Towards Group-Activities Based Community Detection

Sumeet Kumar
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
sumeetku@cs.cmu.edu

Kathleen M. Carley
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
kathleen.carley@cs.cmu.edu

UbiComp/ISWC'18 Adjunct,, October 8–12, 2018, Singapore, Singapore
ACM 978-1-4503-5966-5/18/10.
https://doi.org/10.1145/3267305.3274765

Abstract
Mobile-phone based activity recognition has been successfully applied to many useful scenarios like measuring the 'calories burnt' by a person. Unlike activities that are performed by a person alone, many activities are performed in a group-setting for example 'classroom teaching'. Because people often make friends with whom they are together, it's natural to look for communities in which people are engaged in similar physical-activities. Automated ways to learn such communities involve fusing physical-sensor-data from multiple users and hence, is a challenging problem. In this research, we measured physical-activities of seventy-two students located on two different university campuses for ten days. Using this data, we propose a model to detect communities based on similar physical-activities. Detecting such communities could be of great use e.g. it allows to invite new members who could be interested in similar activities and find those members who are in the community but are not actively engaged.

Author Keywords
Activity recognition; Mobile sensors; Community detection

ACM Classification Keywords
I.5.3 [Pattern recognition]: Clustering; J.4 [Computer applications]: Social and Behavioral Sciences

Short Description of the Experiment

Data Collection: Using a mobile application [4] that uses sensor-data to predict physical activities, we collected activities data of 72 students along with their location for ten days. The detected activities include 'Still' ,'On foot', 'On bike', 'In vehicle' and 'Unknown' along with a confidence score for each activity.

Model: We extend a Louvain modularity based community-detection algorithm to use geo-coded user-activities. The algorithm makes use of a)Spatial proximity b)Physical Activities b)Closeness in time, to find communities.

Result: Model detects communities in which community-members are co-located and engage in the same activity.

Introduction

Modern mobile phones and wearable devices have several built-in sensors (e.g., accelerometer, proximity sensor, light sensor) that allows detecting various activities of the mobile-phone users. Recently, researchers have received great success in learning users' physical activities like 'in-vehicle' or 'on-bike'. Learning such activities from mobile-phone sensors has been successfully applied to many useful scenarios like measuring the 'walking distance' and the 'calories burnt' by a person. Unlike activities that can be performed by a person alone, many activities are performed in a group setting for example 'classroom teaching'. Because people often make friends with whom they often interact, it's natural to look for communities in which people are engaged in similar physical-activities. Though such group-based activities are common, automated ways to find such communities by using mobile-phone sensor data is challenging. The problem requires fusing sensor-data from multiple users. To the best of our knowledge, no existing research addresses this problem. Finding such communities based on physical-activities have utility in growing the community as it allows to invite others interested in similar group-activities.

In this research, we propose an algorithm to detect communities in which users are engaged in same physical-activity e.g. biking. To identify such communities, we ran a user-study to collect sensor data from mobile-phones of students of a university. Using a mobile application[1] installed on Android phones, we collected location and activity data every five minutes, for ten days from seventy-two participants. Using this dataset, we find community-members involved in similar activities, where the activity, as well as the location proximity of participants are important.

[1] https://github.com/sumeetkr/OpenAlerts

In simple terms, detecting a community in a group of nodes involves finding nodes that are closer to each other compared to the rest. Many algorithms have been proposed for such tasks. However, dynamic-spatial-networks in which time and locations are involved, community detection becomes very challenging as there is no explicit notion of closeness. In a network (of users) built from phone-sensors data, the closeness between any two edges is based on three factors a) Spatial proximity (How far are two nodes based on measured GPS coordinates). b) Physical Activities (Are two nodes performing the same physical activity as determined by phone sensors) c) Time closeness (How often are two nodes physically close). We propose an algorithm based on modified Louvain modularity to incorporate these factors.

Related Work

Detecting communities is an active area of research [5, 6, 8, 10]. For detecting communities, the general understanding is that members of a community have a higher connection (interaction) probability compared to members outside the community. Researchers have tried to discover spatial-interaction communities using mobile-phone data [3, 7]. Gao et al. [7] used mobile phone dataset and a modularity function based on Grivan-Newman algorithm to find spatial-interaction communities. Like community detection, activity recognition remains an active area of research. Researchers have used accelerometers embedded in mobile devices to detect activities of users holding the phone [1, 12]. For more details of various activity detection techniques, we refer readers to a survey paper [11].

Though our work is related to other work on finding clusters using mobile data, to the best of our knowledge, no prior work has used physical-activities for learning communities.

Activity Type	Count
Still	1090175
On Foot	3495
Tilting	4105
On Bike	139
In Vehicle	3243
Unknown	6801

Table 1: Data points for each activity type.

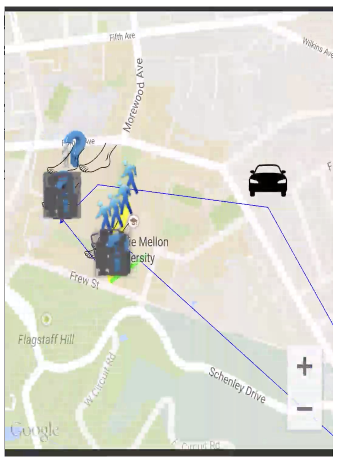

Figure 1: This image shows mobile-screen of a user participating in the data-collection effort. Various icons show different activities of a participant. In the image, the blue line indicates movement, 'leg' icon shows 'on foot' and 'car' icon shows driving and 'a person on desk' (not clearly visible because of overlaps) icon shows 'still' activity.

Data Collection

We conducted a user-study to test the effectiveness of Wireless Emergency Alerts in the United States [9]. As a part of this study, with prior users' consent, the authors collected location of participants along with their activities as predicted by Android activity detection API[2]. During the study, an Android application [4] installed on students' mobile phones sent activities and location data to our server every five minutes. The activities predicted by the Google API includes 'Still' ,'On foot', 'On bike', 'Tilting', 'In vehicle', 'Unknown' along with a confidence score for each activity (see Fig. 1). In total, the dataset has 1,107,958 records (Tab.1) and 72 unique users.

Community Detection Model

There are many algorithms to detect communities. We use 'Louvain', which is a greedy optimization approach for learning communities, as it is one of the fastest algorithms [2]. The algorithm is based on 'Louvain modularity' which is defined as:

$$M = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{w_i w_j}{2m} \right) \delta(c_i, c_j) \tag{1}$$

where $A_{i,j}$ is the edge weight between nodes i and j. w_i is sum of weights of edges linked to node i and is defined as $w_i = \sum_{k \in N_i} A_{ik}$ where N_i is the set of neighbors of node i. $2m$ is the sum of weights of all edges in the network. δ is the delta function and c_i is community assignment of node i. Note that A_{ij} is the only input for the algorithm to find community labels c_i for each node. Equation 1 assumes a simple graph with nodes and weighted edges. In our dataset, the network is not a simple graph as we have locations, activities and time dynamics.

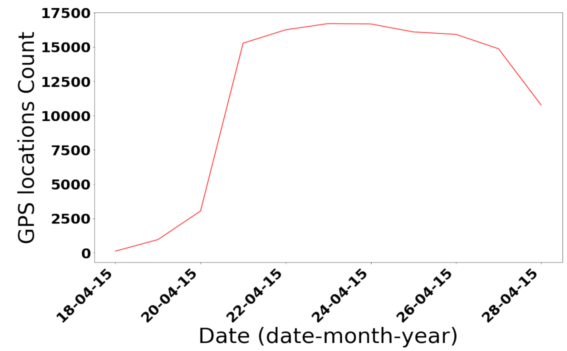

Figure 2: The plot shows the trend of GPS and user-activity data collection. The study started on 18th of April and went till 28th of April. For the first two days, limited data was collected by the study team-members themselves. On the third day, an event was organized to include other participants in the study, leading to increased data collection the day onwards.

From an intuitive understanding of how community members act, we define two people are members of the same community if a) they engage in the same activity 'often', and b) they are physically 'close-by' 'often'. As there is no explicit measure of 'often' and 'close-by', we encode them as parameters in our algorithm. Because we are interested in finding communities in which members are performing the same physical-activity, we build separate networks for each activity type. Thus, the values in $A_{ij_{activity}}$ depends on the group-activity to be modeled. For example, if a few students are engaged in a group-activity like 'classroom teaching', then they are present in the classroom during the class time duration. We can use 'Still' activity for this purpose.

We define $A_{ij_{activity}} = \sum_n \mathbb{1}_{(d(i,j) \leq d_{activity\_threshold})}$

[2] https://developers.google.com/android/reference/com/google/android/gms/location/ActivityRecognitionApi

Activity	threshold
Still	0.1
On Foot	0.1
Tilting	NA
On Bike	1
In Vehicle	1
Unknown	NA

Table 2: Distance threshold used to link two users in the algorithm. The threshold values are in km.

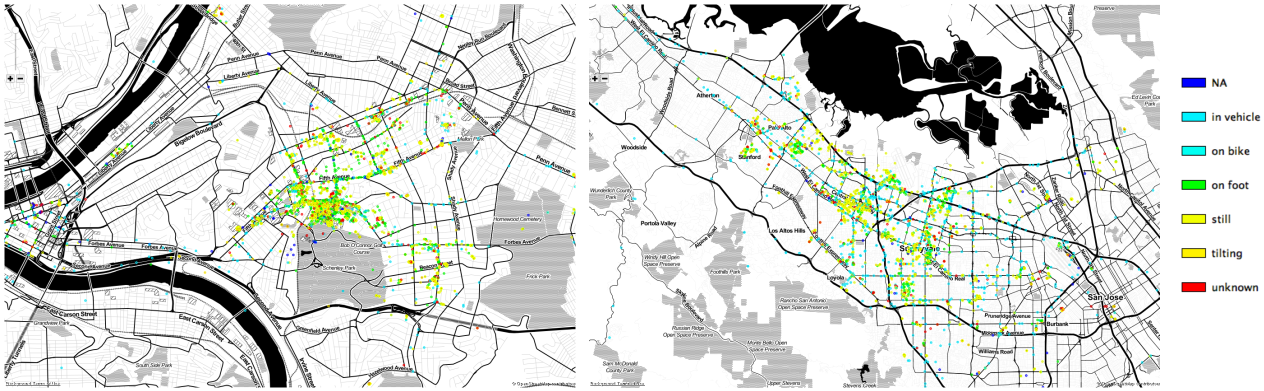

Figure 3: This figure shows an aerial view of GPS locations and corresponding physical activities (color legend on right) for users who participated in this research study. The study was conducted simultaneously at two university campuses. The image on left is for the Carnegie Mellon's Pittsburgh campus in Pennsylvania, and the image on right is for the Mountain-View campus in California.

where $d(i, j)$ is the physical distance between two users measured using their GPS coordinates and n is the number of times they are together preforming the same activity and is counted every five minutes. $d_{activity\_threshold}$ is the threshold distance below which users are assumed to be together.

Evaluation

As the participants of this study were free to move, and were never asked about their community membership, the ground-truth labels of communities are not known. Like many community detection problems, detecting community labels for this dataset is challenging and the evaluation of the model needs to be done in an unsupervised way. In such a scenario, visualizations offer a good alternative to verify predicted communities. We do have some high-level knowledge that can be verified in visualization e.g. students on different campuses should form separate communities.

To build the network structure for community-detection, as discssed earlier, we first divide the data in specific activity types. We call this network Activity-Network. In an Activity-Network, two users are connected (i.e. $A_{ij_{activity}} \neq 0$ for user i and user j) if the distance between them is less than distance threshold in Tab. 2 for activity type being used. These thresholds are based on broad assumptions about the activities that the model is trying to detect. For example, for 'classroom-teaching' two users are linked if they are 'still' and are physically located in less than '0.1' km (size of a large hall with error-margin for indoor GPS positioning). For 'On Bike' and 'In Vehicle', we have higher thresholds as people may be involved in large-distance group-activities (like trekking).

Once we have built $A_{ij_{activity}}$ for an Activity-Network, we use Eqn. 1 to find community labels. To find communities for different activity types, we apply the above steps to four

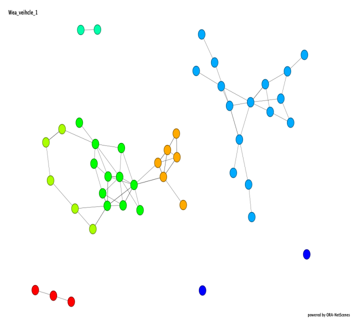

Figure 5: The plot shows the community learned for users involved in 'in vehicle' activity. Color of nodes indicate community-membership labels. Only edges with weight more than 215 are shown for clarity.

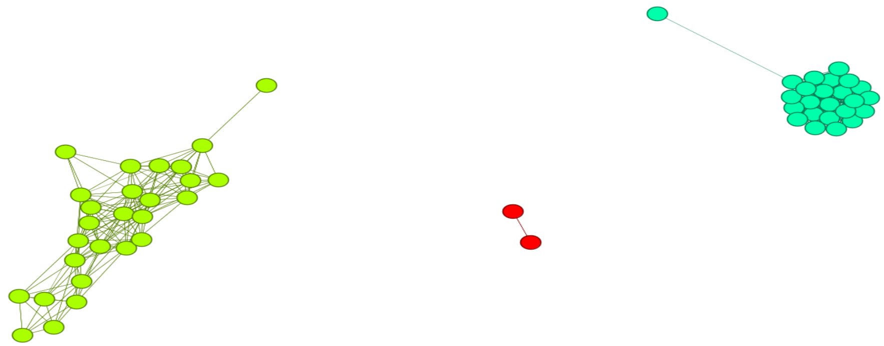

Figure 4: Figure shows community labels learned using the proposed algorithm for users who have recorded 'still' activity. Each node is a separate user. The algorithm uses activity type, distance between two phones and how often the phones are together to find groups. In this figure, We found that one of the groups represents a set of student on the silicon-valley campus of CMU and another larger group represents a small-set of students on the Pittsburgh campus of CMU.

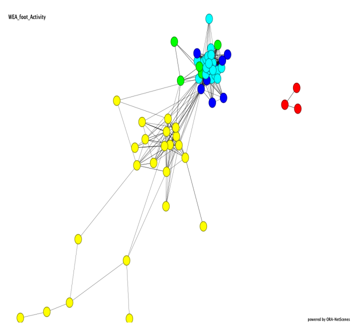

Figure 6: The plot shows the community labels for users who have 'on foot' activity. Color of nodes indicates community membership whereas edge-length (distance between nodes) indicates proximity.

activities excluding 'Unknown' and 'Tilting', as for these activities there is no clear understanding of what group-activities can be performed, and hence a 'threshold' cannot be chosen. Fig. 4, Fig. 6 and Fig. 7, Fig. 5 shows communities of users who have recorded 'Still', 'On Foot', 'On Bike' and 'In Vehicle' activity types respectively. As we have a large number of recordings for the 'Still' activity, Fig. 4 hides links with edge-weight lower than 10500 for clarity.

Limitations

Community prediction without ground-truth labels for communities limits the verifiability of the proposed algorithm. However, community detection is very often an ill posed problem as community-labeling is subjective. Accepting this limitation, we explained the various parameters that affect community labeling. Biased sampling and skewed activities distribution are the other limitations of this study. Data was mostly collected from students as study was conducted on two campuses of a university. This concern is partially mitigated by the fact that the target of study is to learn student activities. The other concern related to the skewed distribution of activities is because of limitation of phone sensors and activity recognition algorithms. Also people do not always carry their phone resulting in more wrongly labeled 'Still' activities.

Conclusion and Discussion

In this research, we presented an approach to learn community membership from phone sensor data. Our approach used activities recorded by mobile-phones (used by participants) and the geo-location of phones to learn communities in which members are engaged in the same activity. In our network, finding community membership is determined by three factors a) Spatial closeness (How close are two nodes in terms of physical distance measured using GPS locations). b) Activities involved (Are two nodes perform-

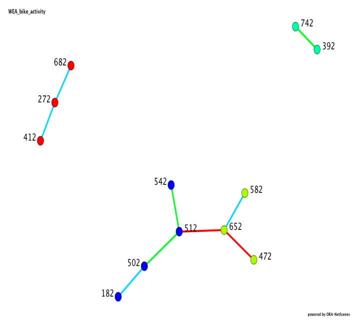

Figure 7: The plot shows the community labels for some of the users who have recorded 'on bike' type activity. Each node represents a user and the links show if they are connected. In the figure, the weight of blue links = 2, green links = 4 and red link = 8.

ing the same activity) c) Closeness in time dimension (How often are two nodes nearby). We proposed an extension of Louvain algorithm to learn community labels and finally visualize results. Our approach, though simple, captures many insights. We find student-communities on two different campuses. We could also find smaller communities engaged in biking and driving. The proposed model, though preliminary, shows that group-activities based community detection is feasible.

REFERENCES

1. Ling Bao and Stephen S Intille. 2004. Activity Recognition from User-Annotated Acceleration Data. In *International Conference on Pervasive Computing*. Springer, 1–17.
2. Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast Unfolding of Communities in Large Networks. *Journal of statistical mechanics: theory and experiment* 2008, 10 (2008), P10008.
3. Paul Expert, Tim S Evans, Vincent D Blondel, and Renaud Lambiotte. 2011. Uncovering Space-Independent Communities in Spatial Networks. *Proceedings of the National Academy of Sciences* 108, 19 (2011), 7663–7668.
4. Joao D. Falcão, Sumeet Kumar, Joel Krebs, and Hakan Erdogmus. 2018 (To-Appear). OpenAlerts: A Software System to Evaluate Smart Emergency Alerts and Notifications. In *Proceedings of the 2018 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2018 ACM International Symposium on Wearable Computers (UbiComp/ISWC'18 Adjunct)*. ACM, Singapore, Singapore. https://doi.org/10.1145/3267305.3274115
5. Santo Fortunato. 2010. Community Detection in Graphs. *Physics reports* 486, 3-5 (2010), 75–174.
6. Santo Fortunato and Darko Hric. 2016. Community Detection in Networks: A User Guide. *Physics Reports* 659 (2016), 1–44.
7. Song Gao, Yu Liu, Yaoli Wang, and Xiujun Ma. 2013. Discovering Spatial Interaction Communities from Mobile Phone D ata. *Transactions in GIS* 17, 3 (2013), 463–481.
8. Sumeet Kumar and Kathleen M. Carley. 2018. People2Vec: Learning Latent Representations of Users Using Their Social-Media Activities. In *Social, Cultural, and Behavioral Modeling*, Robert Thomson, Christopher Dancy, Ayaz Hyder, and Halil Bisgin (Eds.). Springer International Publishing, Cham, 154–163.
9. Sumeet Kumar, Hakan Erdogmus, Bob Iannucci, Martin Griss, and João Diogo Falcão. 2018. Rethinking the Future of Wireless Emergency Alerts: A Comprehensive Study of Technical and Conceptual Improvements. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 2, Article 71 (July 2018), 33 pages. DOI: http://dx.doi.org/10.1145/3214274
10. Andrea Lancichinetti and Santo Fortunato. 2009. Community Detection Algorithms: A Comparative Analysis. *Physical review E* 80, 5 (2009), 056117.
11. Oscar D Lara, Miguel A Labrador, and others. 2013. A Survey on Human Activity Recognition Using Wearable Sensors. *IEEE Communications Surveys and Tutorials* 15, 3 (2013), 1192–1209.
12. Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, and Michael L Littman. 2005. Activity Recognition from Accelerometer Data. In *Aaai*, Vol. 5. 1541–1546.