LASS: Local-Activity and Social-Similarity Based Data Forwarding in Mobile Social Networks

Zhong Li, Cheng Wang, Siqian Yang, Changjun Jiang, and Xiangyang Li, *Senior Member, IEEE*

Abstract—This paper aims to design efficient data forwarding schemes based on *social similarity* for mobile social networks (MSNs) without infrastructure support. Diverse definitions of social similarity, acting as the criteria of relay selection, produce various forwarding schemes; and the appropriateness and practicality of definitions determine the performance of the schemes indeed. A popular definition has recently been proven to be more efficient than other existing ones, *i.e.*, the more common interests between two nodes, the larger social similarity between them. In this work, we indicate that schemes based on such definition ignore the fact that members within the same community, *i.e.*, with the same interest, usually have different levels of internal activity, which possibly results in a low efficiency of data delivery in some situations. Thus, we design a new data forwarding scheme for MSNs based on community detection in dynamic weighted networks, called LASS, taking into account the difference of members' activity within each community, *i.e.*, *local activity*. To the best of our knowledge, the proposed scheme is the first one aware of different levels of local activity within communities. Through extensive simulations, LASS achieves better performance than state-of-the-art protocols.

Index Terms—local activity, social similarity, data forwarding, mobile social networks

1 Introduction

According to different underlying network architectures, MSNs (Mobile Social Networks) can be classified into two broad types, *i.e.*, *centralized MSNs* and *distributed MSNs*. The former, the traditional online social networking services (MySpace, Facebook *etc.*) migrate their centralized social applications or portal web sites to mobile devices. It is an extension of web-based social networks. Cellular networks provide the most popular network infrastructure to support centralized MSNs [1]. The latter, mobile users disseminate data in a decentralized way which is on the basis of opportunistic contacts. Thus, the ad-hoc networks are fit for the distributed MSNs, *e.g.* proximity-based applications using Bluetooth. The social construction and underlying infrastructure for above two MSN types are shown in Fig.1.

In the aspects of privacy protection, performance bottleneck and geo-location application, decentralized MSNs have a more effective underlying network architecture for mobile social networks. An important research trend of this area is to investigate how to achieve better data dissemination efficiency (deliver ration and throughput-delay tradeoff) by mining and utilizing the social relationship.

In mobile social networks, most data forwarding algorithms are "encounter-based". Recently, people have found social information has big impact on data forwarding. Thus, some "social-aware encounter-based" forwarding schemes are received enormous attention, see [2]–[9]. Hui *et al.* [3] contributed a milestone work BUBBLE RAP to the data forwarding scheme through creatively exploiting node local centrality and social community structures. Gao *et al.* [4] investigated the multicast with known community structures for delay tolerant networks. Nguyen *et al.* [6] proposed a new efficient scheme Nguyen's Routing that delivers data to nodes having more common communities to gain a high data delivery ratio. **In the literature, one common critical assumption has been widely utilized: two nodes can contact with a higher probability if they have more social similarity. The measurement of social similarity is various and one of the most popular methods is using common interests or common communities.**

In this paper, we show that the measurement of social similarity based on the number of common communities has an imperfection for some cases, *i.e.*, it ignores the fact that the members within the same community usually have different levels of local activity, which possibly results in a potentially low efficiency in terms of deliver ratio and latency due to the misalignment estimation of nodes' contact probability. **Here, node local activity is a statistics of encounter probability in a node's communities. More detailed explanation will be provided in Section 3.1.**

As depicted in Fig.2(a), assuming destination w has common interests both with node u and v; node u has more common interests with destination w than v, but the activity of u in common interests of u and w is weaker than that of v in common interest(s) of v and w. Therefrom, it is

- *Zhong Li, Siqian Yang and Changjun Jiang are with the Department of Computer Science, Tongji University, Shanghai, China.*
  E-mail: 007lizhong@gmail.com
- *Cheng Wang is with the School of Information Technology and Engineering, University of Ottawa, Canada.*
- *Xiangyang Li is with the Department of Computer Science, Illinois Institute of Technology, Chicago, IL, 60616.*

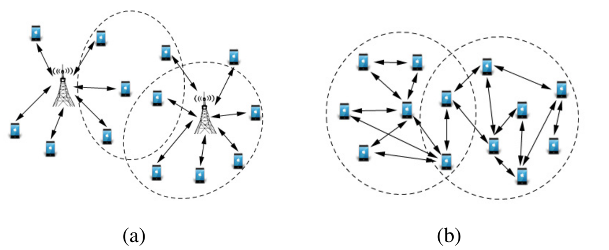


Fig. 1: The areas circled by the dotted line are the friendship communities. The picture (a) shows the centralized case and (b) presents the distributed case.

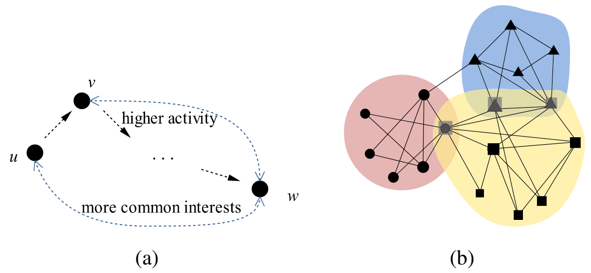


Fig. 2: Figure (a) depicts the transmitting choice for node u to destination w in data forwarding algorithm. In figure (b), different sizes of icons represent each node's different levels of local activity in its community. In the overlapping area, one node has many different activity attributes, corresponding to each belonging community respectively, depicted by overlapping icons of square and triangle or square and circle.

uncertain to determine whether v should be chosen as the next relay node due to lack of measurement criteria.

We design a new data forwarding scheme for MSNs based on social similarity in dynamic weighted networks, called LASS (Local-Activity and Social-Similarly), which has two novel characteristics compared with the existing community-based schemes as follows: (1) LASS takes into account the diversity of members' activity within each community based on weighted network models, *i.e.*, *local activity*, rather than making no difference to all members in terms of activity. (2) The global activity of a node can be defined by an *activity vector* whose entries are the local activity of this node for all current communities; then, we give a novel definition of the *social similarity* of two nodes by the *inner product* of their activity vectors, by which the forwarding scheme can operate according to a simple rule of choosing the relay node with higher social similarity to the destination. To the best of our knowledge, this proposed LASS scheme is the first one aware of the different levels of local activity within communities; furthermore, the inner product based definition may lead to new ideas in measuring the social similarity among nodes.

Here, the structure of social networks can be modeled more close to reality. As illustrated in Fig.2(b), the different interest groups emerge through the community detection algorithm and the different levels of local activity is held by users. The major contribution of this paper is two-folds:

- We propose a data forwarding scheme LASS by investigating the local activity for mobile social networks. The *local activity* is defined to describe different levels of node internal activity. It is a statistics of encounter probability in a node's communities. Then, according to the community detection results, *activity vector* is formed by using the local activity. Meanwhile, the *social similarity* is also computed by the vector inner product method for settling our problem. Comparing with the popular social forwarding algorithms, *e.g.*, BUBBLE RAP [3] and Nguyen's Routing [6], LASS has competitive effects, which can achieve 66.74 percent delivery ratio and only have 26.07 percent overhead ratio with controlled delay in a reasonable range.
- We give a self-adaptive weighted dynamic community detection algorithm (SAWD) so as to gain community structures and the node local activity for LASS. The *communication critical value* and the *weighted density embryo* are defined to help form communities. Both definitions assist us in avoiding some low weighted edges to shape communities. In the dynamic environment, we clearly distinguish and tackle two kinds of network changes: "out-pool" case and "in-pool" case. Comparing with AFOCS (a newly overlapped and dynamic community detection algorithm), SAWD can save some repeated operations on edges to avoid the unnecessary computing and is fit for the weighted graph.

The rest of the paper is organized as follows. Section 2 provides the basic network model. Section 3 gives the designing of LASS data forwarding scheme. In order to obtain the node local activity, Section 4 gives the community detection algorithm SAWD. In Section 5, we establish our experiment environment and gain some interesting results about social relationships and data forwarding performance. Section 6 states several issues about our algorithm's designing and gives some possible extensions for the future. In Section 7, we review some related studies. Finally, we conclude the paper in Section 8.

2 Network Model

2.1 Dynamic Weighted Graph

We model the mobile network, consisting of users in the mobile social network, as a *dynamic weighted graph* which can be defined as a time sequence of network graph, denoted by $\mathcal{G} = \{G_0, G_1, ..., G_t, ...\}$, where $G_t = (V_t, E_t, W_t, F_t)$ represents a time dependent *network snapshot* recorded at time t, V_t denotes the set of nodes, $E_t = \{(u,v)|u,v \in V_t\}$ denotes the edge set, $W_t = \{w_{ij} \in [0,1)|i,j \in V_t \text{ and } (i,j) \in E_t\}$ denotes the set of weights on edges at time t, and $F_t : E_t \to W_t$ is a mapping that assigns weights to edges. The node and edge sets change over time. For a node u, let d_u^t and $N_t(u)$ denote the degree and the set of all its neighbors at time t respectively. **Especially, in our study, the value of w_{ij} denotes an encounter probability in mobile ad-hoc networks. Extended explanations about w_{ij} will be provided in Section 5.2 and Section 6** .

2.2 Community Structure

A community is a structure that has a group of tight-knit nodes with more internal links than external links, [10]–[12]. If people have common interests or often encounter with each other, they may form a community. But, the definition of it is quiet subjective (most of the definitions are concerned with the special community detection algorithms or social applications), there is not a uniform definition about community. Here, let $\mathcal{C}_t = \{C_1, C_2, ..., C_k\}_t$ denote the *network community structure at time t*, *i.e.*, a collection of subsets of V_t, where the element $C_i \in \mathcal{C}_t$ and its induced subgraph form a community of G_t. Particularly, allow $C_i \bigcap C_j \neq \emptyset$, *i.e.*, the network communities can overlap with each other. For a node u, let $Com_t(u)$ denote the set of labels of all communities containing u at time t, *i.e.*, $\{C_l | l \in Com_t(u)\}$; let $\mathcal{C}_t(u)$ denote the set of all community structure containing u at time t. Some notations are listed in TABLE 1.

TABLE 1: Main notations used in this paper

Notation	Meaning
$\mathcal{C}_t$	the community structure at time t
$N_t(u)$	the set of neighbor labels of node u at time t
$Com_t(u)$	the set of community labels of node u at time t
$\mathcal{C}_t(u)$	the set of all communities containing node u at time t
x_t	the average communication level at time t
$O_t(u,v)$	the weighted density embryo generated by (u,v) at time t
$\Phi(O_t(u,v))$	the weighted density function
$\Gamma(C_i, C_j)$	coupling coefficient
α	the threshold of combining criterion
a_{ij}^t	the local activity value for node i in community j at time t
A_i^t	the activity vector of node i at time t
SS_{uw}^t	the social similarity between node u and w at time t

3 LASS Data Forwarding Scheme

In this section, we propose our LASS (Local-Activity and Social-Similarity) data forwarding scheme.

3.1 About Local Activity

Definition 1 (Local Activity): Denote the node local activity as a_{ij}^t at time t, where i denotes the label of a node and j denotes the label of its belonging community. Then,

$$a_{ij}^t = \begin{cases} \frac{\sum_{(i,k)\in C_j} w_{ik}}{\sum_{(k',k'')\in C_j} w_{k'k''}} & j \in Com_t(i) \\ 0 & \text{otherwise} \end{cases}$$

Definition 2 (Activity Vector): We define an activity vector $A_i^t = (a_{i1}^t, a_{i2}^t, ..., a_{ij}^t, ..., a_{ik}^t)$ for each mobile node i at time t, where a_{ij}^t denotes the local activity value of node i in community j. The value of k represents the number of communities after applying the following SAWD community detection algorithm.

An activity vector contains three-dimensional meanings: time, the number of communities, the local activity.

We give **a metaphor** to explain the meaning of the node local activity. Assuming there exists a basketball club (community) in a university. Two students A and B are belonged to this club. If A has many interactions with other members in the club, while B has few interactions with members, we will say, A has a high local activity and B has a low local activity. If there exists more than one community which A and B are belonged to, A and B will have different local activity in each community. **In data forwarding, local activity is important because if the message is given to a node having low local activity, it will bring about a low efficiency in terms of delivery ratio.**

There exist some other methods and concepts which need to be differentiated from our local activity.

In PROPHET [13], they simply use the encounter history to predict the future delivery probability. This algorithm is not as good as community-based prediction in social networks. Because in social networks, nodes belonging to the same community means they are more likely to meet each other. To some extent, it can reflect social preference. That is to say, community structure is good at delivery prediction. Therefore, using node local activity(associated with community structure) to design data forwarding scheme is better than PROPHET.

In Simbet [2] and BUBBLE RAP [3], they use betweenness centrality in data forwarding. Betweenness measures the extent to which a node lies on the shortest paths linking other nodes. A node with a high betweenness centrality has a capacity of facilitating interactions between the nodes that it links. However, not only global but also local centrality are just fit for unweighted graphs. In unweighted graphs, there will be an edge if there exists a contact between two nodes. **But in reality, the contact probability may be too low to be utilized in data forwarding, *i.e.*, betweenness centrality can not reflect the encounter probability.** To some extent, local centrality and local activity both can represent the importance of a node in its communities, but, they are not the same concept. **The former is only with node degrees, the latter is a statistics of encounter probability in a node's communities.**

We find different nodes in the same community have different local activity values and the same node in different communities has different local activity values. Next, we take advantage of the node local activity to develop the social similarity between two nodes.

3.2 About Social Similarity

There are many kinds social similarity measurements, such as cosine angular distance [14], Hamming feature distance [9], the number of common communities (interesting groups) [6]. However, the above distance-based methods cannot give a meaningful explanation in real social networks. The common interests-based method has a problem that if we choose a node having more common communities with the destination as a relay node, the chosen node may be a node with low local activity in its community. In this paper, we introduce the inner product method to define social similarity.

Definition 3 (Social Similarity): Given two activity vectors $A_u^t = (a_{u1}^t, a_{u2}^t, ..., a_{uj}^t, ..., a_{uk}^t)$ of node u and $A_w^t = (a_{w1}^t, a_{w2}^t, ..., a_{wj}^t, ..., a_{wk}^t)$ of node w, we define the social similarity between u and w at time t as SS_{uw}^t, having $SS_{uw}^t = A_u^t \cdot A_w^t$, where the symbol $\cdot$ denotes the inner product of vectors.

We assume that node w is the destination node. There exists a unicast session from node u to node w. The candidate relay is node v. If the social similarity between v and w is larger than u and w, there will be two facts that can be proved. One, the candidate node v has more common interests with the destination node w, *i.e.*, the number of non zero vector component is large. The other, referring to the destination node w, the candidate node v has high local activity values on the corresponding non zero vector component.

Using the inner product-based method, we need not to distinguish the different social features and it can guarantee the number of common interests and the high local activity. Besides, the method can deal with not only the uniform but also the nonuniform activity distribution in vectors. A special case is the binary value of activity. It is reduced to the case that the data forwarding only relies on the number of common interests.

3.3 LASS Algorithm

Based on Definition 1, 2 and 3, the description of LASS is presented in Algorithm 1. In order to make it clear, an example is also given to show the process of the data forwarding.

Algorithm 1 LASS: a session from node u to w at time t

```
for each encountered node $v_i$ do
  calculate $SS_{uw}^t$ and $SS_{v_iw}^t$
    if $SS_{v_iw}^t > SS_{uw}^t$
      add $SS_{v_iw}^t$ to the set $Temp^t$
if $Temp^t \neq \emptyset$
  sort the values in set $Temp^t$ in descending order
  choose the largest $SS_{v_iw}^t$ from $Temp^t$
    node $u$ transmits the message to node $v_i$
else
    node $u$ maintains the message
```

There is an example as shown in Fig. 3. At time t, node u transmits a message to destination node w in the mobile social network. Node u meets nodes v_1 and v_2. The vectors of node u, v_1, v_2 and w are

$$\begin{aligned}
A_u^t &= (0.5, 0, 0.5, 0.2, 0.3, 0) \\
A_{v_1}^t &= (0.8, 0,\ 0\ , 0.9, 0.5, 0) \\
A_{v_2}^t &= (0.7, 0, 0.5, 0.6,\ 0\ , 0) \\
A_w^t &= (0.9, 0, 0.3, 0.5, 0.7, 0)
\end{aligned}$$

According to Algorithm 1, we calculate the social similarity and gain $SS_{uw}^t = 0.91$, $SS_{v_1w}^t = 1.52$, $SS_{v_2w}^t = 1.08$. From above results, node v_1 and v_2 can both use as the next hop. But $SS_{v_1w}^t > SS_{v_2w}^t$, so we finally choose node v_1 and transmit the message from u to v_1. After that, node v_1 keeps on doing the similar operations like above ways. The performance evaluation about LASS will be presented in Section 5.3.

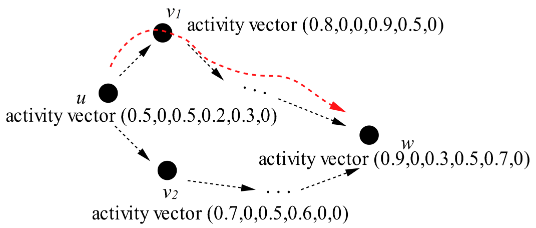

v_1
activity vector (0.8,0,0,0.9,0.5,0)
u
activity vector (0.5,0,0.5,0.2,0.3,0)
w
activity vector (0.9,0,0.3,0.5,0.7,0)
v_2
activity vector (0.7,0,0.5,0.6,0,0)

Fig. 3: Node u transmits a message to destination node w. At time t, u meets two nodes v_1 and v_2. Through LASS algorithm, finally, we choose node v_1 as the next hop. The data forwarding path is marked by the red dotted line with arrow.

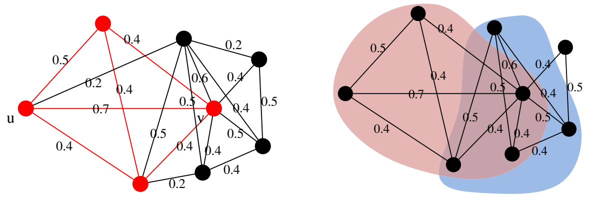
4 Self-Adaptive Weighted Dynamic Community Detection

In LASS algorithm, we need to fix the community structure and gain the value of node local activity. Thus, an efficient community detection algorithm is required in our dynamic weighted graph. There have existed many related algorithms [15]–[19] and some of them can be transplanted to the dynamic weighted graph, but they have many problems. For example, K-clique-based algorithms [16], [17] require prior community information about K as inputs. It is not real for social networks, because we can not know the number of communities in advance,*i.e.*, the prior value of K. Some modularity-based algorithms [15], [18], [19] have the problems of resolution limit and extreme degeneracy [20]. In addition, when meeting the dynamic environment, some algorithms need to repeat identification. The cost of time and computation is very high.

We refer to algorithm AFOCS in literautre [6] and design our self-adaptive weighted dynamic (SAWD) community detection algorithm[1]. AFOCS is a newly overlapped and dynamic detection algorithm. It avoids above-mentioned problems. But, it is only for unweighted graph and has some repeated operations on nodes and edges. Because when nodes change, the corresponding edges must also change. When the number of adding or removing nodes is large, the unnecessary repeated work becomes bored.

First of all, SAWD gives some weighted concepts and criterions(*e.g.* weighted density embryo, weighted criterion of communities) to handle the weighted graph. Then, SAWD classifies the adding or removing nodes or edges into "out-poll" and "in-pool" cases to save repeated operations. SAWD has two steps: 1) We treat the mobile social network as a static one and identify the initial weighted community structure using a weighted criterion in Section 4.1. 2) We deal with the evolving structures using the local information of community structure, *i.e.*, a distributed tracking method in Section 4.2.

[1] SAWD refers to algorithm AFOCS, some correctness verification about the algorithm can be seen in [6].



(a) (b)

Fig. 4: In fig (a), the red subgraph shows a weighted density embryo $O_t(u, v)$. Correspondingly, $E_t(u, v)$ is the set of red lines. In fig (b), the red community is generated by the edge weighted 0.7 and the blue community arises from the edge weighted 0.6. Two communities overlap with each other.

4.1 Initializing Community Structure

First, according to a criterion *weighted density* (Equation 1), nodes are classified into different groups, *i.e.*, raw communities. Then, on the basis of a *combining criterion of communities* (Definition 8), the highly overlapped raw communities will merge.

Definition 4 (Communication Critical Value): Define the *communication critical value* at time t, denoted by x_t, as $x_t = w_t^M$, where w_t^M is the median value of the set of weights W_t.

Communication critical value reflects a medial encounter probability or traffic between any two nodes in social networks. Thus, a sequence $\{x_t\}$ is formed over time. x_t is important because it can avoid some low weighted edges to form meaningless communities in social networks, *i.e.*, those rarely meeting nodes can not form communities.

Given a communication critical value x_t, we can obtain a spanning subgraph of G_t by deleting edges whose weights are smaller than x_t; we call such spanning subgraph *filtered graph*, and denote it by $G_t(x_t)$.

Before defining the *community*, we give a notion called *weighted density embryo*.

Definition 5 (Weighted Density Embryo, WDE): Given an edge (u, v) at time t, an induced subgraph of $G_t(x_t)$ whose all nodes belong to $N_t(u) \cap N_t(v)$ is called x_t-level *weighted density embryo* (WDE) generated by (u, v) at time t, denoted by $O_t(u, v; x_t)$.

For brevity, denote WDE $O_t(u, v; x_t)$, the node and edge sets of $O_t(u, v; x_t)$ by $O_t(u, v)$, $V_t(u, v)$ and $E_t(u, v)$, respectively, without confusion.

An example of weighted density embryo $O_t(u, v)$ is depicted in Fig.4(a).

Then, we define the *weighted density* of WDE $O_t(u, v)$ by

$$\Phi(O_t(u, v)) = \frac{|E_t(u, v)|}{\binom{|V_t(u,v)|}{2}}. \tag{1}$$

Now, we can give a weighted criterion for determining whether a WDE is a *community*.

Definition 6 (Weighted Criterion of Communities): A WDE $O_t(u, v)$ is a community iff the weighted density satisfies that $\Phi(O_t(u, v)) \geq \delta(O_t(u, v))$, where

$$\delta(O_t(u, v)) = \frac{\binom{|V_t(u,v)|}{2}^{1-\frac{1}{\binom{|V_t(u,v)|}{2}}}}{\binom{|V_t(u,v)|}{2}}.$$

Algorithm 2 Constructing Initial Community Structure

Input: $G_0 = (V_0, E_0, W_0, F_0)$
Output: the set of initial communities $\mathcal{C}_{init}$

```
x_0 ← apply Definition 4 on G_0
E' ← E_0
for each $w_{uv} \in W_0$
  if $w_{uv} < x_0$
    $E^{'} \leftarrow E^{'} \setminus (u, v)$
sort the edge weight in a descending order
from the largest weighted edge $(u, v) \in E^{'}$
  if $Com_t(u) \bigcap Com_t(v) = \emptyset$
    find $O_t(u, v)$ according to Definition 5
    if $\Phi(O_t(u, v)) \geq \delta(O_t(u, v))$ and $|V_t(u, v)| \geq 4$
      $\mathcal{C}_{raw} = \mathcal{C}_{raw} \bigcup \{V_t(u, v)\}$
$\mathcal{C}_{init} \leftarrow \mathcal{C}_{raw}$
for $C_i, C_j \in \mathcal{C}_{raw}$ and !Done
  if $\Gamma(C_i, C_j) \geq \alpha$
    $C^{'} \leftarrow$ combine $C_i$ and $C_j$
    $\mathcal{C}_{init} = (\mathcal{C}_{init} \setminus \{C_i, C_j\}) \bigcup \{C^{'}\}$
    Done ← False
```

The threshold $\delta(O_t(u, v))$ is an increasing function [6], a relaxation version of the traditional density threshold, *e.g.*, complete graph. According to Definition 6, some nodes and edges can be grouped into different raw communities, but there exist some substructures which are highly overlapped. Then, a combining criterion is necessary to help them merge into large ones. Before proposing the criterion, we define

Definition 7 (Coupling Coefficient): For two weighted communities, say C_i and C_j, the *coupling coefficient*, denoted by $\Gamma(C_i, C_j)$, is defined as:

$$\Gamma(C_i, C_j) = \frac{\sum_{(u,v) \in C_i \bigcap C_j} w_{uv}}{\min\{\sum_{(u',v') \in C_i} w_{u'v'}, \sum_{(u'',v'') \in C_j} w_{u''v''}\}} + \frac{\sum_{u \in C_i \bigcap C_j} \sum_{v \in C_i \bigcap C_j} w_{uv}}{\min\{\sum_{u' \in C_i} \sum_{v' \in C_i} w_{u'v'}, \sum_{u'' \in C_j} \sum_{v'' \in C_j} w_{u''v''}\}}.$$

The coupling coefficient is comprised of two parts, one is the intra edge weights ratio, the other is the intra node weights ratio. Based on it, we have

Definition 8 (Combining Criterion of Communities): Two communities C_i and C_j should be combined, if their coupling coefficient $\Gamma(C_i, C_j) \geq \alpha$, where α is a given threshold.

Note that the parameter α will be determined in the experiment in Section 5.1, *i.e.*, we will choose an optimal value of α that makes the community detection have good effectiveness. Fig.4(b) shows two weighted overlapping communities.

The procedures of constructing initial community structure are described in Algorithm 2.

4.2 Distributed Tracking Method

After constructing the initial communities, with the passage of time, the edge weights will vary due to strength changes

Algorithm 3 Finding Changed Edges

```
Input: the community structures $G_{t-1}$ and $G_t$
Output: the set of $\Delta E_t$
$x_{t-1} \leftarrow$ apply Definition 4 on $G_{t-1}$
$E_{t-1}^{'} \leftarrow E_{t-1}$
for each $w_{uv} \in W_{t-1}$
  if $w_{uv} < x_{t-1}$
      $E_{t-1}^{'} \leftarrow E_{t-1}^{'} \setminus (u, v)$
$x_t \leftarrow$ apply Definition 4 on $G_t$
$E_t^{'} \leftarrow E_t$
for each $w_{uv} \in W_t$
  if $w_{uv} < x_t$
      $E_t^{'} \leftarrow E_t^{'} \setminus (u, v)$
compare $E_{t-1}^{'}$ and $E_{t-1}^{'}$
  get the set of changed edges $\Delta E_t$
```

of social relationships, such as new people making friends with each other, users joining in or withdrawing from the entire social network or local communities. So, we need to cope with the dynamic changes. It shows in two aspects, one is the physical mobility, the other is the strength changes of relationships. Here, we compare a network to a "pool". Reflected in the weighted graph, the changes can be classified into two types: 1) the number of nodes changes and the weight of edges also changes, called "out-pool" changes; 2) the number of nodes does not change but the weight of edges changes, called "in-pool" changes.

Like AFOCS [6], we also handle the dynamic changes through adding or removing nodes or edges, but we classify them into "out-pool" and "in-pool" cases. Our SAWD algorithm can save repeated operations on nodes and edges, which results from the asymmetric processing results of two endpoints for an edge. As soon as finding the social changes, the distributed tracking method can deal with all nodes and edges changes simultaneously. The detailed procedures are presented in Algorithm 4 $\sim$ 7 in APPENDIX A and B. Some explanations about the tracking method are described as follows:

- Through checking the sets V_t and E_t, we can find the insertion and deletion actions of nodes and edges. Especially with varying edge weights, the adding and removing edges are found by Algorithm 3.
- For simplicity, we assume that every node has a community label set $Com_t(i)$, including solitary nodes. In final experiment results, if we find the number of nodes in a community is only one, we will discard it.
- We distinguish two types of nodes, one is *the foreign node* with its $Com_t(u) = \emptyset$ and its $N_t(u) = \emptyset$, *i.e.*, it is not in the current network pool. The other is *the solitary node* with its $Com_t(u) \neq \emptyset$ and its $N_t(u) = \emptyset$, *i.e.*, it is in the current network pool. Let C_s^t denote the set of solitary nodes at time t.
- The "out-pool" case includes adding foreign nodes to the current social network and removing nodes from the network. The "in-pool" case includes adding edges and removing edges operations.

Our object is to find a good community assignment which maximizes the overall internal weighed density function. According to above definitions and algorithms of SAWD, the high weighted substructures are clustered into different communities. As time goes by, the overall internal weighed density can always maintain the maximum value.

Algorithm 2 is centralized and only executes once at the initial constructing stage. The following dynamic Algorithm 4$\sim$ 7 uses local structure information to handle all changes in a distributed way.

5 Performance Evaluations

5.1 Parameter Choosing for Combining Threshold α

Our detection algorithm (SAWD) does not need any prior user-input information about communities, *e.g.*, the number of communities. The only parameter required to be fixed is the combining threshold value α. By the following NMI (Normalized Mutual Information) experiments, we determine an appropriate value for α to guarantee a good detection effect. Once gained, it will be used in the step of constructing initial static community and does not need to change in future dynamic operations. Moreover, it is only concerned with the detection method, not with the real online social networks.

5.1.1 Network Generation for NMI Experiments

We choose *LFR undirected and weighted benchmark* [21] to generate a synthetic social network. That is to say it can produce undirected weighted graphs with possible overlapping communities and satisfies the power-law degree distribution. We refer [22] to choose some parameters: exponent for the weight distribution $\beta = 1.5$ and the number of memberships for the overlapping nodes $om = 2$. We freeze the number of nodes $N = 1000$, topology mixing parameters $\mu_t = 0.1$ or $\mu_t = 0.5$ and the number of overlapping nodes $on = 0.1$ or $on = 0.3$. Then, we vary the weighted mixing parameter μ_w from $0 - 0.6$ to find the best value of α.

5.1.2 Metrics

We use NMI overlapping version [23] as metrics, *i.e.*, calculating the NMI score $N(X|Y)$. It is one of the most important entropy measures in information theory. $N(X|Y)$ can be interpreted as the average relative lack of information to infer random variable X given Y, $N(X|Y) \in [0, 1]$. The higher the NMI score is, the more similar the two community partitions are. If $N(X|Y)$ equals 1 , it means the two kinds of community partitions are exactly coincident. Therefore, we make our detection algorithm (SAWD) as X, the LFR benchmark as Y.

5.1.3 Experiment Results and Analysis

From large numbers of tests, we gain the combining threshold α ranging from $0 - 1.8$. We select the representative values $0.4 - 1.4$ to analyze an appropriate value for α . Because in this scope, the NMI score shows better than in other scopes. The experiment results are shown in Fig.6.

We can see, all curves declines as μ_w increases. μ_w indicates the fraction of the strength of a node which lies on links connecting the node to the nodes outside its community with respect to the total strength of this node [22]. The larger the μ_w is, the weaker the node strength is in its community. The weak strength results in the difficulty in community detection. Another view is from the horizontal and vertical perspectives. The small value of μ_t means a clear mixing topology and a dense community inner structure. The small on represents that the community structures approach to a disjoint status. The above two factors cause us to find the community structure easily. Thus, Fig.6 (a) shows the best performance in NMI tests and Fig.6 (d) is the worst case in all tests. As depicted, not only in the best case but also in the worst case, **the appropriate value for α is** 0.6. This value contributes a good NMI score, which is excellent for community partition.

5.2 Constructing Social Graphs

The evaluation of LASS (Local-Activity and Social-Similarity) is based on MIT Reality Mining Dataset [24]. In trace files, 97 Nokia 6600 mobile phones were carried by users over the course of nine months in MIT campus and its surroundings. In the long-term observation, phone users use Bluetooth sightings with 5 minutes interval to record the direct contacts between nodes. The trace files offer insights into the real world interactions between mobile users from different aspects and constitute a valuable database for various studies. But, they do not provide the information of social relationship directly. We need to construct a weighted social graph using bluetooth device scanning records in those files.

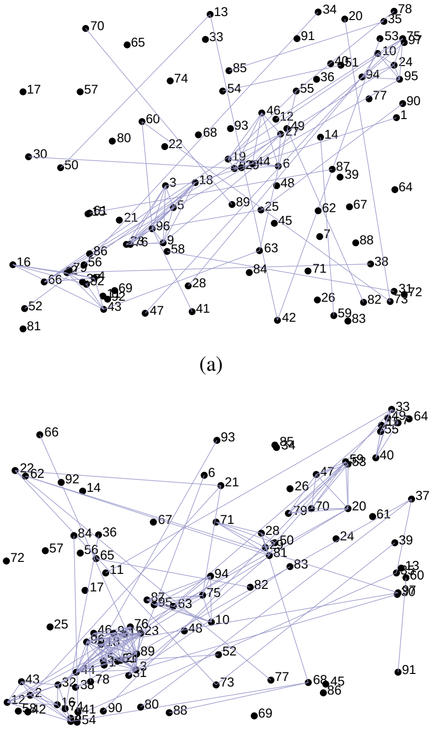
Based on the fact that the encounter probability can reflect the strength of social relationship in MSNs [24], we construct our social graphs as following steps.

• We capture a period between date $2004-09-10$ and $2005-03-23$ from the original data since there are no significant groups of device contacts before and after this period.

• We add the number of direct contacts between node pairs iteratively in chosen period t_1 to t_q. $\sum_{k=1}^{p} l_{ij}^{t_k}$ denotes the overall numbers of contacts between node i and j in time period t_1 to t_p, $\sum_{k=1}^{p} l_{*}^{t_k}$ denotes the overall numbers of contacts for all nodes. Thus, we have a matrix with $w_{ij}^{t_p} = \frac{\sum_{k=1}^{p} l_{ij}^{t_k}}{\sum_{k=1}^{p} l_{*}^{t_k}}$, where $1 \leq p \leq q$.

• For simplicity, we change the matrix to a symmetric one in order to cope with the invalidation of Bluetooth devices. Finally, we gain a weighted matrix representing the social graph with social relationship links.

Some discussion about the datasets selection and the meaning of the edge weight will be seen in Section 6.


(a)

(b)

Fig. 5: The dense parts are the potential community structures. The solid black lines stand for the weighted social links. Figure (a) and (b) are captured at $2004-10-06$ and $2004-10-16$ respectively.

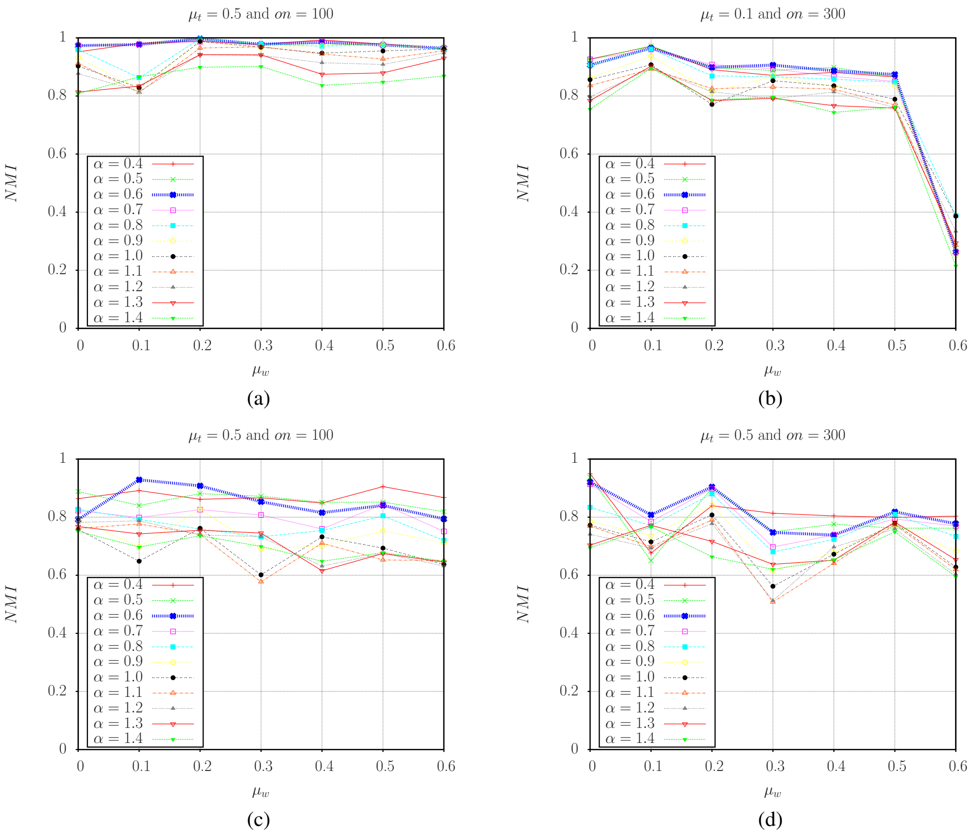
Here, in Fig.5, we give a visual presentation about social graphs founded on MIT Reality Mining Dataset. The two pictures are gained at different network snapshots. They have been processed by deleting some edges whose weights are smaller than communication critical value x_t. Because our constructing procedure is cumulative, with the passage of time, the social phenomenon (community structure) shows more and more clear and stable among users. Different shapes of communities are forming stable groups in Fig.5. This cumulative social phenomenon has positive effect on data forwarding.

5.3 Data Forwarding Experiment

5.3.1 Algorithm Comparison

In this section, we compare our LASS algorithm against Epidemic [25], PROPHET [13], Simbet [2], BUBBLE RAP [3] and Nguyen's Routing [6]. Above algorithms are "encounter-based" strategies in nature. Particularly, the last three have "social-aware" properties further.

- Firstly, we compare LASS with Epidemic and PROPHET. In Epidemic, each relay forwards the message to all the meeting nodes until arriving at its destination. In PROPHET, through encounter history, each node predicts the probability of future encounters. We use the default PROPHET settings as recommended in [13] and refer to [3] to set the aging parameter for delivery prediction.
- Secondly, we compare LASS with Simbet, BUBBLE RAP and Nguyen's Routing. In Simbet, it calculates betweenness and similarity for mobile nodes. A message is forwarded to a node if the node has higher simbet utility metric than the current one. We use the same parameters as stated in [2], *i.e.*, setting the similarity coefficient $\alpha = 0.5$ and betweenness



(a)

(b)

(c)

(d)

Fig. 6: Parameter Choosing for Combining Threshold α

coefficient $\beta = 0.5$. In BUBBLE RAP, it provides a hierarchical forwarding strategy. A node first bubbles the message up the hierarchical ranking tree using the global centrality. When the message reaches the community of the destination node, local centrality is used instead of the global centrality. In Nguyen's Routing, a smart community detection algorithm AFOCS is proposed and applied to data forwarding in mobile ad-hoc networks. A message is forwarded to a meeting node if the node shares more common community labels with the destination than the current one. Note that we select settings or parameters which bring about the best performances for above five contrastive algorithms respectively.

5.3.2 Simulation Setup

We choose the ONE simulator as our experimental tool [26]. It not only provides various mobile models including some complex mobility scenarios in daily life, but also can incorporate real world traces. In MIT trace files, one of the most important records is the contact between Bluetooth devices. It includes the start time, end time and communication peers. These discrete contact events can be taken as the inputs of the ONE simulator. In order to model connecting and disconnecting, we reorder the start times and end times. Corresponding to communication peers, we set the start time as up and the end time as down. The form of the extracted trace data is like:

$$\begin{array}{ccccc}
0 & CONN & 93 & 96 & up \\
0 & CONN & 93 & 14 & up \\
128 & CONN & 85 & 17 & up \\
129 & CONN & 94 & 29 & up \\
 & & \cdots & & \\
1169 & CONN & 28 & 5 & down \\
1169 & CONN & 28 & 17 & down
\end{array}$$

For all simulations conducted in this work, each node generates 1000 packets during the simulation time. The packet size is distributed from $50KB$ to $100KB$ uniformly. Data transmission speed is $2Mbps$ and the transmission range is in $10m$. The buffer size of each node is $5MB$. The source and destination pairs are chosen randomly among all nodes. Each emulation is repeated 20 times with different random seeds. Without losing precision, we set the update interval is 1. The interface of the underlying network is assigned to Bluetooth.

5.3.3 Metrics

- Delivery Ratio: the ratio of the number of successfully delivered messages to the total number of created messages.
- Average Delay: the average messages delay for all the successful sessions.
- Overhead Ratio: the proportion of the difference between the number of relayed messages and successful-

ly delivered messages out of the successfully delivered messages.

5.3.4 Experiment Results and Analysis

MIT Reality Mining Dataset is a long-term observation repository. Thus, some cumulative social phenomena (local activity, community structure *etc.*) require a period of time to reveal. Here, we make TTL[2] from 30min to 1mon. The experiment results are illustrated in Fig.7 (a)-Fig.7 (f).

Fig.7 (a)-Fig.7 (c) show the delivery ratio, overhead ratio and average latency of LASS, Epidemic and PROPHET algorithms respectively.

In terms of delivery ratio, shown in Fig.7 (a), due to Epidemic's flooding-based copy strategy, Epidemic performs better than PROPHET during the initial phase. At time 3 days, it reaches the peak value, close to 50 percent. But after that, the delivery performance decreases because of network congestion resulting from large numbers of copies [3]. Similarly, the turning point also appears in PROPHET at 1 week because it needs redundant relays to adapt to the fluctuation of meeting probabilities [3]. But, due to using the encounter history to predict the next hop, after the turning point, its delivery ratio shows better than Epidemic. By contrast, LASS goes up in steps and shows the best delivery performance among them, which outweighs Epidemic 32.63 percent and PROPHET 81.5 percent. At 1 month, it finally reaches 66.74 percent delivery ratio. Although LASS is also an unlimit copy algorithm like above two, its turning points will not emerge too early. This is because we use social similarity strategy to forward data. It means that the relay node has a high chance to meet the destination.

In terms of overhead ratio, illustrated in Fig.7 (b), due to the nature of flooding, the disadvantage of Epidemic is obvious among the three algorithms. It exceeds 64.64 percent than PROPHET. However, LASS performs well thanks to its social similarity forwarding scheme. To a great extent, through the whole TTL experiment period, the scheme controls the number of relaying copies with the overhead ratio only at 26.07 percent on average. In Fig.7 (c), the delays of all the three algorithms arise with TTL increasing. But, because of Epidemic's large numbers of copies, it can achieve the lowest delay rapidly among three. PROPHET falls in between Epidemic and LASS. LASS is a little higher than PROPHET with 19.01 percent. This is because the balance effect between the number of copies and the precise relay choosing strategy, *i.e.*, when we aspire to find a good data forwarding scheme, the cost may be a little higher delay which is due to a relatively fewer copies.

Fig.7 (d)-Fig.7 (f) show the delivery ratio, overhead ratio and average latency of our LASS, Simbet, BUBBLE RAP and Nguyen's Routing algorithms respectively.

In Fig.7 (d), LASS performs best among four algorithms. Its delivery ratio is higher than Nguyen's Routing with 34.64 percent, BUBBLE RAP with 46.18 percent and Simbet with 120 percent on average. Simbet and BUBBLE RAP use betweenness as centrality metrics without considering node contact frequency. As long as there exists an edge between two nodes, the edge will be used in betweenness calculation. But, in real social networks, the edge may only have a trivial effect in data forwarding. Thus, both of them have a lower delivery ratio than LASS. Nguyen's Routing tends to send messages to nodes having many interests with the destination, but, it may deliver them to nodes which have low activity in their communities (or interests groups). It is the main reason for the low delivery ratio of Nguyen's Routing.

In Fig.7 (e), the overhead ratio of Simbet and BUBBLE RAP are much higher than LASS and Nguyen's Routing. The reason is LASS and Nguyen's Routing prefer to choose the similar interests nodes as relays, which can control the number of copies in sessions. In the enlarged legend, the overhead ration of LASS and Nguyen's Routing are descending with TTL increasing. This is because both of them use social similarity strategy (one use the inner product of local activity vectors as similarity, the other use the number of common interests as similarity) to delivery message. As time goes by, the social phenomena are becoming more and more clear, which makes the algorithms more and more suitable for the social network, *i.e.*, just fewer copies can handle the data forwarding. On average, LASS keeps a low overhead ratio of 26.07. It is better than Nguyen's Routing which is 44.72. This is due to that LASS is good at finding high activity nodes in pace with dynamic network. In Fig.7 (f), the delays of four algorithms go up with TTL increasing. LASS and Nguyen's Routing are close to each other and slightly higher than BUBBLE RAP and Simbet. The gap is caused by few copies of LASS and Nguyen's Routing .

From above results and analysis, LASS has proved its competitive ability, which can achieve 66.74 percent delivery ratio and only have 26.07 percent overhead ratio with controlled delay in a reasonable range.

6 Discussion

In this section, we further explain several issues about our algorithm's designing and give some possible extensions for the future.

6.1 DataSets Selection

There exist many collections of social networks datasets, such as CRAWDAD[4], Haggle iMotes[5] projects and Stanford SNAP Graph Library [6]. In above collections, Infocom 06 dataset, Sigcomm09 dataset, MIT Reality Mining

[2] We do experiments from date 2004-10-01 to date 2004-11-01. Because MIT is a long-term observation dataset, we choose a large TTL-1 month, instead of several days. A larger TTL(larger than 1 month) also can be done with more simulation time. But through our experiment analysis, the overall trend is similar with 1 month.

[3] Here, in order to avoid the serious declining of performance, we process Epidemic algorithm with copy-limits. Although the turning point still exists, its performance will not descend too much and will maintain relatively steady as time goes by.

[4] http://crawdad.cs.dartmouth.edu/

[5] http://www.haggleproject.org

[6] http://snap.stanford.edu/data/index.html

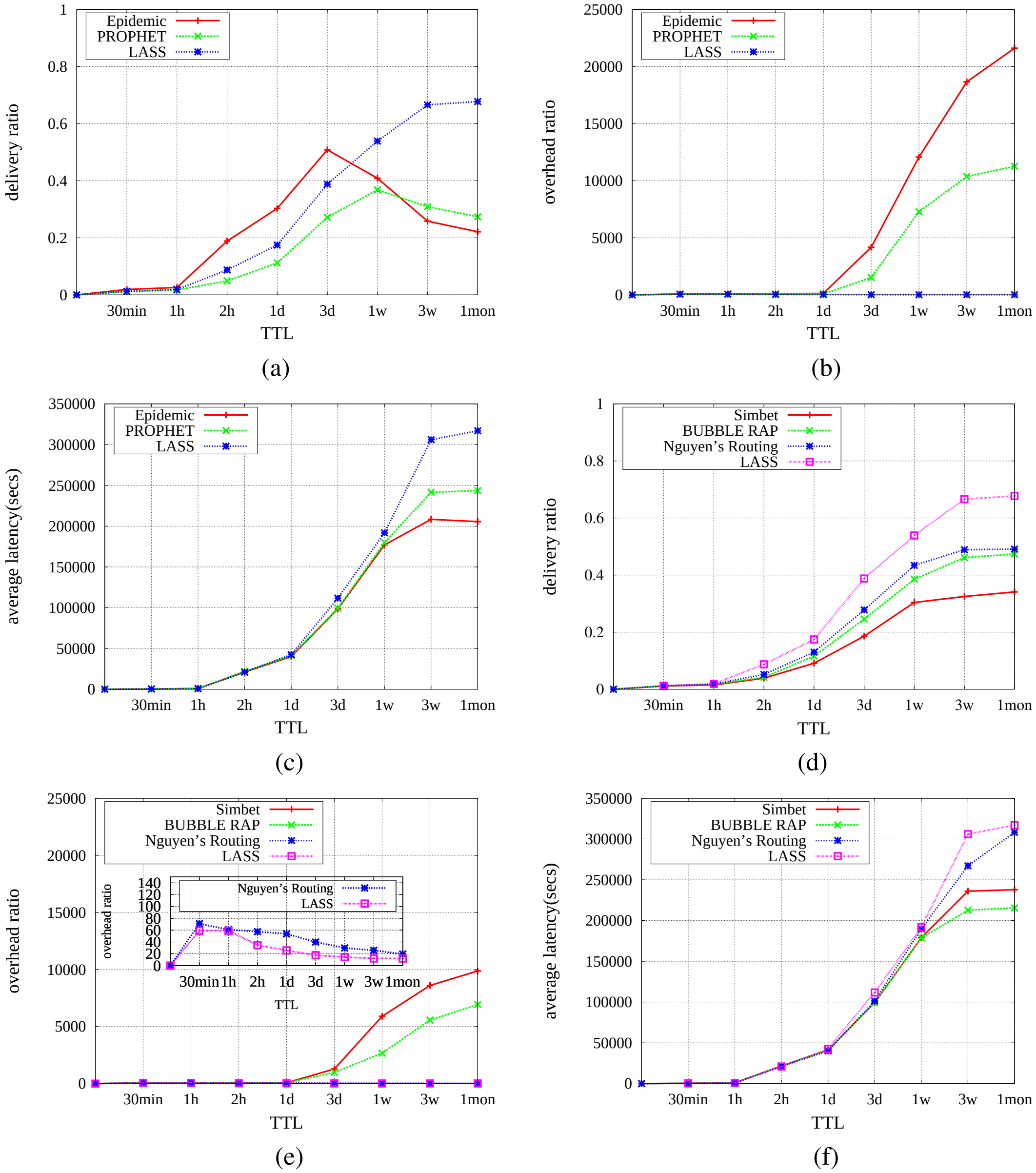


Fig. 7: Simulation Results on Reality Mining Dataset

dataset and Facebook dataset will be found. Based on them, some studies about relationship inference, behavior medeling and prediction, complex social studies, and information dissemination are carried out. In these datasets, one kind is the social friendship information, the other kind is the social proximity information. The former is about logical relationship, the latter is about geographical relationship. In our study, because the underlying network is the distributed mobile ad-hoc network, the second kind data (using Bluetooth discovery to gain proximity information) is appropriate. In order to observe the social impact on data forwarding, we choose a long term observation-MIT Reality Mining Dataset as our experiment dataset. Our algorithm can also be applied to other datasets to validate it effectiveness.

6.2 Choice about the Communication Critical Value

With continual adding and removing actions, the communication critical value x_t is generated by calculating the median value of the weight set W_t at each time snapshot. This method can tackle both uniform and power-law distribution. But, maybe, there will have some more precise mathematical methods than ours to deal with the problem, which can be studied in the future.

6.3 Edge Weight

In our study, the edge weight represents the encounter probability between two nodes. The edge weight may be concerned with mobility intensity, traffic interests or some other physical/logical social properties. But it is not in the scope of our research.

6.4 Bounds for Combining Criterion of Communities

Like [6], we also use experimental method to gain the threshold for combining criterion of communities, *i.e.*, Definition 8. But in theory, we could give the upper and lower bound of it. The bounds meet two conditions, one is after combining, the community structure satisfies the weighted criterion of communities, *i.e.*, Definition 6; the other is if two communities can combine with each other, we should try our best to do this. This theory bounds will be studied in the future.

7 Related Work

In mobile social networks, most data forwarding algorithms are "encounter-based". Epidemic [25] and Spray-and-Wait [27] are **"simple encounter-based"** algorithms. After that, some studies [13], [28]–[30] use the history of node contacts, spatial information or contextual information to predict the future encounter probability. These **heuristic methods** aim at finding appropriate relay nodes who are like to meet the destination nodes.

Recently, people have found social information has big impact on data forwarding. Because some social relationships can reflect people's preference, which is important in node encounter prediction. Therefore, some **"social-aware encounter-based"** algorithms emerge. They can be classified into two kinds.

On the one hand, some studies have shown that exploiting social relationships can achieve better data forwarding performances (our work is belonged to this kind). Daly and Haahr [2] proposed SimBet data forwarding algorithm in delay tolerant MANETs. It uses betweenness centrality and social similarity to increase the probability of a successful data forwarding. Authors show that SimBet performs well, especially when the connectivity is low. **But, it does not consider contact frequencies between node pairs.** Hui *et al.* [3] proposed an algorithm called BUBBLE RAP in DTNs, with making use of node centrality and weighted k-clique community structure to enhance delivery performance. It is better than Daly and Haahr [2]. **But, it needs to give a priori value of k to identify meaningful community structure, which is impractical in mobile social networks. Moreover, it has the same problem with [2], *i.e.*, using betweenness to calculate global and local centrality, not considering node encounter probability.** Gao *et al.* [4] studied multicast in DTNs from the social network perspective. With known community structures, authors formulates the relay selection as a unified knapsack problem. **But this method assumes that community structures are already known and some parameters' optimization requires global information to support.** Fan et al. [31] studied a geo-community-based broadcasting schemes for mobile social networks by exploiting node geo-centrality and geo-community. Nguyen *et al.* [6] proposed an overlapping community based data forwarding algorithm, called Nguyen's Routing. An efficient community detection method is designed for tracing the evolution of the overlapping communities in mobile networks. Taking advantage of the overlapping community structure, Nguyen's Routing uses the number of common interests as social similarity to design data forwarding scheme. **However, it only focuses on the binary graph and does not consider the node local activity.**

On the other hand, from another perspective, some studies have demonstrated the social relationships limit the freedom transmission between two nodes. Li et al. [5] introduced socially selfish properties into data forwarding scheme in delay tolerant networks, where protocol SSAR considered both users' forwarding willingness and their contact opportunity. Li et al. [7] studied a joint rate control, routing, and capacity allocation scheme to achieve optimal multirate multicast in dynamic wireless networks, which addressed social selfishness of users by differentiating relay costs towards different destinations. Lin *et al.* [8] proposed a PrefCast algorithm. It considers users' heterogeneous preferences for different content objects in mobile social dissemination, and meanwhile produces the maximal total utility for all users. Wu *et al.* [9] proposed a social feature-based multipath routing scheme in DTNs. It is based on the idea that the social features will play an important role in data forwarding in social contact networks. Finally, the scheme makes the routing problem become a hypercube-based feature matching process.

8 Conclusion

This paper designs a social-based data forwarding scheme for mobile social networks, called LASS (Local-Activity and Social-Similarity), which measures the social similarity through nodes' different levels of activity. Extensive simulations on MIT Reality Mining Database show that it can achieve a very good data forwarding performance in an efficient way. To the best of our knowledge, our work is the first one to be aware of different levels of activity within communities; furthermore, it enriches the methodology of social similarity's measurement.

There are several interesting issues left to study. First, we can consider the direction of the social relationship in the future. Second, in this paper, we only concern with the unicast in mobile social networks. But in reality, there exist many session modes, for example, broadcast, multicast.

References

[1] N. Kayastha, D. Niyato, P. Wang, and E. Hossain, "Applications, architectures, and protocol design issues for mobile social networks: A survey," *Proceedings of the IEEE*, vol. 99, no. 12, pp. 2130–2158, 2011.

[2] E. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," in *Proc. ACM MobiHoc 2007*.

[3] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 11, pp. 1576–1589, 2011.

[4] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in delay tolerant networks: a social network perspective," in *Proc. ACM MobiHoc 2009*.

[5] Q. Li, S. Zhu, and G. Cao, "Routing in socially selfish delay tolerant networks," in *Proc. IEEE INFOCOM 2010*.

[6] N. P. Nguyen, T. N. Dinh, S. Tokala, and M. T. Thai, “Overlapping communities in dynamic networks: their detection and moibile applications,” in *Proc. ACM MobiCom 2011*.

[7] H. Li, C. Wu, Z. Li, W. Huang, and F. Lau, “Stochastic optimal multirate multicast in socially selfish wireless networks,” in *Proc. IEEE INFOCOM 2012*.

[8] K. Lin, C. Chen, and C. Chou, “Preference-aware content dissemination in opportunistic mobile social networks,” in *Proc. IEEE INFOCOM 2012*.

[9] J. Wu and Y. Wang, “Social feature-based multi-path routing in delay tolerant networks,” in *Proc. IEEE INFOCOM 2012*.

[10] M. Girvan and M. Newman, “Community structure in social and biological networks,” *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, p. 7821, 2002.

[11] M. Porter, J. Onnela, and P. Mucha, “Communities in networks,” *Notices of the AMS*, vol. 56, no. 9, pp. 1082–1097, 2009.

[12] S. Fortunato, “Community detection in graphs,” *Physics Reports*, vol. 486, no. 3-5, pp. 75–174, 2010.

[13] A. Lindgren, A. Doria, and O. Schelen, “Probabilistic routing in intermittently connected networks,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 7, no. 3, pp. 19–20, 2003.

[14] F. Papadopoulos, M. Kitsak, M. Á. Serrano, M. Boguná, and D. Krioukov, “Popularity versus similarity in growing networks,” *Nature*, vol. 489, no. 7417, pp. 537–540, 2012.

[15] M. E. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Physical review E*, vol. 69, no. 2, p. 026113, 2004.

[16] 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.

[17] M.-S. Kim and J. Han, “A particle-and-density based evolutionary clustering method for dynamic networks,” *Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 622–633, 2009.

[18] D. Duan, Y. Li, Y. Jin, and Z. Lu, “Community mining on dynamic weighted directed graphs,” in *Proc. ACM CNIKM 2009*.

[19] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008.

[20] A. Khadivi, A. Rad, and M. Hasler, “Network community-detection enhancement by proper weighting,” *Physical Review E*, vol. 83, no. 4, p. 046104, 2011.

[21] A. Lancichinetti and S. Fortunato, “Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities,” *Physical Review E*, vol. 80, no. 1, p. 016118, 2009.

[22] ——, “Community detection algorithms: A comparative analysis,” *Physical Review E*, vol. 80, no. 5, p. 056117, 2009.

[23] A. Lancichinetti, S. Fortunato, and J. Kertész, “Detecting the overlapping and hierarchical community structure in complex networks,” *New Journal of Physics*, vol. 11, p. 033015, 2009.

[24] N. Eagle, A. Pentland, and D. Lazer, “Inferring friendship network structure by using mobile phone data,” *Proceedings of the National Academy of Sciences*, vol. 106, no. 36, pp. 15 274–15 278, 2009.

[25] A. Vahdat, D. Becker *et al.*, “Epidemic routing for partially connected ad hoc networks,” Technical Report CS-200006, Duke University, Tech. Rep., 2000.

[26] A. Keränen, J. Ott, and T. Kärkkäinen, “The one simulator for DTN protocol evaluation,” in *Proc. ICST SIMUTools 2009*.

[27] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Spray and wait: an efficient routing scheme for intermittently connected mobile networks,” in *Proc. ACM SIGCOMM 2005*.

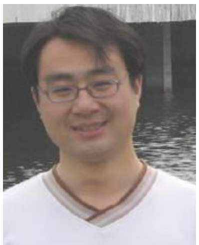
[28] J. Burgess, B. Gallagher, D. Jensen, and B. Levine, “Maxprop: Routing for vehicle-based disruption-tolerant networks,” in *Proc. IEEE INFOCOM 2006*.

[29] J. LeBrun, C. Chuah, D. Ghosal, and M. Zhang, “Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks,” in *Proc. IEEE VTC 2005*.

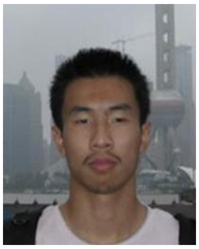
[30] M. Musolesi, S. Hailes, and C. Mascolo, “Adaptive routing for intermittently connected mobile ad hoc networks,” in *Proc. IEEE WoWMoM 2005*.

[31] J. Fan, J. Chen, Y. Du, W. Gao, J. Wu, and Y. Sun, “Geocommunity-based broadcasting for data dissemination in mobile social networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 4, pp. 734–743, 2013.

Zhong Li received her B.S. and M.S. degree at Department of Computer Science and Technology from Shandong Normal University in 2007 and 2010, respectively. She is currently a Ph.D. student in Department of Computer Science at Tongji University in Shanghai, China. Her research interests include wireless communication, social network analysis and distributed computing.



Cheng Wang received his Ph.D. degree in Department of Computer Science at Tongji University in 2011. Currently, he is a post doctoral fellow in School of Electrical Engineering and Computer Science at University of Ottawa. His research interests include wireless networking, mobile social networks, and mobile cloud computing.



SiQian Yang received his B.S. degree at Department of Computer Science and Technology from Tongji University in 2012. He is currently a Ph.D. student in Department of Computer Science at Tongji University in Shanghai, China. His research interests include social network analysis, delay tolerant networking and distributed computing.

Changjun Jiang received the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 1995 and conducted post-doctoral research at the Institute of Computing Technology, Chinese Academy of Sciences, in 1997. Currently he is a Professor with the Department of Computer Science and Engineering, Tongji University, Shanghai. He is also a council member of China Automation Federation and Artificial Intelligence Federation, the Vice Director of Professional Committee of Petri Net of China Computer Federation, and the Vice Director of Professional Committee of Management Systems of China Automation Federation. He was a Visiting Professor of Institute of Computing Technology, Chinese Academy of Science; a Research Fellow of the City University of Hong Kong, Kowloon, Hong Kong; and an Information Area Specialist of Shanghai Municipal Government. His current areas of research are concurrent theory, Petri net and formal verification of software, concurrency processing and intelligent transportation systems.

Xiangyang Li received M.S. (2000) and Ph.D. (2001) degree at Department of Computer Science from University of Illinois at Urbana-Champaign. He received his Bachelor degree at Department of Computer Science and Bachelor degree at Department of Business Management from Tsinghua University, P.R. China, both in 1995. He has been with Department of Computer Science at the Illinois Institute of Technology since 2000. Currently he is an Associate Professor of Department of Computer Science, IIT. He also holds visiting professorship or adjunct professorship at the following universities: TianJin University, WuHan University, NanJing University, Tongji University and Microsoft Research Asia. His research interests span the wireless ad hoc networks, computational geometry, game theory, and cryptography and network security. He served various positions (such as conference chair, local arrangement chair) at numerous international conferences. He is an editor of Ad Hoc & Sensor Wireless Networks: An International Journal. He recently also co-organized a special issue of ACM MONET on non-cooperative computing in wireless networks and a special issue of IEEE Journal of Selected Area in Communications. He is a senior member of IEEE computer society, and a member of ACM.

Appendix A
"Out-Pool" Case

The "out-pool" case contains tracking foreign nodes algorithm and tracking missing nodes algorithm.

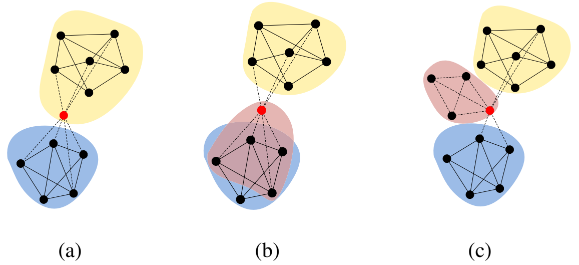

(a)
(b)
(c)

Fig. 8: The red node is a foreign node with added edges depicted by dotted lines. In (a), the foreign node joins its adjacent community. In (b), the foreign node forms new communities with neighbors. In (c), the foreign node unites the solitary nodes to form a new community.

Algorithm 4 Tracking Foreign Nodes

```
Input: the current community structure $\mathcal{C}_t$
Output: the updated structure $\mathcal{C}_{t+1}$
 1: if node $u$ is added without edges, then
 2:   $C_s^t = C_s^t \bigcup \{u\}$
 3: else $u$ with edges
 4:   $x_t \leftarrow$ apply Definition 4 on the community graph of $\mathcal{C}_t$
 5:   for each $w_{uv} \in W_t$
 6:     if $w_{uv} < x_t$
 7:       $E_t \leftarrow E_t \setminus (u, v)$
 8:   update the set of $N_t(u)$
 9:   $C_1, C_2, ..., C_k \leftarrow$ adjacent communities of $u$
10:   for $i = 1$ do to $k$
11:     $O_t(u, v) \leftarrow$ the induced subgraph of $G_t(x_t)$ based on $C_i \bigcup \{u\}$
12:       if $\Phi(O_t(u, v)) \geq \delta(O_t(u, v))$ and $|V_t(u, v)| \geq 4$
13:         $C_i \leftarrow C_i \bigcup \{u\}$
14:       else
15:   $O_t(u, v) \leftarrow$ the induced subgraph of $G_t(x_t)$ based on $C_i \bigcap N_t(u)$
16:         if $\Phi(O_t(u, v)) \geq \delta(O_t(u, v))$ and $|V_t(u, v)| \geq 4$
17:           define $V_t(u, v)$ of $O_t(u, v)$ as a new community $C^{'}$
18:   for $v \in C_s^t$ and $Com_t(u) \bigcap Com_t(v) = \emptyset$
19:       $O_t(u, v) \leftarrow$ the induced subgraph of $G_t(x_t)$ based on
20:                   $N_t(u) \bigcap N_t(v)$
21:         if $\Phi(O_t(u, v)) \geq \delta(O_t(u, v))$ and $|V_t(u, v)| \geq 4$
22:           define $V_t(u, v)$ of $O_t(u, v)$ as a new community $C^{'}$
23: merge overlapping communities on $C_1, C_2, ..., C_k$ and $C^{'}$
24: update $\mathcal{C}_t$ to $\mathcal{C}_{t+1}$
```

Firstly, we analyze Algorithm 4 about adding foreign nodes case. There are two possibilities, one is the node added without edges, the other is added with edges. If node u satisfies the former case, we simply join u to the current community structure. If u is the latter case, it becomes a little complicated and needs three operations, as illustrated in Fig. 8: 1) Because u is added with edges, it may join to its adjacent communities, *i.e.*, step $9-13$. 2) Uniting its neighbors, the foreign node u may form new communities, *i.e.*, step $15-17$. 3) Considering the set of solitary nodes, node u may shape new communities, *i.e.*, step $18-21$.

Secondly, we study Algorithm 5 about tracking missing nodes case. 1) If node u is a solitary node or $d_u = 1$, we simply remove the node from the current community structure. 2) Otherwise, there are two operations, as illustrated in Fig. 9. One is the remaining structure can maintain the original community, *i.e.*, step $8-11$, the other is the remains may form new communities, *i.e.*, step $13-16$.

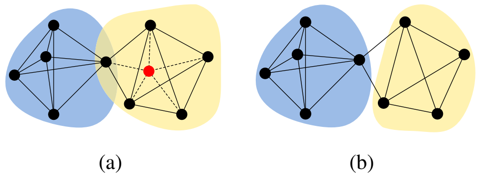

(a)
(b)

Fig. 9: The red node represents the missing node with removed edges depicted by dotted lines. In (a), the remaining structure can maintain the original shape. In (b), the remains forms two new communities.

Algorithm 5 Tracking Missing Nodes

```
Input: the current community structure $\mathcal{C}_t$
Output: the updated structure $\mathcal{C}_{t+1}$
 1: if $u$ is a solitary node or $d_u^t = 1$
 2:   $\mathcal{C}_t \leftarrow \mathcal{C}_t \setminus \mathcal{C}_t(u)$
 3: else
 4:   $x_t \leftarrow$ apply Definition 4 on the community graph of $\mathcal{C}_t$
 5:   for each $w_{uv} \in W_t$
 6:     if $w_{uv} < x_t$
 7:       $E_t \leftarrow E_t \setminus (u, v)$
 8:   for each subset $C_l$ in $\mathcal{C}_t(u)$ or in $\mathcal{C}_t(v)$
 9:     $O_t(u, v) \leftarrow$ the induced subgraph of the filter graph $G_t(x_t)$
10:               based on the remaining nodes in one subset $C_l$ of $\mathcal{C}_t(u)$
11:       if $\Phi(O_t(u, v)) \geq \delta(O_t(u, v))$ and $|V_t(u, v)| \geq 4$
12:         $C_l \leftarrow V_t(u, v)$
13:       else
14:         sort the weight of $E_t(u, v)$ in a descending order
15:         from the largest weighted edge $(u, v) \in E_t(u, v)$
16:   do Algorithm 2 step $8-10$ to gain new communities sequence $C^{'}$
17: merge overlapping communities
18: update $\mathcal{C}_t$ to $\mathcal{C}_{t+1}$
```

Appendix B
"In-Pool" Case

The "in-pool" case contains tracking adding edges algorithm and tracking removing edges algorithm.

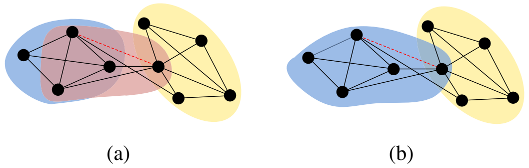

(a)
(b)

Fig. 10: The red dotted line represents an adding edge. In (a), the new edge shapes a new community. In (b), for an adding edge, one of its endpoints joins the community of the opposite side.

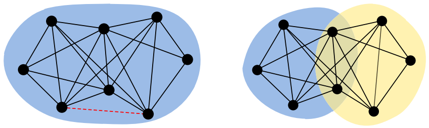
Firstly, we discuss Algorithm 6 about adding edges case. There are two possibilities, one is two endpoints of the adding edge are in the same community, the other is in the different communities. In the former case, community structure does not change, because adding edges increases the weighted density of communities. In the latter case, we further divide it into two operations, as illustrated in Fig.

Algorithm 6 Tracking Adding Edges

```
Input: the current community structure $\mathcal{C}_t$
Output: the updated structure $\mathcal{C}_{t+1}$
  if $Com_t(u) \bigcap Com_t(v) \neq \emptyset$
    $\mathcal{C}_{t+1} \leftarrow \mathcal{C}_t$
  else
    if $Com_t(u) \neq \emptyset$ and $Com_t(v) \neq \emptyset$
      if $Com_t(u) \bigcap Com_t(v) = \emptyset$ then
        $O_t(u,v) \leftarrow$ the induced subgraph of $G_t(x_t)$ based on
                 $N_t(u) \bigcap N_t(v)$
        if $\Phi(O_t(u,v)) \geq \delta(O_t(u,v))$ and $|V_t(u,v)| \geq 4$
          define $V_t(u,v)$ of $O_t(u,v)$ as a new community $C^{'}$
        else
          for each subset $C_l$ in $\mathcal{C}_t(u)$ or in $\mathcal{C}_t(v)$
            $O_t(u,v) \leftarrow$ the induced subgraph of $G_t(x_t)$ based on one
                subset $C_l$ of $\mathcal{C}_t(u) \cup \{v\}$ or one subset $C_l$ of $\mathcal{C}_t(v) \cup \{u\}$
            if $\Phi(O_t(u,v)) \geq \delta(O_t(u,v))$ and $|V_t(u,v)| \geq 4$
              $C_l \leftarrow C_l \cup \{v\}$ or $C_l \leftarrow C_l \cup \{u\}$
    if $(Com_t(u) = \emptyset$ and $Com_t(v) \neq \emptyset)$ or $(Com_t(v) = \emptyset$ and
       $Com_t(u) \neq \emptyset)$
      only do Algorithm 6 step $5-9$
  merge overlapping communities
  update $\mathcal{C}_t$ to $\mathcal{C}_{t+1}$
```

10. 1) If the adding edges come from current nodes, we decide whether the edge (u, v) can form a new community, *i.e.*, step $4-8$. Besides, we still need to judge whether the node u or v will join the community of the opposite side, *i.e.*, step $9-14$. 2) If the adding edges come from the new foreign nodes, we only need to process the edge (u, v), *i.e.*, judging whether to shape a new community or not, described in step $15-16$. **Some operations about two endpoints have been done in Algorithm 4**.

Secondly, we study Algorithm 7 about tracking removing edges case. There are also two possibilities, one is two endpoints of the removing edge are in the different community, the other is in the same communities. In the former case, the community structure does not change. In the latter case, as illustrated in Fig. 11, we only need to concern the case that the removing edges come from the existing networks. **Because if the removing edges come from the missing nodes, the corresponding operations have been done in Algorithm 5**. Therefore, in Algorithm 7, 1) We decide whether the remaining structure can still maintain or not, *i.e.*, step $4-8$. 2) Otherwise, the remains can form some new communities, *i.e.*, step $9-13$.


(a)

(b)

Fig. 11: The red dotted line represents a removing edge. In (a), the remaining structure can still maintain. In (b), the remaining structure forms two new communities.

Algorithm 7 Tracking Removing Edges

```
Input: the current community structure $\mathcal{C}_t$
Output: the updated structure $\mathcal{C}_{t+1}$
  if $Com_t(u) \bigcap Com_t(v) = \emptyset$
    $\mathcal{C}_{t+1} \leftarrow \mathcal{C}_t$
  else
    if $Com_t(u) \neq \emptyset$ and $Com_t(v) \neq \emptyset$
    for each subset $C_l$ in $\mathcal{C}_t(u)$ or in $\mathcal{C}_t(v)$
      $O_t(u,v) \leftarrow$ the induced subgraph of $G_t(x_t)$ based on the
      remaining nodes in one subset $C_l$ of $\mathcal{C}_t(u)$ after removing $(u,v)$
        if $\Phi(O_t(u,v)) \geq \delta(O_t(u,v))$ and $|V_t(u,v)| \geq 4$
          $C_l \leftarrow V_t(u,v)$
        else
          sort the weight in $E_t(u,v)$ in a descending order
          from the largest weighted edge $(u,v) \in E_t(u,v)$
    do Algorithm 2 step $8-10$ to gain new communities sequence $C^{'}$
  merge overlapping communities
  update $\mathcal{C}_t$ to $\mathcal{C}_{t+1}$
```