

Distributed Spatio-Temporal Social Community Detection Leveraging Template Matching

Yanzhi Ren*, Mooi Choo Chuah[†], Jie Yang*, Yingying Chen*

*Dept. of ECE, Stevens Institute of Technology [†] Dept. of CSE, Lehigh University
Castle Point on Hudson, Hoboken, NJ 07030 Bethlehem, PA 18015
{yren2, jyang, yingying.chen}@stevens.edu chuah@cse.lehigh.edu

Abstract—Community association is an important attribute of a social network because people may belong to varying groups with different characteristics at different times. Traditional community detection approaches often rely on a centralized server and are only useful for offline data analysis. In this paper, we propose and evaluate a distributed community detection approach that allows individual users to detect their own communities based on local observations. Our proposed template-matching method derives dynamic spatial and temporal characteristics of social communities by exploiting human’s mobility patterns. Our template matching method allows users with similar moving patterns to be grouped together as one community. Our results using both simulation as well as real experiments demonstrate that our method can detect local communities effectively with high detection rate and low false positive rate.

I. INTRODUCTION

In recent years, we have witnessed pervasive use of mobile devices in our lives. For example, mobile phones are used not only to communicate but also to sense our surroundings. Using RFID tags and mote sensors to collect useful patients’ data or track their whereabouts is another example. Toward this end, sensors (e.g. bluetooth), embedded within smartphones allow us to collect traces which are infeasible in the past. Such traces capture encounter events between humans and hence can be mined to extract social relationships between device owners. Furthermore, researchers have proposed schemes to extract not only social relationships between owners of mobile devices but also their social communities [1], [2], which make our daily life more convenient, e.g. virtual marketing among users of online social network sites, identifying popular online social media [3] of different social groups. In addition, social community information has been used in wireless networks in various ways, e.g., coping with the propagation of malware on smart phones [4], controlling the rate of disease propagation [5], and facilitating the packet forwarding in Delay Tolerant Networks (DTNs) [6]–[8].

Many community detection schemes require a centralized server [1], [2], [9]. [9] introduced the Kernighan-Lin algorithm and attempted to improve the initial division of a network by optimizing the number of graph edges within and between the partitions using a greedy algorithm. Further, a similarity measure is used in a hierarchical clustering scheme [1] to guide the merging of different communities. The merging process terminates when the overall community structures achieve the largest modularity Q value as described in [3]. All these methods rely on a centralized server. However, in

a self-organizing social network, it will be useful to let each mobile device infer its own local community [10] based on the information it collects instead of relying on a centralized server.

However, little work has been done to address the distributed community detection problem in a social network. Yang *et al.* [11] proposed to dispatch a group of special computational agents into a network to collect information for community extraction. They autonomously collect a local view of the network and partition the complete network into communities. Pan *et al.* proposed a distributed community detection scheme for Pocket Switched Networks [10], which is mostly related to our study. In this work, each mobile device senses and detects its own local community by analyzing the history of mobile devices it has encountered. Only encounter events are used to construct the social relationships between users.

Additionally, previous work mostly focus on extracting static community information by analyzing whole traces [10], [12]. These approaches only use the humans’ encounter events to extract social relationships [6]. There are a few drawbacks of static community extraction. First, it cannot capture the time-varying information presented in a trace [5]. Second, more encounter events do not necessarily mean a closer relationship between two users and vice versa. For instance, one active student may encounter many students as he moves around campus frequently. However, he does not necessarily has a close social relationship with each of these students. To address these issues, in this paper, we design a distributed community detection method that infers social relationships among users by taking into consideration of users’ moving patterns in the physical space over time instead of considering only encounter events between users. The moving patterns of users are extracted from traces collected by using users’ mobile device. We introduce the concept of community layers and derive social communities by leveraging template matching based on user’s spatial and temporal information. Template matrix is constructed and communicated between users to reduce the communication cost.

To validate the feasibility of our approach, we conducted experiments by applying our approach on real human moving traces collected using RFID tags in an office building environment. We further performed simulation studies based on analysis of synthetic traces generated using random waypoint group mobility model (RPGM) [13]. Our simulation results show that our approach can derive individual users’ local

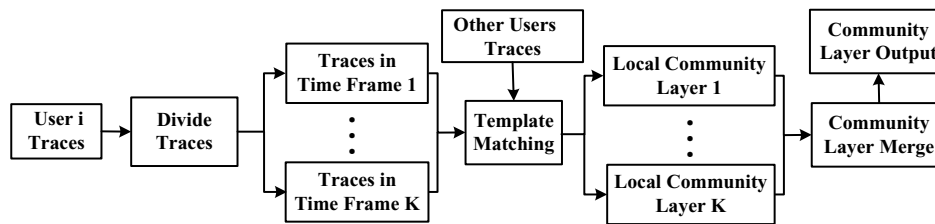


Fig. 1. Flow overview: components to extract community layers.

communities with high detection rate and low false positive rate.

The rest of the paper is organized as follows. In Section II, we present our proposed distributed community detection method. In Section III, we describe our experimental studies using RFID traces collected from a real office building environment. We also present our simulation results in Section III. Finally, we conclude our work in Section IV.

II. DISTRIBUTED EXTRACTION OF COMMUNITY INFORMATION

Only using encounter events to extract the social relationships between users is not enough to capture complex human behaviors in real-world scenarios. In this work, we propose a template matching approach, where users' temporal and spatial moving patterns are used to construct template and study whether the users involved have close relationships. We define a *community layer* as a group of local community members who appear during the same time period: the community members within the same community layer often meet frequently with one another during a time period. For local community members, we use the same definition as that in [10]: a user's local community contains a set of users who have close social relationships with himself.

Method Overview. Before we describe our method, we define a few concepts used in our method: the physical space is divided into multiple spatial areas with each area composing of multiple spatial partitions. Spatial and temporal traces collected from each mobile device include the spatial partitions each user visits during the measurement periods which may be a few hours or a day. Our distributed social community detection scheme is based on analyzing such traces collected from mobile devices.

The flow of our detection scheme is illustrated in Figure 1. Our detection scheme first searches through a user's collected trace to identify the time slots when a user moves from one spatial area into another. Then, our detection scheme divides this user's trace into different time frames according to those identified time slots. Thus, the length of each time frame is a variable depending on how soon a user moves from one area into another. Next, our detection scheme constructs a template matrix for each time frame extracted by the user. Each template matrix records the spatial partitions a user visits during a time frame. There will be two types of template matrices: one type records spatial partitions a user i visits in different time frames, and the other type are matrices which this user i receives from other users he encounters. Our template matching algorithm

is then used to compare user i 's template matrix with another user's template matrix from the same time frame. All users that have similar template matrices with user i will be included in i 's local community layer built for this time frame. The community layers extracted in different time frames will later be merged based on the number of common members they share to generate final reported social community structures.

In the following subsections, we first discuss how spatial partitions are defined. We then describe the necessary information that each node needs to keep for community detection. Next, we present how to divide each trace into different time frames according to the spatial partitions. Finally, we describe the method of detecting community members at different layers, and the strategy of merging similar community layers together.

A. Deriving Spatial Partitions

Recall that in our approach, the whole space is divided into multiple independent spatial areas. Each area is further subdivided into several partitions and thus each spatial area is often composed of multiple partitions. In our approach, each partition is a meaningful place, e.g., an office room, a classroom, a rectangular area or some other spatial divisions. Thus, the partition size and shape vary in different scenarios. We also assume that a unique identifier is assigned for each partition. Such partitioning of the physical space can be represented using a spatial partition set as $\{P_1, P_2, \dots, P_M\}$ where M is the total number of partitions. The above partitioning allows us to reduce the memory requirement for our method. Instead of using the accurate locations represented by x and y coordinates, each user now records only the defined spatial partition that he is currently located.

An example of spatial partitioning is depicted in Figure 2. In Figure 2, two large areas are defined: Area A is used for graduate students' laboratories and Area B is used as offices for faculty and staff. Area A is further subdivided into 6 partitions (P_1 to P_6) and area B is subdivided into 3 partitions (P_7 to P_9) according to the division of office rooms. Thus, the partition set can be represented as $\{P_1, P_2, \dots, P_9\}$.

Our method models the moving patterns of a user using a sequence of states where each state records the information of a spatial partition each user resides at a time slot. This information can be inferred from the traces collected via a user's wireless device [14], e.g., Bluetooth or RFID readings. Each wireless device sends out beacons periodically. For each time frame that consists of N beacon slots, a user can record his states as $S = \{s_{t_1}, s_{t_2}, \dots, s_{t_N}\}$ where t_1, t_2, \dots, t_N

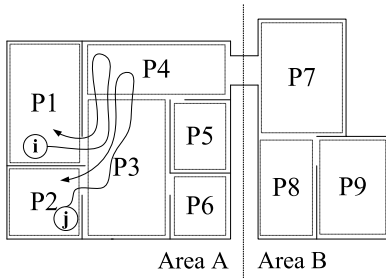


Fig. 2. Example of user i and j 's moving patterns under spatial areas and partitions defined by our scheme.

represent various beacon slots, and $s_{t_l} = \{(t_l, P_m), m \in M\}$ with P_m being the partition where the user resides at time slot t_l . In the example given in Figure 2, user i starts from partition P_1 , then enters into partition P_3 and P_4 before moving back into partition P_1 in 6 time slots. User i 's moving pattern can then be represented using a sequence of states $\{(1, P_1), (2, P_3), (3, P_4), (4, P_4), (5, P_3), (6, P_1)\}$. Similarly, the moving patterns of user j shown in Figure 2 can be represented as a sequence of states $\{(1, P_2), (2, P_3), (3, P_4), (4, P_4), (5, P_3), (6, P_2)\}$. To further optimize memory usage, user i 's states can be recorded as $\{(1, P_1), (2, P_3), (3, P_4), (5, P_3), (6, P_1)\}$ since he is at partition P_4 for both time slots 3 and 4. This optimization also helps to reduce the communication bandwidth requirement since in our method nodes (i.e., users) exchange such information when they encounter each other.

In our scheme, two tables are maintained at each node i for storing some relevant information: *Spatial Partition Table (SPT)* and *Others' Spatial Partition Table (OSPT)*.

- *SPT*: In this table, user i records the information of the spatial partitions he visits at each beacon slot. Each entry in this table includes the partition identifier representing where a user is at and the identifier of the corresponding time slot.
- *OSPT*: When user i encounters another user j , he stores j 's SPT and OSPT which it receives from j . Thus, user i can collect information of other users' moving patterns.

B. Appropriate Division of Time Frames

A straightforward method of extracting time frames is to divide the whole time duration into fixed time periods with each period counting as one time frame. However, different users may have different life patterns, e.g., some students often spend a whole day in the library while others may take various classes and move from one classroom to another. Thus, instead of using time frames with fixed durations, we design to extract time frames with variable durations for each individual.

We define a time frame of user i as a time period during which user i often stays with a fixed group of closeby users. Such set of people that stay in the same area often remain unchanged during that time period. For instance, when one student takes a class from 10:00 AM to 11:30 AM, his/her neighbors often form a stable set unless he/she gets out of the class area. Thus, the information of spatial areas where users are located can be used to help us appropriately subdivide each user's trace into multiple time frames.

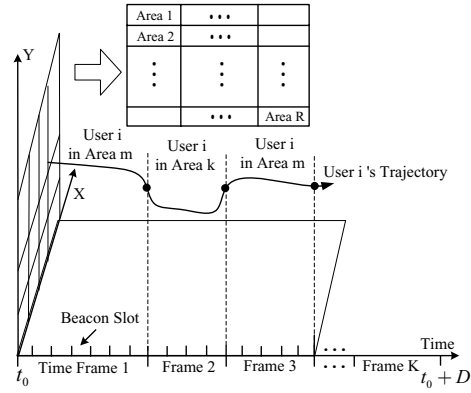


Fig. 3. The extraction of time frames.

An illustration of time frame extraction is shown in Figure 3. Assume that the whole geographical space is divided into R areas, e.g., Area 1 to Area R , the starting time slot is t_0 , and that the trace collection duration is D hours. Recall that each area composes of multiple spatial partitions. Our detection module searches through a device's SPT to determine when user i has moved from one area to another. If user i has moved from one area to another at time slot t_{move} , the first time frame will be $T_0 = (t_0, t_{move})$. The next time frame will have a starting timeslot of t_{move} . This process iterates through all the time slots during the D hours.

C. Template Matching Algorithm

Instead of using encounter events to determine if different users show similar moving behaviors, we compare users' moving patterns as represented by the states they maintain so that we can infer social relationships among these users. First, our scheme constructs template matrices of different users using the information maintained at each user's SPT and OSPT. Then, our scheme compares these matrices using a template matching algorithm we design to determine users with similar behavioral patterns to a particular user.

1) *Template Construction for Users*: Based on our time frame division strategy, user i can identify closeby users that show up in the same time frame as himself to be those that potentially have close social relationships with himself. Recall that the community layer of user i is defined as a set of closeby users with similar moving patterns as user i during one time frame. Our detection module analyzes the records in SPT and OSPT to identify a user's local community layers.

Assume that the entries in a user i 's SPT can be divided into K parts corresponded to K time frames. We denote N_k as the number of beacon slots in the k th time frame and the M as the total number of partitions. User i then constructs a $M \times N_k$ matrix C_k^i for the k th time frame and then fills it with the information extracted from the recorded states as follows: $\forall m, n, C_k^i(m, n) = 1$ if user i is in partition P_m at beacon slot n of the k th time frame; $\forall l, n, C_k^i(l, n) = 0.5$ if partition P_l is the neighbor of the partition P_m to which user i belongs at time slot n . Similarly, template matrices for other users (e.g., user j) in OSPT are also constructed: user i searches through its OSPT for the entries of another user

j . If there also exists N_k entries in the same k th time frame for user j , then user i also constructs a $M \times N_k$ matrix for user j based on these entries. This matrix is used to compare with that of user i for moving pattern similarity evaluation. Otherwise, no matrix is constructed for user j .

An example of the template matrix construction is illustrated in Figure 4. For two users i and j in Figure 2, we construct a template matrix which includes that user's moving patterns in the k th time frame. Since user i starts from partition P_1 at the first beacon slot, we fill the template matrix (C_k^i) with $C_k^i(1,1) = 1$. From beacon slot 2 to 6, user i enters into partition P_3 and P_4 and moves back to partition P_1 , as a result, we have $C_k^i(3,2) = 1$, $C_k^i(4,3) = 1$, $C_k^i(4,4) = 1$, $C_k^i(3,5) = 1$ and $C_k^i(1,6) = 1$. Furthermore, the entries corresponding to neighboring spatial partitions would be assigned a value of 0.5 in each slot as illustrated in Figure 4. The above process is repeated for all the time frames user i has from its SPT. Similarly, the template matrices for other relevant users (e.g., user j) are constructed using the information in OSPT. These constructed template matrices are used in the template matching process that is described next.

2) *Template Matching for Users*: The template matrices contain users' moving patterns and we analyze them inspired by the idea of template matching [15]. Specifically, to measure the similarity between each pair of template matrices extracted in each time frame, we use the Pearson correlation coefficient [16]. In general, the Pearson correlation coefficient (PCC) is a statistical method that measures the degree of the linear relationship between two given random variables. The Pearson correlation coefficient value ranges from -1 to 1. Correlation 1 and -1 means that there is a perfect positive/negative linear relationship between the two random variables. More specifically, given two corresponding rows (e.g., the d -th row) of template matrices C_k^i and C_k^j : $C_k^i(d) = \{i_1, i_2, \dots, i_{N_k}\}$ and $C_k^j(d) = \{j_1, j_2, \dots, j_{N_k}\}$, the Pearson correlation coefficient is defined as:

$$\text{corr}(C_k^i(d), C_k^j(d)) = \frac{1}{N_k - 1} \sum_{l=1}^{N_k} \left(\frac{i_l - \bar{i}}{\sigma_i} \right) \left(\frac{j_l - \bar{j}}{\sigma_j} \right) \quad (1)$$

where \bar{i} (\bar{j} , resp.) and σ_i (σ_j , resp.) are the mean and standard deviation of $C_k^i(d)$ and $C_k^j(d)$.

We compute the Pearson correlation coefficient between each pair of corresponding non-zero rows from two template matrices. The *total correlation* between two template matrices is the average of the Pearson correlation coefficients of these non-zero rows. User i calculates the correlation with user j which has the templated matrix constructed. If the total correlation is larger than a threshold T , user j will be listed in the current community layer A_k of user i for the k -th time frame.

An example of the PCC computation is illustrated in Figure 4. We compute the PCC between each pair of corresponding non-zero rows of user i and j 's template matrices, and also the total correlation between two template matrices.

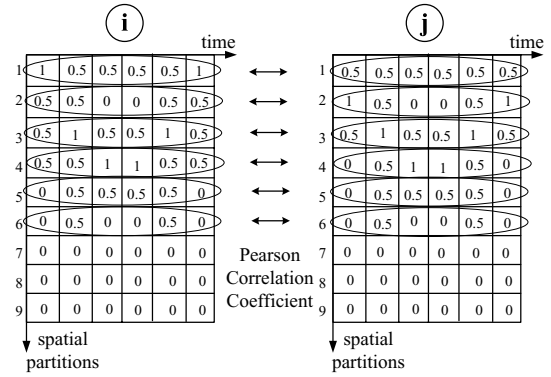


Fig. 4. Template matching.

3) *Merging the Community Layers*: Similar social communities may appear repeatedly at different time frames. Hence, communities discovered at different time frames need to be merged. Let the identified K community layers corresponding to the K time frames be: A_1, A_2, \dots, A_K . Community layer A_p is compared with all the other community layers (A_q) to see if A_p satisfies one of the following conditions:

- It is part of the bigger community A_q and hence can be removed.
- It can be merged with A_q using the community merge operation under an adjustable threshold τ :

$$\frac{|A_p \cap A_q|}{\text{Max}(|A_p|, |A_q|)} > \tau \quad (2)$$

- It is a superset of A_q , hence A_q is removed.

At the end of this operation, the two sets A_p and A_q are unioned to form a new A_p . This merging process iterates through the K community layers. The communities that remain at the end of this merging process will be reported as the final communities each user belongs to.

III. PERFORMANCE EVALUATION

In this section, we first study the feasibility of our proposed method using RFID traces collected in a real office building environment. Then, via simulations, we compare the performance of our proposed method with the k-clique approach developed in [10].

We use the following metrics to evaluate the effectiveness of our detection scheme: (a) **detection rate**: it is defined as

RFID Tags	Group Assignment		Template Matching		K-clique
	Scenario 1	Scenario 2	Layer 1	Layer 2	Local Community
1	2,4	8	2,4	8	2,4,8,9,10
2	1,4	6,7	1,4	6,7	1,4,7,10
3	10	5	10	5	5,10
4	1,2	9,10	1,2	9	1,2,8,9,10
5	6	3	6	3	3,6
6	5	2,7	5	2,7	5
7	8,9	2,6	8,9	2,6	2,8,9
8	7,9	1	7,9	1	1,4,7,9
9	7,8	4,10	7,8	4,10	1,4,7,8
10	3	4,9	3	9	1,2,3,4

TABLE I
EXPERIMENTAL ASSIGNMENTS AND DETECTION RESULTS

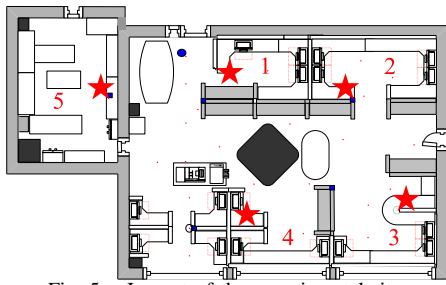


Fig. 5. Layout of the experimental site.

the percentage of local community members that are detected by the community detection scheme; (b) **false positive rate**: it is the percentage of local non-community members that are mistakenly detected.

A. Experimental Validation

1) *Experimental Setup*: We conducted our experiments on the second floor of Buchard building at Stevens Institute of Technology, which is a 70 ft by 80 ft area as shown in Figure 5. This is a large area containing office rooms and lounges and we subdivide it into five partitions denoted as 1 to 5 in Figure 5. We connect RFID readers to the Linux machines with 2 GHz CPU, 1 GB RAM and 20 GB disk to serve as our landmarks. Then, we deployed five landmarks which were used to collect the RSS readings from the RFID tags carried by the participants in our experiment. The locations of the landmarks are marked as the red stars in Figure 5. These locations are chosen to ensure that each partition has one landmark to record the location information of our participants so that we know which partition the participants are currently in.

We then recruited 10 graduate students and gave each of them a RFID tag. Each student is assigned to a group in each experimental scenario, and members within a group are expected to move together in each scenario. We define two scenarios and each scenario lasts for 60 minutes. In each scenario, the participants have different group members and the assignments of the group members are illustrated in the Group Assignment column of Table I. We also configured each RFID tag to send a beacon periodically so that our landmarks can collect the RSS readings from these RFID tags. The beacon time is set as 2 minutes. At every beacon time, each landmark records the RSS readings of participants currently within its reception range. We can then analyze these RSS readings to infer which partition a respective RFID tag (and hence a user) is at. The determination of which area a participant is currently located is based on the strongest RSS reading among all readings collected by landmarks that can hear that tag.

2) *Experimental Results*: The Template Matching column in Table I presents the community layer detection results of our proposed method using the traces collected from our experiment. When compared with the known group assignment in Table I, we found that each participant can detect almost all of their local community members in both Scenario 1 and 2: only Participant 4 and 10 did not detect their members in Scenario 2. The total detection rate is 0.93 and the false positive rate

is zero. In addition, Scenarios 1 and 2 are carried out by the same set of participants and hence represent how these users form different groups at different time periods. This indicates that our method can correctly categorize users into different community layers at different time periods.

Table I also presents the local community detection results of the k-clique method [10] in the K-clique column. Comparing our template matching based method with k-clique, the key observation is that our proposed method achieves a higher detection rate and lower false positive rate than the k-clique method: the detection rate is only 0.81 with a false positive rate of 0.11 for the k-clique method. Furthermore, the k-clique method cannot discriminate the difference in group memberships for Scenario 1 and 2. It can only list all the group members in the same local community.

B. Simulation Methodology

To evaluate our approach in more complex real-world scenarios, we implemented our work in a home-grown trace-driven simulator. We used synthetic traces generated by the well-known random waypoint group mobility model [13]. In this mobility model, a group of nodes select a random destination and moves to that destination at a speed uniformly distributed between zero and a maximum speed. After reaching the destination, these nodes pause for a fixed period before selecting another destination and this process is repeated until an experimental duration ends. Individual nodes within a group move slight differently but their movements are constrained such that they remain as a group. Our traces consisted of 80 participants moving within a 4750m by 4750m square area. To simulate nodes visiting different places, we divide the physical space into 4 areas and multiple partitions with $M = 6, 10,$ and 20 respectively. To simulate the changing group patterns during different time periods, for every 6000 seconds, users leave their current groups and select a new group to join in. Each trace contains information about the ID of each wireless device, and its x and y coordinates at each time slot which is set as 100 seconds.

C. Simulation Results

In the first set of simulations, we evaluate the effectiveness of our proposed distributed community detection method in terms of the detection rate of users in the same mobility group. We also compare our results with those obtained using the k-clique method.

In particular, we studied the detection rate as a function of the transmission range of the nodes using both methods. Figure 6(a) presents the final detection rate versus the transmission range of the devices.

Overall, we found that the detection rate of our method is stable over the changing transmission range and is around 80% to 85%. Our detection rate is higher than that using k-clique method when the transmission range is below 220m. As the transmission range increases, the k-clique method has a higher detection rate but with the cost of a much higher false positive rate (over 40%). This is because the number of neighbors a

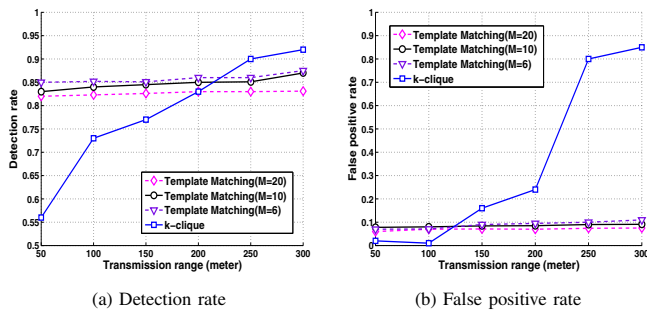


Fig. 6. Detection rate and false positive rate under different transmission range

user can hear grows with increasing transmission range and the k-clique method includes all other users that a user can hear for a sufficiently long time period in its local community group, while our method considers whether nodes belong to the same partitions during a time frame before including them in the same community group.

However, the detection rate of our proposed method is less sensitive to the changes in transmission range. Since in our method, the nodes exchange history movement information of other nodes they previously encountered, it is easier for our method to detect real community members, and thus our detection rate is not impacted much by the change of node’s transmission range.

In addition, we studied the impact of changing the number of partitions M . Our results are also presented in Figure 6(a). We observed that $M = 6$ can achieve better detection rate than $M = 10$ and 20. The reason is that when the area is divided into more partitions, the probability that the nodes in the same community have similar template matrices decreases. As a result, the detection rate decreases.

Turning to examine the false positive rate when detecting the local communities, Figure 6(b) shows that the false positive rate of our scheme remains low at about 8% across all the transmission range changes. However, the false positive rate of the k-clique method grows rapidly to over 80% as the transmission range increases. This is because the k-clique method only used encounter events to extract relationships between users and did not consider the mobility patterns of the users. Thus, when the transmission range increases, it tends to include more users in its local community list, although they are actually not from the same moving group.

We observed that $M = 20$ achieves the lowest false positive rate among the three different settings of partitions. This is because when the area is divided into more partitions, the probability that the users from different community groups having similar template matrices decreases, consequently, our method achieves more accurate community detection.

Overall, these simulation results are inline with the observations we made in our experiments. Our findings indicate that our distributed approach is feasible and effective in detecting community members.

IV. CONCLUSION

In this paper, we proposed a distributed community detection scheme, which allows each user to discover their own

communities based on local observations by using template matching. The community layer concept is introduced to differentiate local community members who appear in different time periods. Instead of only relying on encounter events between users, our method constructed template matrix based on users’ spatial and temporal mobility patterns and performed template matching to infer social relations between them. Our template matching approach reduces the overall communication cost between users.

Through both experimental and simulation studies, we show that by considering the characteristics of human movement patterns, our proposed method can effectively detect local communities within a social network. Our results demonstrate that our distributed community detection scheme can detect local communities with high detection rate and low false positive rate when comparing to the existing approaches.

REFERENCES

- [1] J.Scott, *Social Network Analysis: A Handbook*. Sage Publication Ltd, 2000.
- [2] M. Girvan and M. E. J. Newman, “Community structure in social and biological networks,” in *Proceedings of the National Academy of Sciences of the United States of America*, June 2002.
- [3] L. Tang, X. Wang, and H. Liu, “Uncovering groups via heterogeneous interaction analysis,” in *Proceedings of IEEE International Conference on Data Mining(ICDM)*, 2009.
- [4] F. Li, Y. Yang, and J.Wu, “Cpmc: An efficient proximity malware coping scheme in smartphone-based mobile networks,” in *Proceedings of IEEE Infocom*, 2010.
- [5] J. Yang, Y. Ren, Y. Chen, and M. C. Chuah, “A social community based approach for reducing the propagation of infectious diseases in healthcare,” *ACM SIGMOBILE Mobile Computing and Communications Review (ACM MC2R)*, 2010.
- [6] F. Li and J. Wu, “Localcom: A community-based epidemic forwarding scheme in disruption-tolerant networks,” in *Proceedings of IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks(SECON)*, 2009.
- [7] M. Chuah and A. Coman, “Identifying connectors and communities: understanding their impacts on the performance of a dtn publish/subscribe system,” in *Proceedings of ACM SPSNA*, 2009.
- [8] M. Chuah, “Social network aided multicast delivery scheme for human contact-based networks,” in *Proceedings of ACM First Simplex Workshop*, 2009.
- [9] B.W.Kernighan and S.Lin, “An efficient heuristic procedure for partitioning graphs,” in *Bell System Technical Journal*, vol. 49, pp. 291–307, 1970.
- [10] P. Hui, E. Yoneki, S. yan Chan, and J. Crowcroft, “Distributed community detection in delay tolerant networks,” in *Proceedings of Sigcomm Workshop MobiArch*, 2007.
- [11] B. Yang and J. Liu, “An autonomy oriented computing (aoc) approach to distributed network community mining,” in *Proceedings of the First International Conference on Self-Adaptive and Self-Organizing Systems*, 2007.
- [12] M.E.J.Newman, “Detecting community structure in networks,” *The European Physical Journal B - Condensed Matter and Complex Systems*, vol. 38, pp. 321–330, March 2004.
- [13] X. Hong, M. Gerla, G. Pei, and C. Chiang, “A group mobility model for adhoc wireless network,” in *Proceedings of ACM international workshop on modeling, analysis and simulation of wireless and mobile systems*, 1999.
- [14] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, “Locating the nodes,” *IEEE Signal Processing Magazine*, July 2005.
- [15] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. Wiley, 1973.
- [16] G. Casella and R.L.Berger, *Statistical Inference*. Duxbury Press, 1990.