Platys Social: Relating Shared Places and Private Social Circles

Pradeep K. Murukannaiah and Munindar P. Singh

North Carolina State University

Raleigh, NC 27695-8206, USA

{pmuruka, singh}@ncsu.edu

Abstract

In the traditional, offline world, users naturally group their connections—the people they know—into social circles, assigning them different priorities. Social circles and priorities among connections facilitate intelligent collaboration by helping a user determine how to interact with whom. Social circles can be valuable in online applications. However, existing online approaches don't readily support such grouping: they either require a user to manually tag connections or rely purely on broad-brush acquaintanceship between connections.

Platys Social is a novel approach that learns a user's social circles and prioritizes his connections by bringing together contextual information and user interactions. Platys Social runs incrementally and can execute on a resource-limited mobile device. It can potentially avoid moving a user's private information to a remote site.

We exercised Platys Social in a study of six users over ten weeks. We found that Platys Social effectively learned the users' social circles.

Keywords: J.9 Mobile Applications < J Computer Applications, K.4 Computers and Society < K Computing Milieu

1 Introduction

Users today increasingly participate in online social interactions, especially media-driven interactions that may have no offline correlate.

Although online interactions can be rewarding for users, they open up new challenges.

Control: how flexible can a user be in choosing with whom he interacts?

Cognitive overload: can a user prioritize interactions and information so as to reduce his cognitive burden?

Data privacy: can we support the above without storing a user's private information outside of his personal devices?

A major source of the first two of these challenges is that online relationships today exhibit a flat structure, or as Deresiewicz [1] puts it, everyone in the online world is a faux friend. By contrast, in traditional (offline) settings, users implicitly categorize their connections into multiple *social circles*, such as family, classmates, colleagues, friends from different cities, and so on. Further, a user may have different priorities among their connections. Recognizing a user's social circles and priorities can benefit several applications.

Social network sites, supporting (i) friendship suggestions, (ii) fine-grained privacy policies, and (iii) enhanced social search by ranking paths to a target individual [2].

Email, facilitating email triage by prioritizing incoming messages.

Social virtual worlds, mapping the offline social circles of a user to his avatar in a virtual world [3].

Social networking applications increasingly support users structuring their connections ("groups" on Facebook; "circles" on Google+). Manually creating social circles and prioritizing their member connections, especially as they change over time, is tedious and time consuming [4]. Lampinen and colleagues [5] describe an extensive study that highlights the challenges in privacy for users and the inadequacy of using static groupings of connections. Grouping is not effective without prioritization of a group's members. Thus, we need automatic approaches for recognizing social circles and priorities.

Community detection [6] is a widely used approach for identifying groups of users in social networks. Informally, a *community* in a network is a set of nodes with dense edges within the set, and sparse edges to the rest of the network. However, existing social networks merely include acquaintance relationships; communities in such networks are coarser than social circles. For example, a user's college connections may all fall into one community if they have sufficiently many mutual acquaintances, despite reflecting many social circles. Further, detecting communities in a social network presupposes knowing the global network structure, which makes the approach infeasible unless users provide their private data to a third party.

We propose a novel approach called *Platys Social* that addresses the above challenges through the following characteristics.

Automatically learning and maintaining the social circles of a user and the priorities among the connections in each circle.

Exploiting contextual information and offline user interactions for learning,

yielding social circles that are more meaningful than the communities based merely on acquaintance relationships.

Privacy preserving by employing only the information locally available to a user and not storing private information outside of a personal device.

Let us define two key terms.

- A user's *connection* is anyone the user recognizes and relates to in some context. A connection of a user may have multiple identifiers—offline or online—such as appearance, name, phone number, email ID, Bluetooth device address, and so on.
- A *social circle* of a user is a set of connections the user perceives as a logical group. A user's social circles are ego-centric in that they are defined from a user's perspective, not necessarily from the connections' perspectives.

2 Social Circles and Connection Priorities

Platys Social seeks to address the following main questions.

What is a natural basis for logically grouping a user's connections?

We propose the notion of a *place-based identity*. A place, contrasted with geospatial position, is a conceptual construct with high salience for user actions and interactions [7]. A typical user visits several logical places and shares such places with others. Examples of *shared places* include home, workplaces, classrooms, friends' homes, restaurants, and so on. It is interesting that a user can identify most of his connections in conjunction with such shared places. For instance, family members can be typically identified with one's home, classmates with classrooms, coworkers with one's workplace, and so on.

How can we prioritize the connections in a social circle?

We propose to do so based on the frequency of interactions. Platys Social categorizes a user's connections into two main categories as follows.

- A *strong connection* is one with whom the user interacts frequently.
- A *weak connection* is one with whom the user interacts infrequently.

2.1 Platys

Our informal answers above presuppose a framework for gathering user information such as the places he visits, his connections, interactions, and so on. *Platys* (`http://sites.google.com/site/platysproject/`) is an active effort in building a framework for (i) efficiently sensing the low-level information about a user such as his position, environment, and actions, (ii) learning the high-level concepts such as the places and social circles from the sensed information, and (iii) supporting intelligent applications that exploit place and social circles.

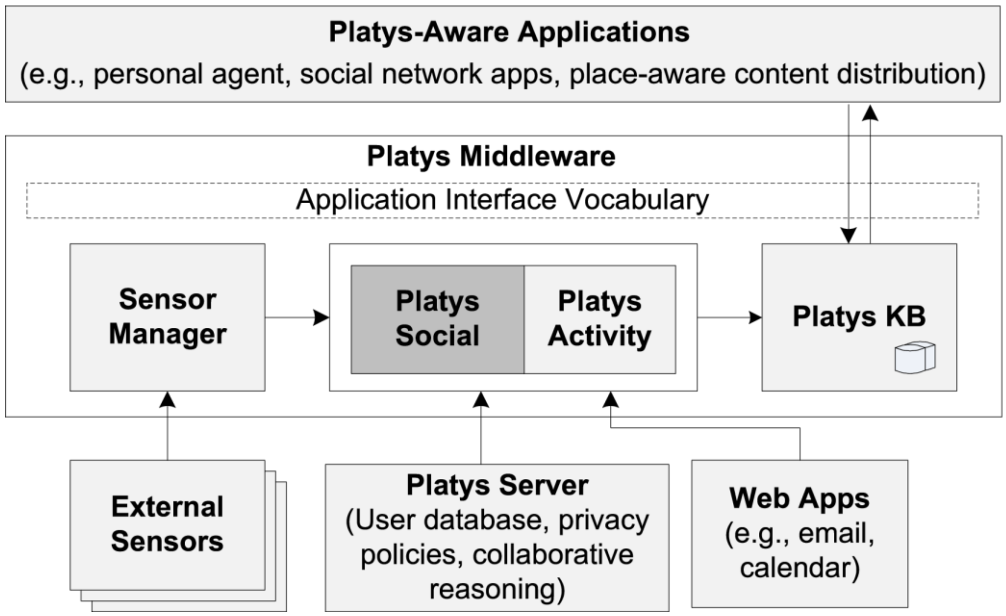

Figure 1: The Platys architecture, highlighting the focus of this paper.

Figure 1 shows the Platys architecture as consisting of three major components: sensors, middleware, and applications. In principle, all these components can be installed on a user's personal device. Smart phones are our devices of choice: they come with a variety of sensors; are almost always carried by a user; and are increasingly powerful.

2.2 Platys Social

Platys Social is a component of the Platys middleware. Figure 2 shows its structure.

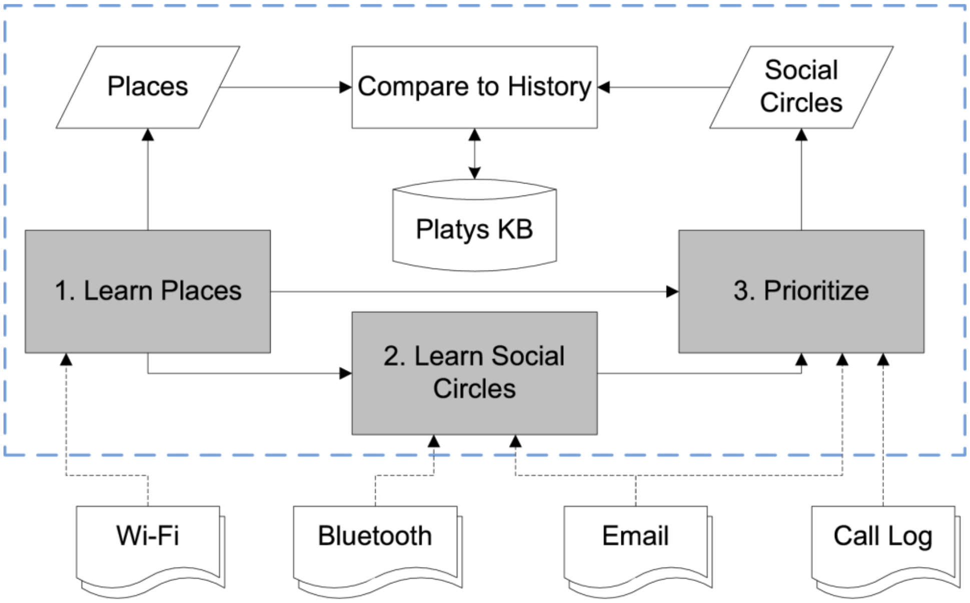

Figure 2: The architecture of Platys Social, highlighting its learning modules.

2.2.1 Place Learning

In order to group a user's connections using place-based identities, we should first identify the socially significant places a user visits. Platys Social identifies such places by exploiting the prevalence of Wi-Fi access points (APs) in modern urban environments. The sensor manager in a Platys-enabled device continually scans Wi-Fi channels. For each scan, the middleware logs a timestamped vector of APs, where each AP is associated with a (i) unique address (*BSSID*), (ii) user-defined name (*SSID*), and (iii) Received Signal Strength Indicator (*RSSI*).

Considering each scan event in the Wi-Fi AP log as a data point, we perform cluster analysis to discover significant places. Clustering presupposes a distance measure between any two data points, which we define as the so-called cosine distance

$$cos_{\delta}(i,j)=1-\frac{rssi_i \odot rssi_j}{\|rssi_i\|\times\|rssi_j\|}, \tag{1}$$

where $rssi_i$ and $rssi_j$ are vectors of $RSSI$ values for scan events i and j of lengths $\|rssi_i\|$ and $\|rssi_j\|$, respectively, and $rssi_i \odot rssi_j$ represents their dot product.

A challenge we face in clustering APs is that the number of clusters in the data (number of places a user has visited) is unknown a priori. We build a dendrogram of APs using Matlab's hierarchical clustering package (`http://www.mathworks.com/products/statistics/`). We use the distance that maximizes the silhouette coefficient (which combines cluster separation and cohesion [8]) to cut the dendrogram. Once the clusters are computed, we ignore clusters with APs of low RSSI values as noise. Each remaining cluster corresponds to a place and is associated with (i) a set of consistent APs, (ii) a cumulative RSSI value for each AP, and (iii) a set of timestamps.

2.2.2 Social Circle Learning

Once we identify the places a user visits, how may we identify people in those places? Bluetooth appears to be a promising technology—it has a short range and most users have mobile devices equipped with it. Thus, the Platys middleware scans for Bluetooth devices continually and records a Bluetooth device log similar to the above Wi-Fi AP log. To learn the social circle corresponding to a place, we group all Bluetooth devices found in the intervals corresponding to the timestamps associated with the place. This leads to social circles that contain (i) a set of Bluetooth devices, (ii) a cumulative RSSI value for each Bluetooth device, and (iii) a place.

The above technique relies on users to keep their Bluetooth devices in discoverable mode, which is not a popular practice, despite Bluetooth technology being increasingly secure and energy efficient. To overcome this lack of Bluetooth data, we combine place-based grouping with email-based and call-based grouping. Our intuition is that just as we group a user's connections based on shared places, so can we group them based on co-occurrence in email threads and phone calls (such as in a conference call).

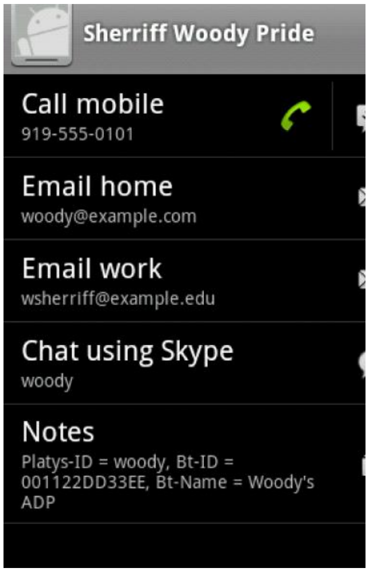

Figure 3: Details of a user's connection maintained by Platys.

Let $C_u = \{c_1 \ldots c_n\}$ be the set of all connections of a user u. As Figure 3 illustrates, Platys aggregates multiple identifiers for each connection of a user in an address book. We define a weight w_{ij} for each pair of connections $c_i, c_j \in C_u$ as the weighted average of the frequency of (i) the co-occurrence of the two connections in a place, (ii) the co-occurrence of the two connections in an email thread, and (iii) the co-occurrence of the two connections in a phone call. Further, we construct a *connection co-occurrence graph*, which is an undirected and weighted graph whose vertex set is C_u, and there exists an edge between c_i and c_j if and only if $w_{ij} > 0$.

Unlike an acquaintance network, the connection co-occurrence graph is based on real interactions and contextual information. In addition, such a graph can be fully constructed using only the local information available to a user. A user's social circles can be learned by identifying communities in his connection co-occurrence graph. We apply the clique percolation method (CPM) [9] to identify the communities. An advantage of CPM is that it finds overlapping communities and the social circles of a user are likely to overlap. CPM works by identifying k-cliques in the graph and constructing communities as a union of adjacent k-cliques (two k-cliques are adjacent if they have $k-1$ nodes in common). As CPM suggests, we choose k=4 and lower the weight threshold until the largest community found is twice the size of the second largest community.

2.2.3 Prioritizing Connections

Once the social circles are learned, we prioritize a user's connections within each social circle on the basis of the user's interactions with them. We consider the following types of interaction between the user and each of his connections.

- Face-to-face interactions estimated by the cumulative Bluetooth proximity of a connection to the user in the place corresponding to a social circle.
- Email interactions measured by the number of email exchanges between a connection and the user.
- Phone interactions measured by the number of phone calls between a connection and the user.

Accordingly, for each connection, we define an *interaction weight* as a weighted sum of face-to-face, email, and phone interactions. Platys Social designates a connection as *strong* if its interaction weight exceeds a threshold, and as *weak* otherwise. The threshold can be set by plotting interaction weights of all connections in a social circle and choosing a point that separates a few from the many. Such a threshold reflects the intuition that social circles have a few strong connections but many weak connections.

2.2.4 Maintaining Social Circles

Platys Social employs an incremental approach to learning to keep a user's social circles up-to-date. Typically, each week it learns (separately for each user) places and social circles, and prioritizes connections. It then compares the learned places and social circles with a history of places and social circles, which can be used to track how the social circles evolve. An advantage of this incremental approach is that each execution of Platys Social involves amounts of data feasible for analysis on a resource-limited personal device.

3 User Study

We conducted a study of six users, all graduate students in their twenties and thirties, who used a Platys-enabled Android phone as their primary phone for ten weeks. The Platys middleware ran as a background service on the phone and recorded Bluetooth and Wi-Fi scans every five minutes. In addition, the middleware could access the user's email and call logs.

In order to acquire the ground truth, we asked each user to maintain a *place calendar* by updating a calendar with all socially significant places they visited each day (home, classrooms, workplaces, restaurants, and so on). Towards the end of the study, each user identified social circles corresponding to the places in their place calendar. In addition, each user prioritized connections in each social circle as strong or weak. The concepts of social circles, strong and weak connections were informally described to the users to capture their natural intuitions.

The learning process was offloaded to a server due to the lack of data analysis software for Android. There are obvious privacy concerns because the server was accessible to the researchers of this study. However, in practical deployment, the server can be thought of as hosting users' personal agents. It need not be a server of the conventional social network site that collects user information.

3.1 Place Learning

In order to evaluate the place learning, we compared the places learned by Platys Social and the places reported by the user in the place calendar. We define two places to be similar if the overlap between the timestamps associated with the two places is greater than a predefined threshold. The Jaccard similarity [8] between the learned and the reported places is

$$PlaceSimilarity = \frac{|Learnedplaces \cap Reportedplaces|}{|Learnedplaces \cup Reportedplaces|}. \qquad (2)$$

Figure 4 shows the *PlaceSimilarity* for each user. The plot indicates that Platys Social is effective in learning places with similarity averaging nearly 85%. We further investigated the places not common between the learned and the reported, and uncovered interesting reasons for such errors: (i) sometimes, the users reported two learned places such as two shops in a mall as one place (ii) some learned places were pass-by places that the users didn't identify as significant, and (iii) some reported places with poor Wi-Fi infrastructure were discarded as noise by Platys Social.

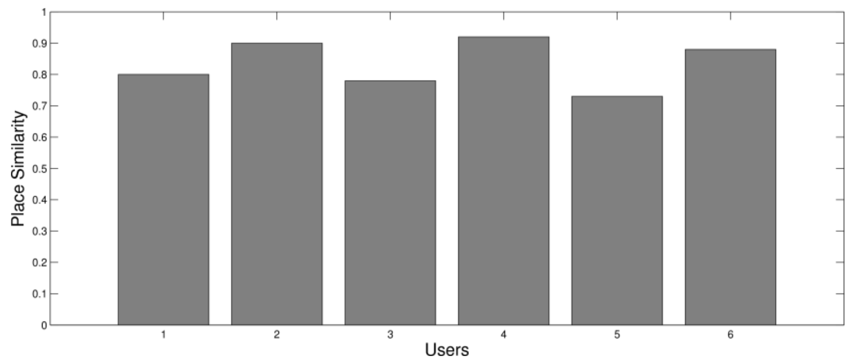

Figure 4: Similarity between places learned by Platys Social and manually identified by users.

3.2 Social Circle Learning

Similar to places, we also evaluated the similarity between the learned and reported social circles. We define *CircleSimilarity* by replacing places with social circles in Equation 2.

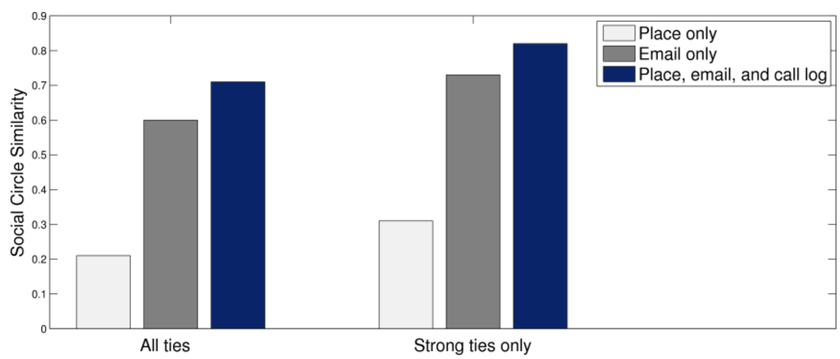

Figure 5: Similarity between social circles learned by Platys Social and the social circles reported by users.

Figure 5 compares *CircleSimilarity* for different criteria, averaged across all users. The email history alone is more effective than using the place information alone. Although we claim that the place-based identities is an elegant mechanism, the reason for its relative ineffectiveness is that many users today are not Bluetooth-discoverable, and the social circles learned contained fewer users than expected. However, it is interesting that combining place and email information enhanced the effectiveness of social circle learning. Looking into the details, we found that users did not use email to interact with all of their social circles. For example, not surprisingly, some users had insignificant email interactions with their family members despite meeting them regularly, as identified from the place information. Our data didn't include any conference calls (unsurprising outside of business settings). Thus we couldn't evaluate the effectiveness of phone calls in learning social circles.

Finally, we analyzed only the strong ties learned by Platys Social and reported by the users. We found that the heuristic used by Platys Social (with place, email, and call logs) learned the strong ties of the users more effectively than all ties. Although Platys Social successfully learned most user-reported weak ties, it learned unreported ties as well. We conjecture that such false positives correspond to *familiar strangers* [10], whom we encounter often albeit without any direct interactions (suitably extending the notion to email threads).

4 Future Research and Conclusions

Platys Social opens up several avenues for future research. First, a Wi-Fi cluster does not quite capture a logical place that Platys envisions. For example, a user may perceive two *seminar halls* to be the same place, despite the two being different rooms. On the contrary, a user may view a coffee shop as two places, both a *caffeine fix* and a *meeting place*. We propose that recognizing user activities can serve to enhance the notion of place. For example, what makes two seminar halls in distant corners of a campus the same place is that similar activities take place in both. A key challenge in activity recognition is to bring together information from various sources such as sensors, browsing data, application usage, and so on [11]. In addition, understanding a user's mobility patterns [12] can provide useful hints for activity and place recognition. For example, a user's mobility pattern in a *theme park* might be quite different from that in a *poster session*.

Second, the *strength* of a connection (tie) classically incorporates the amount of time spent interacting, emotional intensity, intimacy, and reciprocal services that characterize the tie [13]. However, Platys Social captures only the frequency of interactions. It remains to be seen if frequency is an effective surrogate for the other factors and what easy-to-compute attribute may supplement frequency. Frequency alone proves inadequate in some settings. For example, a next-door neighbor may be incorrectly prioritized as a strong connection because of frequent face-to-face interactions. Platys Social can potentially benefit from technologies such as the Sociometer [14], which attempt to model face-to-face interactions.

Third, Platys Social requires manual effort to aggregate multiple identifiers of a user's connections. Performing this task automatically and in a privacy preserving manner is a significant challenge and is essential for wider adoption.

Fourth, Platys Social exploits only the relationship from places to social circles. The implications of the relationship between social circles and places remains to be studied. For example, knowing that two of a user's significant places have social circles with same members is an indication that the two places might be logically the same place.

Fifth, the ten-week duration of our study precludes an effective examination of the changes to users' social circles. Future enhancements to Platys Social and studies over longer durations would help us address the above challenges.

The ideas demonstrated in Platys Social could naturally be combined with a variety of software applications that involve interaction among people: these include not only email, chat, and social networking, but also ad hoc business processes. Platys Social can enhance user experience by helping structure and prioritize not just information flow but a user's actions generally in a manner that is socially salient for that user. Further, because Platys Social takes an ego-centric stance, it is naturally privacy preserving. When implemented on a user's personal device, it could avoid many of the risks associated with sharing information via a third party.

5 Acknowledgments

Thanks to the National Science Foundation for partial support under grant 0910868, and to the reviewers for helpful comments.

References

[1] William Deresiewicz. Faux friendship. *The Chronicle of Higher Education*, December 2009.

[2] Sudheendra Hangal, Diana MacLean, Monica S. Lam, and Jeffrey Heer. All friends are not equal: Using weights in social graphs to improve search. In *Proc. Wkshp. Social Network Mining and Analysis*, pages 1–7. ACM, July 2010.

[3] Adel Hendaoui, Moez Limayem, and Craig W. Thompson. 3D social virtual worlds: Research issues and challenges. *IEEE Internet Computing*, 12(1):88–92, January 2008.

[4] Maayan Roth, Asaf Ben-David, David Deutscher, Guy Flysher, Ilan Horn, Ari Leichtberg, Naty Leiser, Yossi Matias, and Ron Merom. Suggesting friends

using the implicit social graph. In *Proc. 16th International Conf. Knowledge Discovery and Data Mining*, pages 233–242, Washington, DC, July 2010. ACM.

[5] Airi Lampinen, Vilma Lehtinen, Asko Lehmuskallio, and Sakari Tamminen. We’re in it together: Interpersonal management of disclosure in social network services. In *Proc. International Conf. Human Factors in Computing Systems (CHI)*, pages 3217–3226. ACM, 2011.

[6] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3–5):75–174, February 2010.

[7] Steve R. Harrison and Paul Dourish. Re-place-ing space: The roles of place and space in collaborative systems. In *Proc. 6th Conf. Computer Supported Cooperative Work (CSCW)*, pages 67–76. ACM, 1996.

[8] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Pearson, Boston, 2006.

[9] Gergely Palla, Imre Derényi, Illés J. Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, April 2005.

[10] Elizabeth Paulos and Eric Goodman. The familiar stranger: Anxiety, comfort, and play in public places. In *Proc. SIGCHI Conf. Human Factors in Computing Systems*, pages 223–230. ACM, April 2004.

[11] Amit Sheth. Computing for human experience: Semantics-empowered sensors, services, and social computing on the ubiquitous web. *IEEE Internet Computing*, 14(1):88–91, January 2010.

[12] Injong Rhee, Minsu Shin, Seongik Hong, Kyunghan Lee, and Song Chong. On the levy-walk nature of human mobility: Do humans walk like monkeys? In *Proc. 27th IEEE Conf. Computer Communications (INFOCOM)*, pages 924–932. IEEE, April 2008.

[13] Mark S. Granovetter. The strength of weak ties. *American Journal of Sociology*, 78(6):1360–1380, May 1973.

[14] Tanzeem Choudhury and Alex Pentland. Sensing and modeling human networks using the Sociometer. In *Proc. IEEE International Symp. Wearable Computers*, pages 216–222, White Plains, NY, October 2003. IEEE Computer Society.

Pradeep K. Murukannaiah is a PhD student in Computer Science at North Carolina State University. He received a B.E. in Information Science and Engineering from University Visveswaraya College of Engineering, Bangalore. Murukannaiah’s research interests include machine learning, social computing, and semantic web.

Munindar P. Singh is a Professor in Computer Science at North Carolina State University. His research interests include multiagent systems and service-oriented computing with a special interest in the study of interactions among autonomous parties. Singh was the Editor-in-Chief of the *IEEE Internet Computing* from 1999 to 2002. He is a member of the editorial boards of *IEEE Internet Computing*, *Autonomous Agents and Multiagent Systems*, and *Web Semantics*.