users U_1 and U_2. Therefore, we assign values of 0.8 and 0.6 to "apartment" and "office", respectively. Besides, if users U_1 and U_4 happened to be in the same place (e.g., a railway station) at the same time, user U_1 may or may not know user U_4. Therefore, there is a weak trust value between users U_1 and U_4, which we set to 0.4.

Remark. If users *i* and *j* belong to multiple clusters according to κ, we select the maximum location identity trust as the final $LTrust^{\kappa}(i,j)$. Assume users *i* and *j* belong to *C* clusters under fuzzy degree κ. Then $LTrust^{\kappa}(i,j)$ can be obtained as

$$LTrust^{\kappa}(i,j) = \max\{LTrust^{\kappa}(i,j)_1, LTrust^{\kappa}(i,j)_2, \ldots, LTrust^{\kappa}(i,j)_C\}. \tag{23}$$

5.3.2. Time identity trust computation

Suppose that user *i* communicates with user *j* via their mobile phones at a certain time. The call moment refers to the time when communication occurs between users, and a different call moment implies a different trust relationship. For simplicity, we divide a clock cycle into two time slots, *public time* and *private time*. The trust value for a call occurring in *private time* is higher than that for a call occurring in *public time*. In general, we define working hours (8:00 am → 6:00 pm) as public time, and 6:00 pm → 8:00 am as private time. Let $D_{public}(i,j)$ and $D_{private}(i,j)$ denote the call duration in *public time* and *private time*, respectively, between users *i* and *j*. We can obtain $D_{public}(i,j)$ and $D_{private}(i,j)$ from recent CDRs. Let $TTrust^{\kappa}(i,j)_l$ denote the intimacy between users *i* and *j* in fuzzy implicit social graph *l*, which we calculate according to

$$TTrust^{\kappa}(i,j)_l = \frac{D_{private}(i,j)}{D_{private}(i,j) + D_{public}(i,j)}. \tag{24}$$

If users *i* and *j* belong to *C* clusters according to κ, we compute the average value of $TTrust^{\kappa}(i,j)_l$ $(l = 1, 2, \ldots, C)$ as the final $TTrust^{\kappa}(i,j)$ according to

$$TTrust^{\kappa}(i,j) = \sum_{l=1}^{C} \frac{TTrust^{\kappa}(i,j)_l}{C}. \tag{25}$$

5.3.3. Interaction familiarity trust

Interaction familiarity ($\mathcal{IF}$) indicates the level of interactions and the relationships among users who are directly connected. $\mathcal{IF}$ between two users is usually generated through their social interactions. In other words, the value of $\mathcal{IF}$ represents the strength of the relationship between users *i* and *j* [46]. For example, if users *i* and *j* communicate frequently, then they are considered to have higher $\mathcal{IF}$. To address this issue, we consider the contact frequency and direction to calculate $\mathcal{IF}$. If the interaction frequency between users *i* and *j* is higher than that between users *i* and *h*, this indicates that user *j* is more important than user *h* to user *i*. Let $IA_{i \to j} = \{i \to j\}$ and $IA_{j \to i} = \{j \to i\}$ denote the interaction sets between users *i* and *j* initiated by user *i* and user *j*, respectively. Moreover, let $IA_{down}(i)$ denote the set of interactions initiated by user *i*. Otherwise, $IA_{up}(i)$. We express the importance of the interaction frequency through the interaction weight. We assume that the interaction weight decays exponentially at a rate λ over time. Furthermore, we assume that an interaction at the current time makes a contribution of 1 to $\mathcal{IF}$, whereas an interaction that occurred λ ago contributes 1/2, and so on [49]. Hence, we can obtain the weights for IA_{down} and IA_{up} in cluster *l* as

$$IA_{down}^{weight}(i \to j)_l = \sum_{i \to j \in IA_{i \to j}} \frac{1}{2}^{\frac{t_{current} - t(i \to j)}{\lambda}} \tag{26}$$

and

$$IA_{up}^{weight}(j \to i)_l = \sum_{j \in IA_{j \to i} \to i} \frac{1}{2}^{\frac{t_{current} - t(j \to i)}{\lambda}}, \tag{27}$$

where $t_{current}$ is the current time and $t(*)$ is the timestamp for contact $*$.

We denote the interaction familiarity between users *i* and *j* in cluster *l* under fuzzy degree κ as $\mathcal{IF}^{\kappa}(i,j)$. The interaction direction indicates whether the contact is initiated by user *i* or not. In general, contacts initiated by the user are more significant than those he/she did not initiate. To denote the importance of different directional contacts, we introduce the coefficient α $(0 \leqslant \alpha \leq 1)$, which indicates the relative importance. Then we obtain

$$\mathcal{IF}^{\kappa}(i,j)_l = \alpha IA_{down}^{weight}(i \to j)_l + (1 - \alpha) IA_{up}^{weight}(j \to i)_l. \tag{28}$$

Similarly, if users *i* and *j* belong to *C* fuzzy clusters according to κ, we compute the average value of $\mathcal{IF}^{\kappa}(i,j)_l$ $(l = 1, 2, \ldots, C)$ as the final $\mathcal{IF}^{\kappa}(i,j)$:

$$\mathcal{IF}^{\kappa}(i,j) = \sum_{l=1}^{C} \frac{\mathcal{IF}^{\kappa}(i,j)_l}{C}. \tag{29}$$

5.3.4. Interaction evolution trust

The evolution of a user's interaction partners indicates the degree of affinity between the user and a cluster. In general, if a user has a persistent set of contact partners, then he/she has high trust in these partners. To capture the evolution of a users interaction partners, we define the *persistent interaction ratio* ($\mathcal{PIR}$) as the evolution of interaction trust. For i, let $IR(\tau_1)$ and $IR(\tau_2)$ be the sets of interaction partners in cluster l under fuzzy degree κ during time periods τ_1 and τ_2, respectively. $\mathcal{PIR}$ for user i can be expressed as

$$\mathcal{PIR}^{\kappa}(i)_l = \frac{IR(\tau_1)\cap IR(\tau_2)}{IR(\tau_1)\cup IR(\tau_2)}. \tag{30}$$

According to Eq. (30), if user i contacts disjoint sets of users during τ_1 and τ_2, then $\mathcal{PIR}(i) = 0$. However, if user i has a persistent set of contact partners, then the metric would be greater than zero and increases with the number of persistent contact partners.

We compute the average value of $\mathcal{PIR}^{\kappa}(i)_l$ $(l = 1, 2, \ldots, C)$ as the final $\mathcal{PIR}^{\kappa}(i)$ for user i belonging to C clusters:

$$\mathcal{PIR}^{\kappa}(i) = \sum_{l=1}^{C}\frac{\mathcal{PIR}^{\kappa}(i)_l}{C}. \tag{31}$$

We can then compute the dynamic trust value under fuzzy degree κ as

$$DynamicTrust^{\kappa}(i,j) = LTrust^{\kappa}(i,j) + TTrust^{\kappa}(i,j) + \mathcal{IF}^{\kappa}(i,j) + \mathcal{PIR}^{\kappa}(i). \tag{32}$$

5.4. Local trust inference

As discussed above, the final local trust value is calculated as a combination of static and dynamic trust values. Hence, we compute the local trust under fuzzy degree κ according to

$$Ex(i,j) = \rho \cdot StaticTrust^{\kappa}(i,j) + (1-\rho)\cdot DynamicTrust^{\kappa}(i,j), \tag{33}$$

where ρ $(0 \leqslant \rho \leqslant 1)$ is a weight coefficient that is empirically set. This parameter can be learnt from the social interaction history.

According to Definition 6, we normalise the value of $Ex(i,j)$ as follows:

$$\widehat{E}x(i,j) = \frac{Ex(i,j)}{\sum_{k=1}^{N_l} Ex(i,k)}, \tag{34}$$

where N_l is the number of users in cluster l.

We calculate $WEn(i,j)$ as

$$WEn(i,j) = -w_{ij}\cdot Ex(i,j)\cdot \log(Ex(i,j)), \tag{35}$$

where w_{ij} is the weighted coefficient for contact between users i and j, obtained for our system model as

$$w_{ij} = \frac{|IA_{i\to j}| + |IA_{j\to i}|}{|IA_{down}(i)| + |IA_{up}(i)|}. \tag{36}$$

We calculate $\widehat{WEn}(i,j)$ as

$$\widehat{WEn}(i,j) = -\hat{w}_{ij}\widehat{E}x(i,j)\log(\widehat{E}x(i,j)), \tag{37}$$

where

$$\hat{w}_{ij} = \frac{w_{ij}}{\sum_{k\in G_l} w_{ik}}. \tag{38}$$

Hence, we obtain the local trust model: $LOCTrust(i,j) \triangleq \langle \hat{tr}(i,j), \kappa\rangle$.

6. Trust aggregation and transitivity

6.1. Trust aggregation

6.1.1. Gain of trust

Users i and j can belong to multiple communication clusters simultaneously according to the fuzzy parameter κ. In general, the relationship between users i and j involves greater trust in this case. Hence, trust has a profit property whereby the trust value increases for a more trustworthy target. This profit property indicates that a gain of trust occurs when user i tries to trust user j. In other words, as the number of communication groups to which users i and j both belong increases, the gain of trust increases. For example, assume users i and j both belong to "apartment" and "office" clusters, whereas users i

and k only have cluster “Apartment” in common. Then the trust value between users i and j is higher than that between users i and k. To capture this notion, we assume that the trust value increases exponentially with the number of common clusters for users i and j. Therefore, the coefficient for the gain of trust by user i in user j is defined as

$$\varphi_{ij} = \ln\left(1 + \frac{Num(i,j)}{C}\right), \tag{39}$$

where $Num(i,j)$ is the number of clusters in common for users i and j.

6.1.2. Aggregating trust values

In a distributed mobile environment, more than one trust group can be considered for an unknown user in many cases. Therefore, we need to aggregate normalised local trust values. Assume that users i and j simultaneously belong to n fuzzy implicit social graphs. The trust value between users i and j can then be aggregated as n local trust values $LOCTrust(i,j)_1, LOCTrust(i,j)_2, \cdots, LOCTrust(i,j)_n$ as follows:

$$AGTrust(i,j) = LOCTrust(i,j)_1 \oplus \cdots \oplus LOCTrust(i,j)_n \cdot (1 + \varphi_{ij}),$$

where $\oplus$ is a logic additive operator that can be defined as the addition of two trust values.

6.2. Trust transitivity

Users cannot always directly obtain trust recommendations for strangers from trusted neighbours in MSN environments. Hence, we need to propagate local trust values for indirect neighbour nodes.

6.2.1. Decay of trust

Trust has a decay property involving a loss of trust because of negative deviation between the results obtained and the intended target. Thus, there is a risk value when user i tries to trust another user j. As the number of transitive hops increases, the trust decay also increases. We assume that the trust value decays exponentially with the number of hops between users i and j. Therefore, the coefficient for the decay of trust of user i in user j is defined as

$$\varsigma_{i,j} = \xi - \frac{\omega}{e^{Hop(i,j)}}, \tag{40}$$

where ξ is the risk parameter for social networks and $0 \leqslant \omega \leqslant 1$ is a constant environment parameter that depends on the case studied; we set ω to $1/2$ here. $Hop(i,j)$ is the number of transitivity hops between users i and j.

6.2.2. Trust transitivity computation

Trust transitivity is a useful means whereby each user can gain a view of the network that is wider than his/her own experience. However, the trust values stored by user i still reflect only the experiences of user i and his/her acquaintances. To obtain a wider view, user i may wish to ask the friends of his/her friends. Continuing in this manner can give a user a complete view of the network. Assume that local trust values are propagated through n hops from the source to the target via users $[user_0(\text{source}), user_1, user_2, \cdots, user_n(\text{target})]$ and that the trust value held by $user_i$ for $user_{i+1}$ is $LOCTrust(i, i+1)$. We can then compute the transitive trust from $user_0$ to $user_n$ through n hops as

$$TRTrust(i,j) = (LOCTrust(0,1) \otimes \cdots \otimes LOCTrust(n-1,n)) \cdot (1 - \varsigma_{i,j}), \tag{41}$$

where $\otimes$ is a logic multiplicative operator, and can be a *multiplicative* or *min* operator.

7. Performance evaluation

7.1. Data set and experimental set-up

We evaluated the performance of κ-*FuzzyTrust* in experiments using a real MSN data set collected during the Reality Mining Project by the MIT Media Laboratory (http://realitycommons.media.mit.edu). The data came from 100 Nokia 6600 phones programmed to automatically run the ContextLog application as a background process at all times. Of the 100 users, 75 were students or staff of the MIT Media Laboratory; the remaining 25 were incoming students at the MIT Sloan Business School adjacent to the laboratory. The information collected includes call logs, Bluetooth devices in proximity, cell tower IDs, application usage, and phone status (e.g., charging or idle), and comes primarily from the Context application. The study generated data over approximately 4500 h on continuous behaviour in terms of user location, communication, and device usage. Suppose users are grouped into $n = 1, 2, 3, 4, 5$ (5 for default) clusters and the corresponding weight of cluster i is w_i, where $i = 1, 2, \cdots, 5$ and $\sum_{i=1}^{5} w_i = 1$. The users within a cluster are connected densely, and the relationships between interclusters are sparse relatively. The data was stored on each phone’s internal 32 MB memory card. The cards can store approximately four months of behavioural data before they need to be collected by the researchers. In fact, most of the vertices are always more similar to their direct neighbours than their indirect neighbours. In our experiments, hence, let $d = 1$, and $Neigh_1(i)$ corresponds to the direct neighbourhood of i.

We can easily get a social network of trust relationships from contact information, however, cluster is difficult to obtain. We make an assumption that user with continuous cell tower ID are in the same cluster. Then, we process the data set to get cluster information for each user: (1) Sort the call records, with cell tower ID in ascending order. There is a total of 2,642,367 rows, and we choose a smaller data set (the first 80,000). (2) There are a total of 1450 un-duplicated cell tower ID after the filter operation. We suppose that these cell tower ID are involved with $n = 1, 2, 3, 4, 5$ (5 for default); hence, about 300 cell tower ID are taken into a cluster for default. In this way, each cell tower ID is related with a cluster. (3) We make a program to collect cluster information of all of 100 user IDs. (4) Finally, we make another program to filter out the user IDs that are distrust (distrust is beyond the scope of this paper), and the remained are the edges and nodes (users) of our experiments. To measure the performance of our approach, we consider two metrics: effectiveness and trust accuracy. Effectiveness is the ability of*κ-FuzzyTrust* to infer the trust value between two users who are directly or indirectly connected in MSNs with overlapping communication communities. Trust accuracy represents the ability to predict if a user will be trusted or not. The simulation parameters and their default values are listed in Table 2.

7.2. Experimental results and analysis

7.2.1. Static trust

To measure the static trust between two users in an MSN, we used the number of neighbours to measure prestige trust. The more neighbours a user has, the greater is his/her prestige trust. User profile information was used to measure profile trust according to Eq. (18). We then calculates static trust values for users under $\kappa = 0, C = 1$ and $\kappa = 0.25, C = 3$. Because of space limitations, Fig. 4 only shows $StaticTrust^{\kappa}(i,j)$ results for users #1 and ID #2 under $\kappa = 0.25$ and $C = 3$. It is evident that most users have similar static trust trends for users #1 and #2. However, most users have quite high static trust in user #2, but very low static trust in user #1, apart from a few users. This indicates that user #2 and most of the other users are in the same cluster, whereas user #1 is a relatively isolated user.

To evaluate the effect of MSN scale on static trust, we analysed the relationship between static trust and the number of users. We first selected 10 users and their communication records to compute the profile trust $PTrust^{\kappa}(i,j)$ and prestige trust $BTrust^{\kappa}(i)$ under $\kappa = 0, C = 1$ and $\kappa = 0.25, C = 3$. $PTrust^{\kappa}(i, \#1)$ was obtained by computing the profile trust between user #1 and the other users. We then increased the number of users in steps of 20 up to 90 and recalculated the trust values. Table 3 lists the results for an MSN with one cluster. It is evident that $PTrust^{\kappa}(i, \#1)$ $(i = 1, 2, \ldots 10)$ does not change with the user count, but $BTrust^{\kappa}(i))$ depends on the total number of users.

We analysed the influence of cluster count on static trust. We assumed that all users were in a cluster with $\kappa = 0.25$ at the start, and then set the number of cluster to $2, 3, 4$, and 5 in turn. We computed the prestige trust values for the 10 users and

Table 2
Numerical values for the main parameters.

Parameter	Value
λ	1
ρ	0.6
ξ	0.2
ω	1/2
α	0.4–0.6

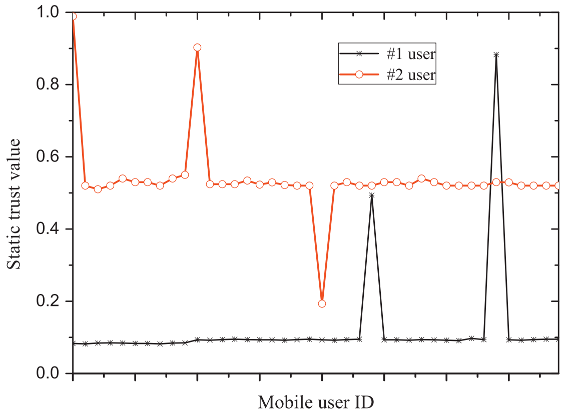


Fig. 4. Static trust results for users #1 and user #2 under $\kappa = 0.25$ and $C = 3$.

Table 3
Static trust values for 10 users for various user counts under $\kappa = 0.25, C = 3$.

User ID		#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
10 users	$BTrust^{\kappa}(i)$	0.3	0.1	0.6	0.6	0.2	0.7	0.8	0.4	0	0
	$PTrust^{\kappa}(i,\#1)$	1	0.2	0.6	0	0.45	0.32	0.7	0.5	0.3	0
30 users	$BTrust^{\kappa}(i)$	0.1333	0.1	0.2333	0.2667	0.2	0.3	0.2667	0.1667	0.1	0.1667
	$PTrust^{\kappa}(i,\#1)$	1	0.2	0.6	0	0.45	0.32	0.7	0.5	0.3	0
50 users	$BTrust^{\kappa}(i)$	0.14	0.14	0.16	0.18	0.14	0.18	0.18	0.2	0.12	0.14
	$PTrust^{\kappa}(i,\#1)$	1	0.2	0.6	0	0.45	0.32	0.7	0.5	0.3	0
70 users	$BTrust^{\kappa}(i)$	0.1571	0.1857	0.1429	0.1714	0.1571	0.1857	0.1286	0.1571	0.1286	0.1857
	$PTrust^{\kappa}(i,\#1)$	1	0.2	0.6	0	0.45	0.32	0.7	0.5	0.3	0
90 users	$BTrust^{\kappa}(i)$	0.1889	0.1667	0.1111	0.1667	0.1444	0.1889	0.1556	0.1222	0.1333	0.1667
	$PTrust^{\kappa}(i,\#1)$	1	0.2	0.6	0	0.45	0.32	0.7	0.5	0.3	0

the profile trust between user #1 and the other nine users. The total user count was 60 in the experiment. The results are shown in Table 4. It is evident that the value of $BTrust^{\kappa}(i)$ decreases as the cluster count increases. There are some zero values for $PTrust^{\kappa}(i,\#1)$, which implies that user i and user "#1" may not be in the same cluster when the cluster count increases, even if they belonged to the same cluster at the start.

7.2.2. Dynamic trust

7.2.2.1. Interaction familiarity trust. To measure the interaction familiarity between two users in an MSN, we use the number of interactions (calls) between them. The more calls they make, the more familiar they are. Fig. 5 shows the results for interaction familiarity trust under $\kappa = 0.25$.

Table 4
Static trust values for 10 users for different cluster counts with $\kappa = 0.25$.

User ID		#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
1 cluster	$BTrust^{\kappa}(i)$	0.1333	0.1167	0.15	0.1833	0.15	0.15	0.15	0.1667	0.1167	0.1857
	$PTrust^{\kappa}(i,\#1)$	1	0.2	0.6	0	0.45	0.32	0.7	0.5	0.3	0
2 clusters	$BTrust^{\kappa}(i)$	0.2344	1	0.1854	0.18	0.3252	0.1167	0.2418	0.2	0.32	0.2383
	$PTrust^{\kappa}(i,\#1)$	1	0	0.6	0	0.45	0	0.7	0	0.3	0
3 clusters	$BTrust^{\kappa}(i)$	0.423	0.5435	0.3436	0.2432	1	0.1167	1	0.1234	0.452	0.1353
	$PTrust^{\kappa}(i,\#1)$	1	0	0.6	0	0	0	0.7	0	0.3	0
4 clusters	$BTrust^{\kappa}(i)$	0.3215	0.6325	1	1	1	0.2021	1	0.1344	0.2203	0.3235
	$PTrust^{\kappa}(i,\#1)$	1	0	0.6	0	0	0	0.7	0	0.3	0
5 clusters	$BTrust^{\kappa}(i)$	0.4244	0.4327	0.3241	1	1	0.1133	1	0.1677	0.3209	1
	$PTrust^{\kappa}(i,\#1)$	1	0	0	0	0	0	0.7	0	0.3	0

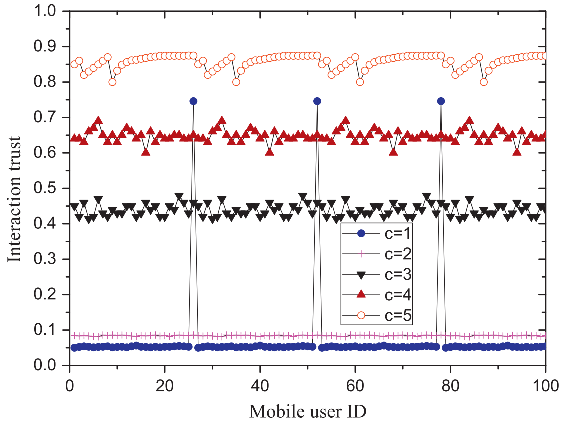


Fig. 5. Interaction familiarity trust for $\kappa = 0.25$ for user #1.

7.2.2.2. Evolution of interaction trust. To measure the trust dynamic characteristic to update the trust of user i in neighbouring user j, we need to incorporate feedback for following a particular recommendation into the trust relationship. User i who has acted on a rating given by neighbouring user j updates the value of trust in this neighbour according to his/her experience. Assume that user i has a trust value of $tr(i,j)_t$ in user j at time t and that $tr(i,j)_t = Ex(i,j)_t$. At time $t+1$, user i rates the trust value of user j as $Ex(i,j)_{t+1}$ via the trust model. Then we define the evolution ratio as

$$\eta = \frac{\beta Ex(i,j)_t + (1-\beta)Ex(i,j)_{t+1}}{Ex(i,j)_t + Ex(i,j)_{t+1}}, \tag{42}$$

where $\beta \in [0,1]$. In the simulations, we let $\beta = 0.65, \kappa = 0.25$, and $C = 3$. Fig. 6 shows the updating principle for η as described by Eq. (42). It is evident that trust values between two users with the same profile (i.e., $PTrust^{\kappa=0.25}(i,j) = 1$) evolve to 1. By contrast, trust values between two users with opposite profiles (i.e., $PTrust^{\kappa=0.25} = 0$) evolve to 0. When $PTrust^{\kappa=0.25} = 0.8$, when a user recommends a neighbour who is negatively rated, trust decreases quickly and recovers slowly. We varied $PTrust^{\kappa=0.25}$ in the interval $[0.5, 1)$ and obtained similar results. The main reason is that $\beta > 0.5$ leads to a slow increase and a fast decrease according to Eq. (42).

7.2.2.3. Location identity trust. We first compute the distribution of interaction locations according to the data set. Then we can calculate the location identity trust results for $LTrust^{\kappa}(i,j)$. The location-related frequency of interactions between user #1 and nine other users under $\kappa = 0.25$ is shown in Table 5. It is evident that user #1 most frequently communicates with other mobile users when in the office. We calculated the trust values between these 10 users according to Eq. (22) and the results are presented in Table 6. The maximum trust values exhibit a scattered distribution. The results indicates that the mobile users have the highest trust in initial user i, and i often has social interactions with the users in the same higher-weighted location, such as at home and in an apartment. Therefore, location identity has a great impact on trust between user i and other users. Since we ignore the direction of edges in MSNs, the results shown in Table 6 are symmetric.

7.2.2.4. Time identity trust. Table 7 shows the interaction distribution by time slot and time identity between user #1 and nine other users. User #1 often has the most frequent communication with other mobile users in public time, but rarely communicates with users #2, #5, #6, #7, #9, and #10 in private time. We can calculate trust values between users i and j using Eq. (24). The results for time identity trust between the 10 users under $\kappa = 0.25$ are shown in Table 8. The results are symmetric because we ignore the direction of communication. It is evident that most mobile users hold similar trust values for a given user in terms of time identity. This phenomenon is caused by the communication duration.

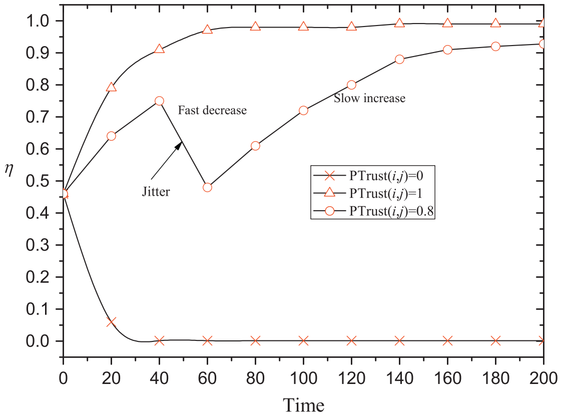


Fig. 6. Evolution of interaction trust for $\kappa = 0.25$ and $C = 3$.

Table 5
Location-related frequency of interactions between user #1 and nine other users under $\kappa = 0.25$.

User ID	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Home	/	0	0	5	0	0	0	1	0	0
Apartment	/	20	34	0	0	0	0	0	0	0
Office	/	12	2	211	0	0	16	0	0	88
Others	/	0	10	5	0	0	20	20	0	21

Table 6
Location identity trust results for 10 users under $\kappa = 0.25$.

User ID	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
#1	/	0.8	0.6	1	0	0	0.6	0.4	1	0.6
#2	0.8	/	0.8	0.4	1	0.6	0.4	1	0.6	0.6
#3	0.6	0.8	/	0.6	0.6	0.6	0.6	0.4	1	0
#4	1	0.4	0.6	/	0	0.6	1	0	0.8	0
#5	0	1	0.6	0	/	1	0.8	0.4	0	0.6
#6	0	0.6	0.6	0.6	1	/	0.4	0.8	0.8	0
#7	0.6	0.4	0.6	1	0.8	0.4	/	0.6	0.4	0.6
#8	0.4	1	0.4	0	0.4	0.8	0.6	/	0.8	0.6
#9	1	0.6	1	0.8	0	0.8	0.4	0.8	/	0
#10	0.6	0.6	0	0	0.6	0	0.6	0.6	0	/

Table 7
Interaction frequency over time for user #1 and nine other users under $\kappa = 0.25$.

User ID	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Private time	/	0	3	15	0	0	0	1	0	0
Public time	/	32	43	206	0	0	36	20	0	109

Table 8
Time identity trust results for 10 users under $\kappa = 0.25$.

User ID	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
#1	/	0.0641	0.4412	0.5343	0.0014	0.0064	0.0232	0.0423	0.24	0.0424
#2	0.0641	/	0.0364	0.0022	0.3214	0.0042	0.0164	0.003	0.214	0.332
#3	0.4412	0.0364	/	0.0201	0.4204	0.0046	0.0524	0.2132	0.0364	0.0042
#4	0.5343	0.0022	0.0201	/	0.0037	0.3232	0.0964	0.0192	0.1464	0.32
#5	0.0014	0.3214	0.4204	0.0037	/	0.0332	0.0164	0.3256	0.4412	0.0232
#6	0.0064	0.0042	0.0046	0.3232	0.0332	/	0.0854	0.1292	0.0054	0.0008
#7	0.0232	0.0164	0.0524	0.0964	0.0164	0.0854	/	0.0632	0.1064	0.0576
#8	0.0423	0.003	0.2132	0.0192	0.3256	0.1292	0.0632	/	0.2353	0.1032
#9	0.24	0.214	0.0364	0.1464	0.4412	0.0054	0.1064	0.2353	/	0.0552
#10	0.0424	0.332	0.0042	0.32	0.0232	0.0008	0.0576	0.1032	0.0552	/

7.2.3. Trust aggregation and transitivity results

7.2.3.1. Trust aggregation inference. To quantitatively measure the performance of our trust model in comparison to the profile-based trust approach [56] (abbreviated as P-model) and the random approach (R-model), we use the *F*-measure for the accuracy using recall and precision jointly [21,50]. The *F*-measure (expressed as $F(i,j)$) is the harmonic mean for precision (denoted by $P(i,j)$), which means the number of users correctly labelled as belonging to the fuzzy implicit social graph G_j, and recall (denoted as $R(i,j)$), which is given by dividing the number of users belonging to G_j by the total number of users really belonging to G_j. Hence, we can obtain $F(i,j)$ by redefining the accuracy metrics in [50], as shown in Table 9, which includes precision, recall, and the *F*-measure.

To evaluate the effectiveness of trust transitivity inference using κ-FuzzyTrust, we set the results obtained from the proposed trust model as the baseline and applied the $\oplus$ operation for addition of two trust values. We evaluated the proposed inference mechanism for trust aggregation. Figs. 7 and 8 show the precision of trust transitivity inference within one hop under $\oplus$. It is evident that the accuracy is high, with a minimum $F(i,j)$ of 0.598 under $\kappa = 0.25$ for a cluster count of one; when the cluster count is five, the accuracy is 0.87. This indicates that κ-FuzzyTrust performs well inferring trust. The accuracy of κ-FuzzyTrust is also higher than that of P-model and R-model in most cases. Fig. 8 shows that the mean error

Table 9
Accuracy metrics.

Metric	Definition
$P(i,j)$	$\frac{\mid Dt_{ij} \cap Et_{ij} \mid}{\mid Et_{ij} \mid}$
$R(i,j)$	$\frac{\mid Dt_{ij} \cap Et_{ij} \mid}{\mid Dt_{ij} \mid}$
$F(i,j)$	$2 \cdot \frac{P(i,j) \cdot R(i,j)}{P(i,j) + R(i,j)}$
$Dt_{ij} = \{e_{ij} : i$ trusts j directly$\}$	
$Et_{ij} = \{e_{ij} : i$ trusts j through trust model$\}$	

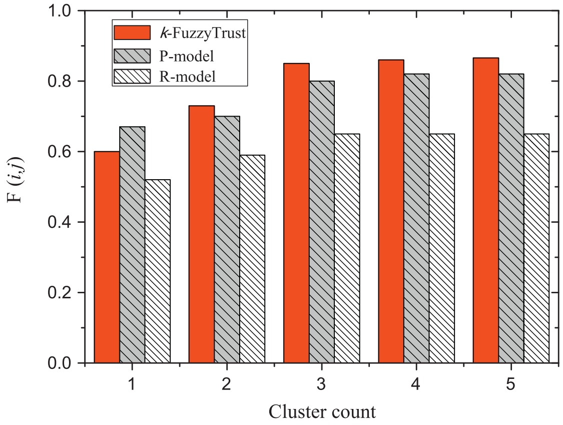


Fig. 7. $F(i,j)$ for various cluster counts.

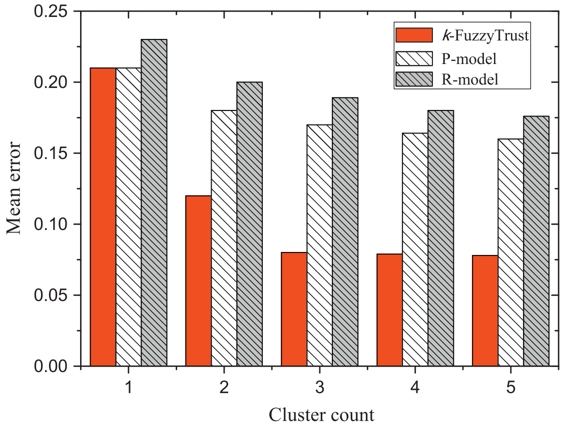


Fig. 8. Mean error for various cluster counts.

performance is very similar for κ-FuzzyTrust and P-model. As the number of cluster counts increases, the performance of κ-FuzzyTrust is much better than that of P-model and R-model. The results confirms that construction of a fuzzy implicit social graph using fuzzy relations between overlapping users, as well as exploring the stable and objective information for inferring trust, can improve the precision of trust inference.

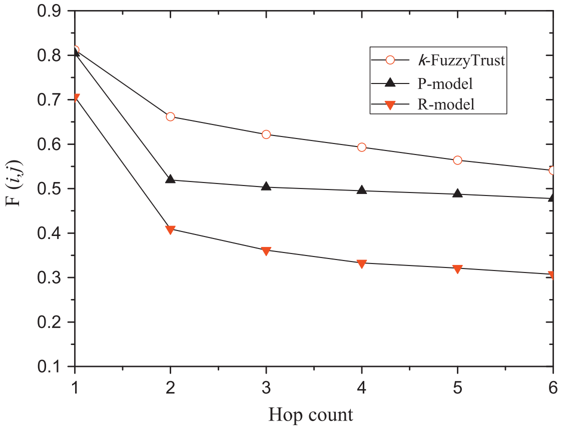


Fig. 9. $F(i,j)$ for various cluster counts.

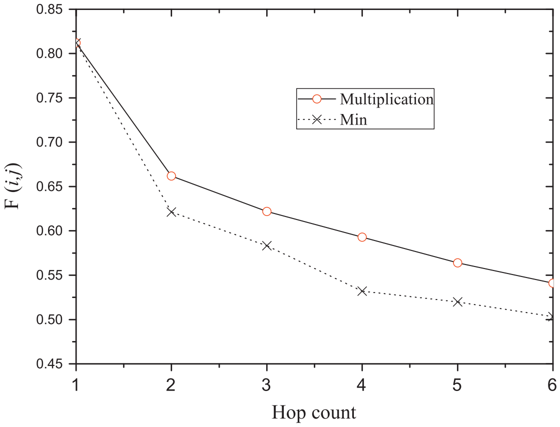


Fig. 10. $F(i,j)$ for various operations.

7.2.3.2. Trust transitivity inference. Fig. 9 shows the precision of trust transitivity inference with different hop counts for P-model, R-model and κ-FuzzyTrust. It is clear that the accuracy of κ-FuzzyTrust is higher than that of the P-model and R-model for different hop counts. Fig. 10 shows the precision of the trust transitivity inference for different hop counts under two $\otimes$ operations: *Multiplication* and *Min*. The *Min* operation [9] is used to return the minimal trust value along the transitivity path as the inference trust value. According to small world theory [33], anyone can connect with and recognise any other stranger within six hops. Hence, we should calculate the precision of trust between two users within six hops. The results reveal that the precision of trust transitivity inference decreases with increasing hop count in both cases. However, the precision of trust transitivity inference is higher for *Multiplication* than for *Min* under various hop counts. Furthermore, Fig. 9 shows that the proposed inference mechanism achieves higher precision for trust transitivity if a high weight is assigned to the basic trust between two mobile users. In addition, the *Multiplication* operation significantly outperforms *Min*, as shown in Fig. 10.

8. Conclusion

We presented a new approach for inferring trust values for large-scale MSNs using a fuzzy implicit social graph to enhance the understanding of trust between mobile users. We first detected fuzzy cluster structures using the *FuzzyDetecting* algorithm for fuzzy degree κ. We then analysed critical MSN properties and constructed a mobile trust context based on two aspects: static attributes and dynamic behaviour patterns, such as user profile and prestige, interaction familiarity and evolution, and location and time identities. We proposed the *κ-FuzzyTrust* inference mechanism for users in large-scale MSNs and discussed the propagation and aggregation of local trust values. Experimental results for a real mobile data set demonstrate that *κ-FuzzyTrust* infers trust well in practice. In particular, the precision of trust transitivity inference for the *Multiplication* operation is higher than that for the *Min* operation under various hop counts. Simulations reveal that our trust model performs better than the conventional R-Model and similar trust models such as P-Model. However, we mainly consider the static attributes of users to calculate the identity of groups, which results in some limitations. Our future work will include the design of a new integrated and comprehensive algorithm for predicting trust and an adaptive approach for membership functions to better characterise the trust semantics between mobile users. We will also study more stable and objective information for different MSN types in detail. Finally, we plan to investigate a systematic technology to create a mobile social context that takes into account both the static attributes and dynamic characteristics in real systems.

Acknowledgments

This research was supported in part by Grant Nos. 61272151 and 61472451, ISTCP Grant 2013DFB10070, the China Hunan Provincial Science Technology Program (Grant 2012GK4106), the Ministry of Education Fund for Doctoral Disciplines in Higher Education (Grant 20110162110043), Shanghai Jiao Tong University 211/985 Project of China (Project No. WF220103001), and the Specialised Research Fund for Doctoral Program in Higher Education of Hunan Provincial Education Department (Grant CX2010B075), and Scientific Research Fund of Hunan Provincial Education Department (Grant 14C0286). We are grateful to the Editor in Chief and the anonymous reviewers for insightful comments that helped to improve this paper.

References

[1] A. Ankolekar, G. Szabo, Y. Luon, B.A. Huberman, D. Wilkinson, F. Wu, Friendlee: a mobile application for your social life, in: Proceedings of the 11th International Conference on Human–Computer Interaction with Mobile Devices and Services, 2009, (article no. 27), doi: 10.1145/1613858.1613893.

[2] V.A. Balasubramaniyan, M. Ahamad, H. Park, CallRank: combating SPIT using call duration, social networks and global reputation, in: Proceedings of the 4th Conference on Email and Anti-Spam, 2007, pp. 18–24.

[3] J.C. Bezdek, J.M. Keller, R. Krishnapuram, N.R. Pal, Fuzzy models and algorithms for pattern recognition and image processing, in: Didier Dubois, Henri Prade (Eds.), The handbooks of fuzzy sets series, Springer Science + Business Media, Inc., 233 Spring Street, New York, NY 10013, USA, 1999.

[4] J.C. Bezdek, B. Spillman, R. Spillman, A fuzzy relation space for group decision theory, Fuzzy Sets Syst. 1 (1978) 255–268.

[5] J.M. Blin, Fuzzy relations in group decision theory, J. Cybernet. 4 (1974) 17–22.

[6] G.H. Canepa, D. Lee, A virtual cloud computing provider for mobile devices, in: Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services, 2010, (article no. 6), doi: 10.1145/1810931.1810937.

[7] P. Carrington, J. Scott, S. Wasserman, Models and Methods in Social Network Analysis, Cambridge University Press, Cambridge, 2005.

[8] K. Chard, K. Bubendorfer, S. Caton, O.F. Rana, Social cloud computing: a vision for socially motivated resource sharing, IEEE Trans. Serv. Comput. 5 (2012) 551–563.

[9] Y. Chen, T. Bu, M. Zhang, H. Zhu, Measurement of trust transitivity in trustworthy networks, J. Emerging Technol. Web Intell. 2 (2010) 319–325.

[10] S. Chen, G. Wang, W. Jia, Cluster-group based trusted computing for mobile social networks using implicit social behavioral graph, Future Gener. Comput. Syst. (2014), http://dx.doi.org/10.1016/j.future.2014.06.005 (Date of Publication [Online]: 12 June).

[11] C. Costa, K. Bijlsma-Frankema, Trust and control interrelations, Group Organ. Manage. 32 (2007) 392–406.

[12] S. Counts, Group-based mobile messaging in support of the social side of leisure, Comput. Support. Coop. Work 16 (2007) 75–97.

[13] D. Damopoulos, G. Kambourakis, M. Anagnostopoulos, S. Gritzalis, J.H. Park, User privacy and modern mobile services: are they on the same path?, Pers Ubiq. Comput. 17 (2013) 1437–1448.

[14] N. Eagle, A. Pentland, D. Lazer, Inferring friendship network structure using mobile phone data, Proc. Natl. Acad. Sci. USA 106 (2009) 15274–15278.

[15] B. Elisa, C. James, F. Elena, Identity, privacy, and deception in social networks, IEEE Internet Comput. 18 (2014) 7–9.

[16] S. Farnham, P. Keyani, Swarm: hyper awareness, micro coordination, and smart convergence through mobile group text messaging, in: Proceedings of the 39th Annual Hawaii International Conference on System Sciences, vol. 3, 2006, p. 59a.

[17] D. Gambetta, Can We Trust Trust? Trust: Making and Breaking Cooperative Relations, Basil Blackwell, 1988.

[18] M. Girvan, M.E.J. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. USA 99 (2002) 7821–7826.

[19] J. Golbeck, J. Hendler, Film trust: movie recommendations using trust in Web-based social network, in: Proceedings of the 3rd IEEE Conference on Consumer Communications and Networking, 2006, pp. 1314–1315.

[20] S. Gregory, An algorithm to find overlapping community structure in networks, in: Proceedings of PKDD, 2007, pp. 91–102.

[21] R. Grob, M. Kuhn, R. Wattenhofer, M. Wirz, Cluestr: mobile social networking for enhanced group communication, in: Proceedings of GROUP'09, 2009, pp. 81–90.

[22] T. Grandison, M. Sloman, A survey of trust in Internet applications, IEEE Commun. Surv. Tutor. Q4 (2000) 2–16.

[23] F. Hao, G. Min, M. Lin, C. Luo, L.T. Yang, MobiFuzzyTrust: an efficient fuzzy trust inference mechanism in mobile social networks, IEEE Trans. Parallel Distrib. Syst. (2013) 1–11. PP(99).

[24] F. Hao, S.S. Yau, G. Min, L.T. Yang, Detecting *k*-balanced trusted cliques in signed social networks, IEEE Internet Comput. 18 (2014) 24–31.

[25] L. Humphreys, Mobile social networks and social practice: a case study of dodgeball, J. Comput. Mediated Commun. 13 (2007) 341–360.

[26] W. Jiang, G. Wang, J. Wu, Generating trusted graphs for trust evaluation in online social networks, Future Gener. Comput. Syst. 31 (2014) 48–58.

[27] G. Kambourakis, Anonymity and closely related terms in the cyberspace: an analysis by example, J. Inf. Secur. Appl. 19 (2014) 2–17.

[28] S. Kim, S. Han, The method of inferring trust in Web-based social network using fuzzy logic, in: Proceedings of the 2009 International Workshop on Machine Intelligence Research, 2009, pp. 140–144.

[29] M. Kim, S. Park, Group affinity based social trust model for an intelligent movie recommender system, Multimed. Tools Appl. 64 (2013) 505–516.

[30] Y.A. Kim, R. Phalak, A trust prediction framework in rating-based experience sharing social networks without a Web of trust, Inf. Sci. 191 (2012) 128–145.

[31] M. Kim, J. Seo, S. Noh, S. Han, Identity management-based social trust model for mediating information sharing and privacy enhancement, Secur. Commun. Netw. 5 (2012) 887–897.

[32] R. Kikin-Gil, Affective is effective: how information appliances can mediate relationships within communities and increase one's social effectiveness, Pers. Ubiq. Comput. 10 (2006) 77–83.

[33] J. Kleinberg, The small-world phenomenon: an algorithmic perspective, in: Proceedings of the 32nd ACM Symposium on Theory of Computing, 2000, pp. 163–170.

[34] E. Kuada, H. Olesen, A social network approach to provisioning and management of cloud computing services for enterprises, in: Proceedings of Cloud Computing 2011, pp. 98–104.

[35] Z. Li, H. Shen, K. Sapra, Leveraging social networks to combat collusion in reputation systems for peer-to-peer networks, IEEE Trans. Comput. 62 (2013) 1745–1759.

[36] J. Li, Z. Zhang, W. Zhang, Mobitrust: trust management system in mobile social computing, in: Proceedings of the IEEE 10th International Conference on Computer and Information Technology, 2010, pp. 954–959.

[37] W. Liu, L. Chen, Community detection in disease-gene network based on principal component analysis, Tsinghua Sci. Technol. 18 (2013) 454–461.

[38] M. Lund, B. Solhaug, K. Stolen, Evolution in relation to risk and trust management, Computer 43 (5) (2010) 49–55.

[39] A. Mohaisen, N. Hopper, Y. Kim, Keep your friends close: incorporating trust into social network-based Sybil defenses, in: Proceedings of IEEE INFOCOM 2011, pp. 1943–1951.

[40] F. Musau, G. Wang, S. Yu, Muhammad B. Abdullahi, Securing recommendations in grouped P2P e-commerce trust model, IEEE Trans. Netw. Serv. Manage. 9 (2012) 407–420.

[41] K. Musial, P. Kazienko, P. Brodka, User position measures in social network, in: Proceedings of SNA-KDD, 2009, doi: 10.1145/1731011.1731017, (article no 6).

[42] H. Orman, Did you want privacy with that? Personal data protection in mobile devices, IEEE Internet Comput. 17 (2013) 83–86.

[43] A. Oulasvirta, M. Raento, S. Tiitta, ContextContacts: redesigning SmartPhone's contact book to support mobile awareness and collaboration, in: Proceedings of Mobile HCI, 2005, pp. 167–174.

[44] R. Pezzi, Information Technology Tools for a Transition Economy, 2009. <http://www.socialcloud.net/papers/ITtools.pdf>.

[45] J.M. Pujol, J. Béjar, J. Delgado, Clustering algorithm for determining community structure in large networks, Phys. Rev. E 74 (2006) 016107.

[46] X. Qiao, X. Li, Z. Su, D. Cao, A context-awareness dynamic friend recommendation approach for mobile social network users, Int. J. Adv. Intell. 3 (2011) 155–172.

[47] J. Reichardt, S. Bornholdt, Detecting fuzzy community structures in complex networks with a Potts model, Phys. Rev. Lett. 93 (2004) 218701.

[48] T.J. Ross, Fuzzy Logic with Engineering Application, third ed., John Wiley & Sons, New York, 2005.

[49] M. Roth, A. Ben-David, D. Deutscher, G. Flysher, I. Horn, A. Leichtberg, N. Leiser, Y. Matias, R. Merom, Suggesting friends using the implicit social graph, in: Proceedings of KDD'10, 2010, pp. 233–241.

[50] S. Shekarpour, S.D. Katebi, Modeling and evaluation of trust with an extension in semantic web, Web Semantics 8 (2010) 26–36.

[51] W. Sherchan, S. Nepal, C. Paris, A survey of trust in social networks, ACM Comput. Surv. 45 (2013) (article no. 47).

[52] B. Spillman, J. Bezdek, R. Spillman, Development of an instrument for the dynamic measurement of consensus, Commun. Monogr. 46 (1979) 1–12.

[53] P. Sun, L. Gao, S. Han, Identification of overlapping and non-overlapping community structure by fuzzy clustering in complex networks, Inf. Sci. 181 (2011) 1060–1071.

[54] H. Takabi, Security and privacy challenges in cloud computing environments, IEEE Secur. Priv. 8 (2010) 24–31.

[55] D. Verma, M. Meila ̆, A comparison of spectral clustering algorithms, Technical Report 03-05-01, Department of Computer Science Engineering, University of Washington, 2003.

[56] F.E. Walter, S. Battiston, F. Schweitzer, A model of a trust-based recommendation system on a social network, Auton. Agent Multi-Agent Syst. 16 (2008) 57–74.

[57] S. Wasserman, K. Faust, D. Iacobucci, Social Network Analysis: Methods and Applications, Cambridge University Press, Cambridge, The Edinburgh Building, Cambridge CB2 2RU, UK, 1994 (ISDN: 0-521-38707-8).

[58] G. Wang, W. Jiang, J. Wu, Z. Xiong, Fine-grained feature-based social influence evaluation in online social networks, IEEE Trans. Parallel Distrib. Syst. 25 (9) (2014) 2286–2296.

[59] G. Wang, F. Musau, S. Guo, M. Bashir Abdullahi, Neighbor similarity trust against Sybil attack in P2P E-Commerce, IEEE Trans. Parallel Distrib. Syst. (2014), http://dx.doi.org/10.1109/TPDS.2014.2312932 (Date of Publication [Online]: 20 March).

[60] G. Wang, J. Wu, FlowTrust: trust inference with network flows, Frontiers Comput. Sci. China 5 (2) (2011) 181–194.

[61] G. Wang, J. Wu, Multi-dimensional evidence-based trust management with multi-trusted paths, Future Gener. Comput. Syst. 27 (5) (2011) 529–538.

[62] L. Wei, H. Zhu, Z. Cao, X. Dong, W. Jia, Y. Chen, A.V. Vasilakos, Security and privacy for storage and computation in cloud computing, Inf. Sci. 258 (2014) 371–386.

[63] S. White, P. Smyth, A spectral clustering approach to finding communities in graphs, in: Proceedings of the 5th SIAM International Conference on Data Mining, 2005, p. 274–285.

[64] L.A. Zadeh, Similarity relations and fuzzy orderings, Inf. Sci. 3 (1971) 177–200.

[65] S. Zhang, R.S. Wang, X.S. Zhang, Identification of overlapping community structure in complex networks using fuzzy c-means clustering, Stat. Mech. Appl. 374 (2007) 483–490.