Social Interaction Discovery: A Simulated Multiagent Approach

José C. Carrasco-Jiménez, José M. Celaya-Padilla, Gilberto Montes, Ramón F. Brena, and Sigfrido Iglesias

Instituto Tecnológico y de Estudios Superiores de Monterrey, Monterrey NL, México
{jc.carrasco.phd.mty,A00811434,A00808911, ramon.brena,sigfrido}@itesm.mx

Abstract. Social interaction inference is a problem that has been of interest in the past few years. The intrinsic mobility patterns followed by humans present a number of challenges that range from interaction inference to identification of social relationships linking individuals. An intuitive approach is to focus on the similarity of mobility patterns as an indicator of possible social interaction among individuals. By recording the access points observed at each unit of time along with the strength of the signals received, individuals may be group based on similar walking patterns shared on space and time. In this paper, an implementation of a multiagent simulation of a University-like environment is tested using NetLogo and a methodology that consists of two phases: 1) Cluster Analysis and 2) Construction of Social Networks is used to discover possible interactions among individuals. The first phase consists of a number of clustering methods that are used to identify individuals that are more closely related given the characteristics that describe their mobility patterns obtained from simulated Wifi data. In the second phase, users belonging to the same cluster are linked within a social network, meaning that there is possible ongoing social interaction or tie that might link the individuals.

1 Introduction

Social interactions can be described in terms of different context variables (e.g. physical proximity, association affiliation, online profiles, etc.). Discovering physical interactions is a challenging task given the dynamic characteristic of human interactions. For this particular work, social interaction is defined as the process of influencing on each other's mobility behaviours due to physical proximity.

Analyzing interaction information from a simulated world environment allows us to control the environment and as a consequence to validate the results obtained, therefore a NetLogo simulation was designed. In this simulation environment, it is possible to configure a Univeristy-like environment with features such as access points, and software agents behaving like real people. The user interface allows us to modify the conditions of the environment in order to study grouping patterns.

By analyzing mobility patterns followed by individuals, we can gain valuable information about the context of users' social world [1]. Social networks are constructed

J.A. Carrasco-Ochoa et al. (Eds.): MCPR 2013, LNCS 7914, pp. 294–303, 2013.

from the empirical evidence of pattern similarity discovered by applying cluster analysis to the Wifi context data. The groups of people that are linked in the social network, are individuals who are most likely to be linked by a social interaction.

Clustering algorithms have been chosen to segment individuals that might be linked by a probable interaction due to their effectiveness in identifying distinct groups of individuals based on their characteristics (e.g. the signals received from the different access points that surround the context) [2]. In other words, objects that are grouped in the same cluster, are more closely related to one another than objects that are grouped in different clusters [3].

The k-means algorithm is a widely used method for data segmentation or grouping when a number of k groups is known; it assigns an observation to a cluster with the nearest mean. For this research project, we tested the k-means implementation supplied in R software environment (as described by Hartigan and Wong) [4] with $k = 4$, since the experiments were initialized with 4 groups. In the k-medoids clustering algorithm the centers for the clusters are restricted to be one of the observations that were assigned to the cluster, i.e., it is more robust in the presence of outliers. The 'fpc'R package [5] has an implementation of an enhanced version called pamk(), which calls pam and clara algorithms for the partitioning around medoids clustering method, as introduced by Kaufman and Rouseeuw [5].This function does not require a k value, instead, it uses the CLARA algorithm to do the partitioning around medoids to estimate an appropriate k value. The DBSCAN clustering algorithm, as introduced by Ester et al. [5], is based on the notion of density of data points. The idea of this algorithm is that a group has to contain at least a minimum number of points (minpoints) within a given radius (eps); in other words, the density of the neighborhood has to exceed some threshold [6].

This paper is organized as follows. Section 2 describes a number of related works followed by Section 3, which describes the simulation implemented in NetLogo platform . In Section 4, there is an overview of the methodology proposed to discover social group interaction followed by experimental results described in Section 5. A brief conclusion is given in Section 6, and future work is described in Section 7.

2 Related Work

The field of computing sciences has given birth to a vast number of applications that allow us to understand human dynamics. Although research in this field is very recent, some approaches to understand complex human dynamics have been made. In a study led by Eagle et al. [7], they show that it is possible to infer friendships based solely on observational data exploiting Bluetooth traces, cellular data, phone data (e.g. SMS, call records), as sources of social information. On the other hand, Mokhtar et al. [8] propose a middleware service that aims to combine both social and physical interactions in order to identify encounters, i.e. interactions between individuals. Encounters are logged from Bluetooth radio conectivity. In a research work [1], Cranshaw et al. introduce an analysis of GPS location for contextual features of human location trail data, to elicit the existance of a physical social network and to analyze the context of the

social interactions. In [9,10], Xu et al. collected and analyzed Wifi data to explore the relationship between physical proximity and social links between individuals. In this work, the physical position of the individuals was estimated from Wifi signal strength and MAC address of nearby Wifi access points. Then, proximity encounters between individuals were computed based on the distance threshold and duration threshold. The positioning of individuals was accomplished by performing a site survey that involves recording the Wifi signal strengths and access points of all floors in the building on the floor map [10]. Nevertheless, this approach requires the construction of radio maps a priori, which makes it unsuitable for highly dynamic scenarios.

3 The Simulation

A university-like environment was designed using NetLogo platform [11]. This multiagent simulation environment offers the tools to simulate, among many other types of environments, social interactions among software agents simulating individuals and access points.

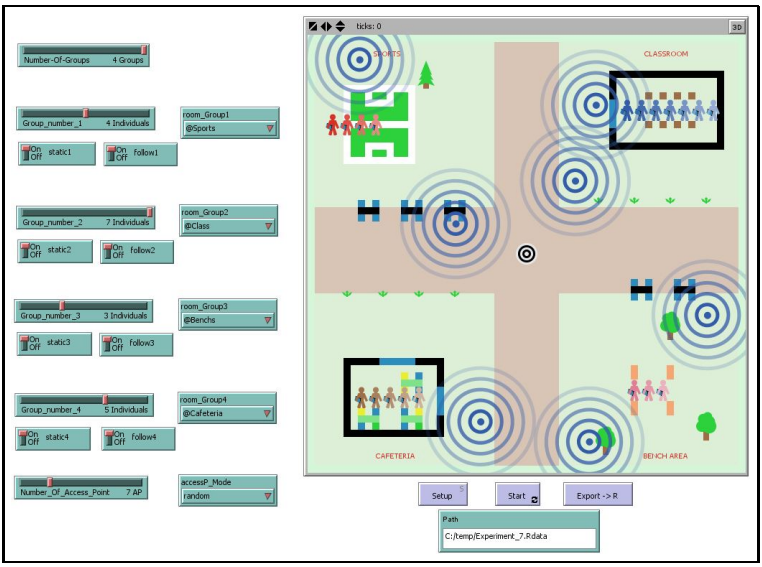
A control panel, as shown in Fig. 1, allows to modify initial settings in order to simulate different social scenarios with different mobility patterns. Among the different configurations, the user can choose the number of initial groups of individuals. On the other hand, mobility pattern generation can be adjusted for each of the initial groups, allowing the user to choose the number of individuals per group, the starting location for each group, as well as the walking patterns. The walking patterns are defined for each initial group, allowing the group to be either static or dynamic. Those groups tha are dynamic can either follow their own path (i.e. random walking), or can follow a leader (i.e. group walking).

Besides the group behaviors described previously, access point configurations can also be adjusted. The simulation control panel also allows the user to choose the number of access points (up to 25) that sorround the environment. In order to generate a more realistic set of Wifi data, the access points generate data following the theoretical signal propagation model shown in (1) and described in [12]. However, in real life scenarios, signals are constantly affected by external factors such as interference from signals, temperature, obstacles and other factors, causing signal degradation. In order to compensate for such factors of signal degradation, Gaussian noise was added to the data generated by the simulated access points, since Bose et al. [13] show that when large samples of RSS are collected over time, the noise is normally distributed among the samples.

$$RSS = -(10nlog_{10}d + A) \tag{1}$$

where:

- n: signal propagation constant (n = 3)
- d: distance from sender.
- A: received signal strength at a distance of one meter.



(a)

Fig. 1. Environment and Control Panel

4 Methodology

4.1 Social Group Discovery

Our approach discovers social group interactions in 2 phases:

1. **Cluster analysis:** in this phase, simulated Wifi data is segmented using the three clustering algorithms described in Section 1. The clustering is based on the Received Signal Strength (RSS) observed by the closest access points (APs) at each unit of time.
2. **Construction of social networks:** the groups of individuals obtained from the cluster analysis can be seen as a social network, where the individuals who belong to the same cluster are linked among themselves forming a social network. Similar movement patterns can be associated to possible physical interaction given the physical proximity through time.

4.2 Data Sets

The data sets analyzed for this project are collections of simulated context data that includes Wifi readings. The data collected contains four types of information with the following signature: wifi{User, Timestamp, MacAddress, RSS}. This signature contains information about the individual that collects the data, the time at which it was collected, the unique ID of the access points observed, and the signal strengths of each observed access point respectively.

An intuitive assumption made in this project is that if two users are close to each other at the same time, i.e. same timestamp, they will observe similar access points with the same MacAddresses and similar RSS (Signal Strength) readings. In other words, if a group of users walk together, they will observe the same movement patterns, detecting the same access points with similar signal strengths as time passes.

4.3 Experimental Settings

For the study of social interaction based on clustering methods, we performed 16 experiments with different initial configurations. All the experiments initially contain four groups. Each group was located at one of the four available initial locations. The numbers of individuals per group was regularly changed from one experiment to another. Table 1 includes the set of experiments taken for this study. In this table, Static refers to the number of groups (i.e. initial number of groups of individuals) that were stationary. Follow-Leader refers to the number of groups that moved together (i.e. following a leader), and Agents refers to the number of simulated individuals that were part of the experiment.

Table 1. Experiment configuration details

	Movement Patterns			Data Set
Experiment	Static	Follow-Leader	Agents	Instances
1	4	0	19	3800
2	4	0	14	2800
3	2	2	18	3600
4	0	4	13	2600
5	2	0	21	4200
6	0	0	12	2400
7	0	0	8	1600
8	2	2	18	3600
9	1	3	18	3600
10	0	0	4	800
11	0	0	5	1000
12	0	4	8	1600
13	0	0	4	800
14	0	0	8	1600
15	0	4	15	3000
16	0	1	15	3000

The initial setup of the simulations show different conditions under which social interactions might take place in real life. Since every single group of people simulated in the social environment has its own selective control, it is possible to change the behavior of each group. For example: it is possible to have one group with no motion at all, while the individuals of a second group may move randomly around, and the other two groups could follow the walking patterns of their leader.

5 Experimental Results

The experimental phase consists of a set of 16 experimental configurations simulating different conditions for social interaction, as described in Sect. 4. For all the experiments, three clustering algorithms were tested: k-means, k-medoids, and DBSCAN. The results obtained for the three algorithms are presented in Table 2.

The k-means algorithm [4] was tested with a parameter $k = 4$ for all the experiments since all of them consist in initial settings of 4 groups. The error rate, computed as shown in (2), was used as a measure of performance of the algorithm. The number of misclassified individuals was computed from visual inspection of the movement patterns observed on the individuals, i.e. we used observational validation. Execution time was also considered in order to see how the algorithm behaved as the number of instances varied. As a result, the error rate for the k-means algorithm, as shown in Table 2, was observed to be 0.246 in average. On the other hand, the execution time for this algorithm was 4.669s in average.

$$ER = \epsilon/\omega \tag{2}$$

where:

- ϵ is the number of misclassified individuals
- ω is the total number of individuals.

k-medoids algorithm implementation [5] showed an average error rate of 0.209 and an execution time of 4.902s, as presented in Table 2. Although k-medoids algorithms require the number of clusters (i.e. k value), the algorithm implementation provided in *R* programming environment [5] provides an enhanced way of selecting the number of clusters automatically based on the input data.

The DBSCAN algorithm implemented in [5] was also tested on the same datasets. This algorithm requires two parameters, the minimum number of points in a cluster, which was set to 2 (minpoints = 2), and the radius around which to search for data points to be added to the same cluster, which was obtained empirically for most of the data sets to be 0.22 (eps = 0.22). For this algorithm, the average error rate was 0.336 and the execution time was 4.642s, as we can see in Table 2.

Table 2 shows that all three algorithms have a similar average execution time. The average error rate shows that k-means and k-medoids algorithms outperform the density-based DBSCAN algorithm. The k-medoids shows a better average performance over the other two algorithms, although the k-means performs almost as well as the k-medoids. The k-means works well for controlled environments, but when the context is very dynamic, as the study case presented in this work, the k value has to be adjusted. In other words, there is a need for a method that estimates the appropriate k value that is adjusted to the model representing the environment.

Among the advantages of the k-medoids algorithm implementation [5] we find the fact that it does not require a predefined k value, and that it outperforms the other two algorithms when two characteristics of the context apply: 1) the environment is dynamic, 2) the distribution of Wifi access point signals cover most of the environment under study. It is important to note that when there are many uncovered regions, users

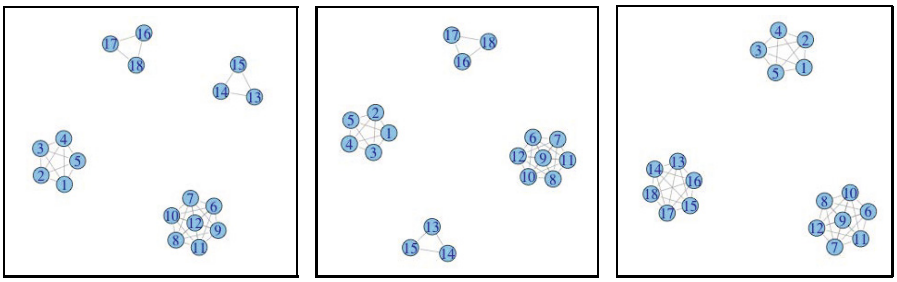
Table 2. Experimental Results

	K-Means			K-Medoids			DBSCAN		
Exp. #	k val.	Error Rate	Exec. Time (s)	k val.	Error Rate	Exec. Time (s)	k val.	Error Rate	Exec. Time (s)
1	4	0	5.152	4	0	4.998	4	0	4.759
2	4	0	3.871	4	0	4.041	4	0	3.885
3	4	0	4.758	4	0	4.917	2	0.270	4.749
4	4	0	3.863	4	0	3.811	4	0	3.953
5	4	0.333	5.510	4	0	5.739	2	0.285	5.392
6	4	0.250	3.790	4	0.333	3.942	4	0.250	3.737
7	4	0.250	2.825	8	0.875	2.877	1	1	2.939
8	4	0	4.889	4	0	5.332	3	0.166	4.714
9	4	0.277	4.780	4	0	5.077	2	0.333	4.813
10	4	0.277	4.898	4	0	5.230	2	0.333	4.973
11	4	0	1.866	1	1	2.057	1	1	1.922
12	4	0	2.354	4	0	6.107	2	0.5	2.366
13	4	0	6.084	2	0.875	3.453	4	0	6.016
14	4	0	4.554	4	0	5.936	1	1	5.120
15	4	0	6.361	4	0.266	7.423	4	0.25	6.114
16	4	0.066	9.158	4	0	7.497	4	0	8.821
Avg. Error Rate:		0.246	–		0.209	–		0.336	–
Avg. Exec. Time (s):		–	4.669		–	4.902		–	4.642

can be grouped in the same cluster even when they are far away from each other. The reason for this is that users are grouped based on the observed access points, and when no access point is in sight, a value of 0 is assigned to the user at that point in time. As a consequence, when two individuals have long trajectories in areas without any signal coverage, they will report high similarities even when they are far away from each other.

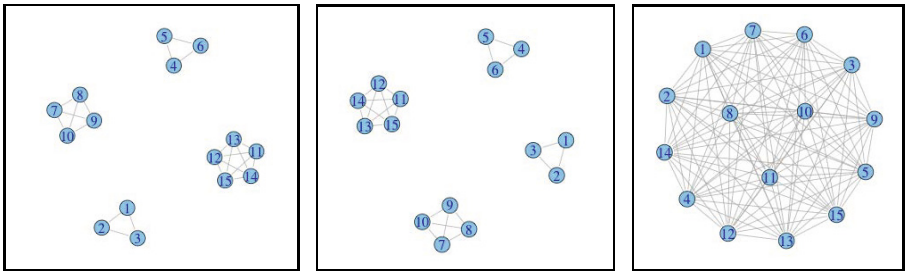
Some of the benefits of grouping individuals based on observed access points and RSS include robustness to noise in Wifi readings, and the ability to apply the methodology in real time, since the results are not affected if the distribution of access points is modified as oppose to methods that require radio maps to be constructed a priori. The results were validated using observation of trajectory traces followed by the human-like agents in the simulation.

Figure 2 shows the resulting social network after running the three algorithms, mentioned above, on the data collected in experiment number 8, described in Table 1. Experiment 8 is a controlled environment where 2 of the groups are static and 2 groups move randomly but following the traces of a leader. As a consequence there are 4 groups perfectly identifiable and separated from each other as shown in Fig. 2 (a) and (b). The image shows the social network constructed using the "igraph" R package [14], based on the clusters identified by each of the three clustering algorithms revised in this project. As it has been mentioned in the analysis of the results obtained for each clustering method, the k-medoids and k-means perform well in average, both having similar results. Figure 2(c) shows that the DBSCAN algorithm has a similar set of clusters than those identified by the other two algorithms. The only difference is that two of



(a) (b) (c)

Fig. 2. Experiment #8 a) K-means. b) K-medoids. c) DBSCAN.

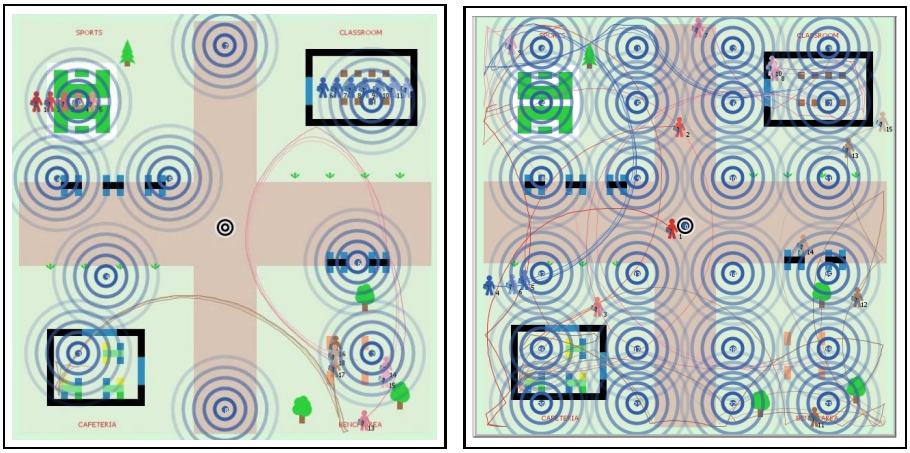


(a) (b) (c)

Fig. 3. Experiment #16 a) K-means. b) K-medoids. c) DBSCAN.

the clusters are merged into one, which means that it misclassified some of the individuals. For this specific scenario, DBSCAN's grouping is not far from the known groups. The scenario describing the patterns followed by the individuals using the settings of experiment 8 can be seen in Fig. 4 (a).

On the other hand, Fig. 3 shows the social networks constructed from the results of experiment 16 shown in Table 1. Experiment 16 consists of 4 non-fixed random groups, one of which follows a leader's trace. For this particular scenario, both k-means and k-medoids were able to find the same clusters while DBSCAN was unable to find the clusters at all, as it is shown in Fig. 3 (c). From visual inspection, we conclude that the groups of individuals found by k-means and k-medoids are the most likely groups given the movement patterns followed by the groups of individuals. Even though this experiment consists of 3 groups walking randomly, 4 groups were detected due to the amount of time they were close to each other, which is the time it took the individuals to leave the cafeteria and classroom. The scenario of experiment 16 is shown in Fig. 4 (b).



(a) (b)

Fig. 4. Experimental Scenarios a) Experiment #8. b) Experiment #16.

6 Conclusions and Future Work

The NetLogo platform contains suitable characteristics to model and simulate mobility patterns in a bidimensional space. Cluster analysis is an efficient way to group individuals based on similar mobility patterns described by contextual Wifi data. Out of the three algorithms we tested in this project, k-means and k-medoids performed fairly well on the data sets obtained from the NetLogo simulations. One of the advantages of k-medoids over k-means, as implemented in [5], is that k-medoids can estimate the number of clusters while k-means requires the user to adjust the parameter manually. The resulting clusters of individuals represent discovered social groups that are likely to be linked by a social interaction. We were also able to identify the k-medoids as the best option for grouping individuals of similar movement patterns in the simulated environment when two conditions applied: 1) dynamic environment 2) high Wifi signal coverage. Also, it is possible to tell that the clustering methodology is robust to the presence of Wifi signal noise because it does not depend on a predefined distribution of access points. The signal noise robustness should make this clustering approach suitable for applications that require real-time analysis.

Real world mobility patterns can be hard to analyze due to the difficulty in controlling the context as well as the restriction on the amount of experiments. From this perspective, having a controlled environment allow us to analyze the power of the proposed methodology and to learn about the expected behavior of the clustering strategies prior to test it in a real and more complex scenario.

As future work, we propose to apply the methodology followed in this paper on data collected on real scenarios. A deeper analysis of unsupervised learning algorithms can also be tested on Wifi data to discover group interactions (e.g. FRiS-Tax, Hierarchical, etc.). Other types of measures to estimate the error rate can also be investigated, furthermore different type of sensors (e.g. Bluetooth, GPS, Magnetic sensors) and online

social network information (e.g. Facebook, Twitter, Foursquare) can be incorporated to improve the cluster accuracy.

References

1. Cranshaw, J., Toch, E., Hong, J., Kittur, A., Sadeh, N.: Bridging the gap between physical location and online social networks. In: Proceedings of the 12th ACM International Conference on Ubiquitous Computing, Ubicomp 2010, pp. 119–128. ACM, New York (2010)
2. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer Series in Statistics. Springer New York Inc., New York (2001)
3. Witten, I., Frank, E., Hall, M.: Data Mining: Practical Machine Learning Tools and Techniques: Practical Machine Learning Tools and Techniques. The Morgan Kaufmann Series in Data Management Systems. Elsevier Science (2011)
4. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2012) ISBN 3-900051-07-0
5. Hennig, C.: fpc: Flexible procedures for clustering, R package version 2.1-4 (2012)
6. Patwary, M.A., Palsetia, D., Agrawal, A., Liao, W.K., Manne, F., Choudhary, A.: A new scalable parallel dbscan algorithm using the disjoint-set data structure. In: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, SC 2012, pp. 62:1–62:11. IEEE Computer Society Press, Los Alamitos (2012)
7. Eagle, N., Pentland, A., Lazer, D.: From the cover: Inferring friendship network structure by using mobile phone data. Proceedings of The National Academy of Sciences 106, 15274–15278 (2009)
8. Mokhtar, S.B., McNamara, L., Capra, L.: A middleware service for pervasive social networking. In: Proceedings of the International Workshop on Middleware for Pervasive Mobile and Embedded Computing, M-PAC 2009, pp. 2:1–2:6. ACM, New York (2009)
9. Xu, B., Chin, A., Wang, H., Wang, H., Zhang, L.: Social linking and physical proximity in a mobile location-based service. In: Proceedings of the 1st International Workshop on Mobile Location-Based Service, MLBS 2011, pp. 99–108. ACM, New York (2011)
10. Zhu, L., Chin, A., Zhang, K., Xu, W., Wang, H., Zhang, L.: Managing workplace resources in office environments through ephemeral social networks. In: Yu, Z., Liscano, R., Chen, G., Zhang, D., Zhou, X. (eds.) UIC 2010. LNCS, vol. 6406, pp. 665–679. Springer, Heidelberg (2010)
11. Tisue, S., Wilensky, U.: Netlogo: A simple environment for modeling complexity. In: International Conference on Complex Systems, pp. 16–21 (2004)
12. CC2431 Location Engine
13. Bose, A., Foh, C.H.: A practical path loss model for indoor wifi positioning enhancement. In: 2007 6th International Conference on Information, Communications Signal Processing, pp. 1–5 (December 2007)
14. Csardi, G., Nepusz, T.: The igraph software package for complex network research. Inter. Journal Complex Systems 1695 (2006)