On Managing Social Data for Enabling Socially-Aware Applications and Services

Paul Anderson, Nicolas Kourtellis, Joshua Finnis, Adriana Iamnitchi
Department of Computer Science and Engineering
University of South Florida, Tampa, FL
[paanders, nkourtel, jfinnis, anda]@cse.usf.edu

ABSTRACT

Applications and services that take advantage of social data usually infer social relationships using information produced only within their own context. We propose to combine social information from multiple sources into a directed and weighted social multigraph in order to enable novel socially-aware applications and services. We present GeoS, our early prototype of a geo-social data management service which implements a representative set of social inferences. We demonstrate GeoS' potential for social applications on a collection of social data that combines collocation information and Facebook friendship declarations from 100 students.

Categories and Subject Descriptors

E.1 [**Data**]: Data Structures—*graphs and networks*;
H.3.4 [**Information Systems**]: Online Information Services—*data sharing*

General Terms

Design, Management, Measurement

Keywords

social graph, social data management, socially-aware applications

1. INTRODUCTION

The popularity of user-generated content tools exposes an unprecedented amount of information about the interaction between Internet users. In particular, two recent classes of Internet applications are revealing much social information: popular online social networks (such as Facebook or LinkedIn); and widely-adopted collaborative tools (such as CiteULike or Delicious) which provide a rich information fabric through tags, annotations, and organization of text. In addition, the ubiquitous GPS and Bluetooth-capable mobile devices can generate yet another layer of social data through location and collocation information.

SNS'10, April 13, 2010, Paris, France.

Social information has been used with encouraging results in applications to infer preferences [12], trust between individuals [21], and incentives for resource sharing [19]. However, such applications typically use social information collected from their own social contexts (such as address books in email clients for improving spam filtering [17]) or from one external source of social information (such as using object-centric networks from Delicious for improving keyword searches [30]). Using social information from only one source is subject to higher bootstrap costs (a user needs to have used that application long enough to have built sufficient social history), is naturally restricted to the application domain (e.g., email), and can be misleading.

In particular, declared relationships, such as friend lists on online social networks, suffer from two limitations. First, the hidden incentives to have many "friends" lead to declaring contacts with little support in terms of trust, common interests, shared objectives, or other such manifestations of (real) social relationships [10]. Second, all contacts appear equal, thus invested with the same level of social closeness.

Richer and more nuanced social information from diverse social sources can lead to a more accurate inference of trust, incentives for resource sharing, or shared interest; can help identify social contexts (e.g., personal or professional); and can enable novel classes of social applications and socially-aware software infrastructures. Examples of social applications include silencing personal phone calls when in professional meetings; creating personalized evacuation routes in natural disaster situations, such that friends or family members end up together; and authorizing trusted friends to "free ride" when they claim low-battery status in a mobile BitTorrent swarm. Socially-aware distributed systems infrastructures can also be enabled: for example, a cloud infrastructure could be built ad-hoc from the personal computers of users who are connected by social solidarity [4] on a specific topic. This "personal" cloud could be used for local computational emergencies, such as search and rescue efforts through computerized image processing or hurricane path prediction for a particular geographical area.

The thesis of this paper is that the wealth of social information exposed from multiple sources can be efficiently mined for the design of distributed computing infrastructures to facilitate improved performance for traditional applications and to enable a whole new set of novel applications. At the core of exploiting social information in the software stack is a service capable of collecting, maintaining, and processing geo-social data.

This paper proposes for discussion some of the challenges

raised by the design of a geo-social data management service that (i) collects social information from multiple social sensors deployed as applications running on the web or on mobile devices; and (ii) represents the social information in such a way as to allow diverse, complex inferences on user social environments. It also describes GeoS (Section 3), our early prototype of such a geo-social data management service, together with a representative set of social inferences that can be used by diverse socially-aware applications and services. As a proof of concept, we evaluate some of these social inference functions on a set of novel social data that combine collocation information and Facebook friendship declarations from students on a medium-sized urban university campus (Section 4).

An important issue for such a service is the privacy protection of social data. Current commercial systems (Google, Facebook, and others) use a centralized architecture that provides them access to an alarming amount of private data (which is at the core of their business model). For a geo-social data management service that collects information from multiple, diverse sources, privacy guarantees become paramount. Without providing definitive solutions, we discuss this and other issues in Section 5.

2. MOTIVATING APPLICATIONS AND SERVICES

We believe that richer social information can enable new classes of applications, running either on mobile devices, or as services on desktops or networked systems.

Social information can disambiguate contexts inferred only from location information. One class of such applications is context-based filters. For example, based on social information, an application could infer a professional context even outside the known work location (for example, a business lunch) and silence personal calls. Another mobile application might wait until an irregular course seminar ended to deliver and display text messages. To allow for such examples, a geo-social data management service needs to store social information collected from multiple sources. The user's location can be helpful to identify context (e.g., restaurant or classroom); collocation information is necessary to help refine the inferred context (e.g., people nearby); but also the type of social ties with the collocated users is important, for example to identify if the lunch is with co-workers (thus, potentially a business lunch) or friends. The type of social tie can be inferred from various sources, such as job-oriented social networks (LinkedIn) or a listing on the company web site. In pervasive computing applications, this approach can complement solutions employed for context-awareness [25] that use multiple virtual and physical sensors [5, 11, 26].

A more complex service generates a personalized emergency evacuation route that directs family and friends towards the same route in order to easily meet when out of danger. This service requires access to the location of a user, a way to infer their socially close relations, and the current locations of these relations.

A location-independent example is a battery-aware BitTorrent application [15] on mobile devices, where a user may rely on social incentives to be allowed to temporarily "free ride" the system when low on battery. The users participating in the same torrent would call a social inference function in the social data management service to determine if the social strength with the free rider is high enough to contribute on their behalf.

Users with health issues often carry around devices to monitor their vital signs. In case of emergency, such a device could alert close friends or family on the patient's status and location. Relatively static data (e.g., family or physician) could be inserted by the user.

Complex content-sharing policies can be enabled through the usage of diverse sources of social information. For example, a content-sharing application could allow users to share videos with each other based on social connections and interests even if not directly connected.

These are just some examples of the benefits of aggregating social (and sometimes location) data from multiple sources and inferring social knowledge from this information. Rich social information can be used in many other scenarios, such as identifying self-serving rings of voters in systems like eBay or Reddit; inferring incentives for storage sharing outside a limited application, as mentioned in [27]; inferring social incentives that would reduce churn in a peer-to-peer network; minimizing false positives in identifying sybil attacks; and many others.

3. GEOS: A GEO-SOCIAL DATA MANAGEMENT SERVICE

Two types of social ties are typically considered [8]. First, *object-centric*, identified through the use of similar resources or participation in common activities. For example, tagging the same items in collaborative tagging communities such as Delicious or CiteULike; repeatedly being part of the same BitTorrent swarms; commenting on other persons' blog or photos; or sharing interest with other users through accessing the same files for work or entertainment. Second, *people-centric*, defined as declared social relationships (e.g., online social networks), declared membership to groups (e.g., networks in Facebook), or collocation information.

We designed and prototyped GeoS, a geo-social data management service responsible for building and maintaining a social graph based on information submitted by different social sensors that can report object- or person-centric social ties. GeoS also implements a set of social inference functions that provide applications and services access to social knowledge. An overview of the GeoS architecture is demonstrated in Figure 1.

3.1 GeoS Social Graph

GeoS represents a social network as a directed and weighted multigraph, where multiple edges can connect two users and each edge is labeled by an activity in common to the two users. These activities can be collected from various *social sensors* that observe: online social networks (Facebook and LinkedIn); website interactions, such as subscribing to blogs (Twitter and LiveJournal) and commenting on material at media-sharing sites (Flickr and YouTube); mobile phone interactions, recorded as collocation via GPS or Bluetooth or as logs from phone communications; email and IM conversations that are mined for keywords in order to extract the topic discussed; etc. These sensors can be applications that are installed on the user's mobile device or PC, or even on the user profile of an online social service. They aggregate and analyze history on a user's interactions with other users and report to GeoS when significant and persistent interac-

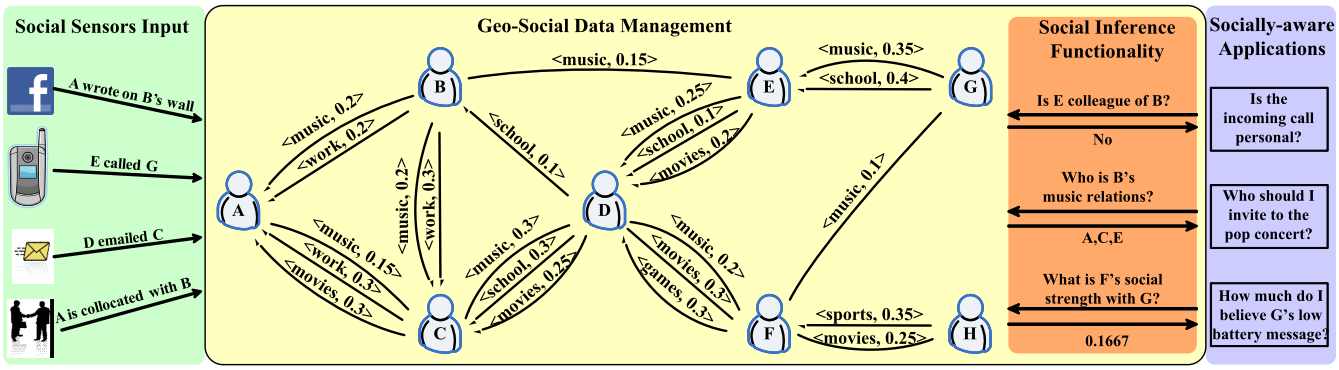

Figure 1: An overview of the GeoS architecture, illustrating the social input given by various social sensors to GeoS, the representation of social ties between users in a directed, weighted, labeled multigraph and a set of applications accessing social knowledge through GeoS provided functionalities.

tions are detected. Activities are identified by a label (e.g., "hiking") and an adaptive weight that specifies the intensity of the activity between two users (e.g., a function of the phone conversation duration). The social sensor is responsible for providing the label based on the social context of the activity and for calculating the weight of the edge reported to GeoS.

The directed, weighted, labeled multigraph data structure presented in this paper is a novel approach for storing social information. The multigraph structure is rarely used in sociology, likely due to the many limitations inherent to survey-based data collection [2]. When it is used, the multigraph is not weighted, but only records boolean multiplex relations [16]. An example of this novel social graph, as created and maintained by GeoS, is illustrated in Figure 1. Note that some of the edges are not reciprocated (e.g., user G to F for music) because activities may not be reciprocal (for example, email or writing on somebody's Facebook wall) or because a user's social sensor considered the interaction negligible based on the user activity.

Online social networks and the abundance of social data available can be mined to infer weights associated with relationships. These weights may represent the level of interaction between users, as in [6], or the strength of object-centric relationships, such as how friends are related in their movie and music preferences [18]. Transforming social data into a measurable weight that accurately represents the social data is one of the main tasks of social sensors.

We chose to represent the graph as directed because of the well-accepted result in sociology that "ties are usually asymmetrically reciprocal" [28]. This design decision, though, also enhances security by limiting the potential effects of spamming, network poisoning or manipulation. For example, a unidirectional intense communication (e.g., via wall postings on Facebook profiles) from Alice to Bob will not affect the strength of the relationship Bob has with Alice.

Other data, such as the time of the last recorded interaction for each pair of users, and the latest location of each user is also stored in GeoS.

3.2 Geo-Social Data Maintenance

The information sent by social sensors specifies the users in the relationship, the type of activity they did together, and an incremental weight. We assume the social sensors may perform analysis on the collected data before sending an input. For example, collocation social sensors can differentiate between routine encounters (such as weekly meetings with familiar strangers at the gym) and interactions between friends [7, 26]. By making GeoS oblivious to the type of social activity reported, we allow extensibility with the types of social sensors: specifically, new social sensors can report to GeoS without requiring modifications to GeoS.

However, maintaining the weight associated with a social tie is the responsibility of GeoS. This weight is dynamic: it increases at the reoccurrence of the corresponding activity and decays over time through an aging process. This aging process should be activity specific, but it should also reflect the user's social habits and interests: users who are less socially active and users with a great number of friends should have their relationships age slower. In the current prototype we implemented a simple aging function which reduces an edge's weight by 10% for every week the two users do not interact. In this way, the connection never completely disappears and the aging happens slowly.

3.3 Social Inference Functions

GeoS currently implements a set of basic social inference functions, presented in the following. More complex social inferences can be built on top of this set.

The function *friend_test(ego, userB, α, x)* is a boolean function that checks whether *ego* is directly connected to *userB* in the social graph by an edge with label α and with a minimum weight of x. A mobile phone application can use this function, for example, to determine whether an incoming call from a coworker should be let through or silenced on weekends.

The function *top_friends(ego, α, n)* returns the top n users in the social graph (ordered by decreasing weights) that are directly connected to *ego* by an edge with label α. An application can use this function, for example, to invite users highly connected with *ego* to share content related to activity α.

The function *neighborhood(ego, α, x, radius)* returns the set of users in *ego*'s neighborhood of *radius* who are connected through social ties of a label α and weight larger than x. The function performs a breadth-first search on the social graph. An application that silences a user's cell phone near coworkers uses this function to infer *ego*'s work neighborhood in the social graph even if not directly connected to the user (i.e., the phone will still be silenced if on a business

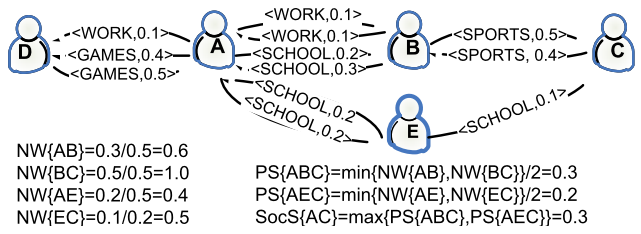

Figure 2: A social strength calculation example.

lunch with a regional manager that the user has not met before).

An extension of the neighborhood function is *proximity (ego, α, x, radius, min_distance)* which filters the results of the neighborhood inference based on physical distance to *ego*. After the location information is gathered for *ego* and the set of users is returned by neighborhood, the function returns the set of users who are within *min_distance* to *ego*. The distance between two coordinates is calculated using the spherical law of cosines. The application silencing the phone uses this function to infer the list of collocated coworkers.

We also implemented a more complex social inference function, called social strength. The function *social_strength(ego, userB)* returns a real number between 0 and 1 that quantifies the social strength between *ego* and *userB* from *ego*'s perspective. This value is normalized to *ego*'s social ties to ensure that the social strength is less sensitive to the social activity of the users. As shown in Figure 2, the normalized weight from A to its adjacent neighbor B (NW{AB}) is the sum of all the weights of the edges from A to B (aggregating over all types of interactions between A and B) divided by the largest of all the sums of weights going from user A to one of its neighbors (D). The path strength from A to C through B (PS{ABC}) is the lowest of all the NW on that path, divided by the length of the path. Finally, the social strength from user A to user C (SocS{AC}) is the largest path strength from A to C.

The social strength request provides users with a quantitative measure on social solidarity [4] based on the frequency and intensity of their social interactions. Such a function could be used, for example, to estimate social incentives for resource sharing, such as storage or hosting a service for another user or a community, or the battery-aware BitTorrent example presented above. Social strength can be applied to dyads that may not be directly connected. Moreover, even if a direct tie exists, an indirect path (through common friends) may be stronger [3]. We limit the length of the indirect path that connects two users to 2, using a well-accepted result in sociology known as the "horizon of observability": Friendkin [9] shows that people know about persons up to two social hops away, even though they do not interact directly with them. We interpret this as an indication that it is unlikely for users who are connected by a shortest path of length 3 or longer to have a non-zero social strength between them.

4. EARLY EXPERIENCES WITH REAL SOCIAL DATA

A GeoS prototype is implemented as a multithreaded service written in Python, using the NetworkX module [22] for handling the directed, multigraph data structure. It also uses Python dictionaries for holding data such as edge weights, the sums of the weight of all the edges between two users, the latest coordinates of each user, and the time of the last interaction between users.

We show experimentally the benefits of using the social mulitgraph structure maintained by GeoS. For this, we used real traces collected from two social sensors applied to a student population. The social data consist of one-month of Bluetooth collocation data and Facebook friend lists for a set of 100 students at the New Jersey Institute of Technology (NJIT). The data collection took place on this medium-sized urban campus, and the subjects were representative of the various majors offered on campus, with 75% undergraduates and 28% women.

Similar to the Reality Mining traces [7], smart phones were distributed to students and an application quietly recorded the Bluetooth addresses of nearby devices and periodically transmitted them to a server. Given that the sample size was small compared to the university population (104 randomly selected out of 9000 students) and that many students are commuters, the trace data is sparse. For example, about half the subjects reported less than 24 hours of data for the entire month, and only 17% of the scans detected other Bluetooth devices in proximity. The typical user provided a few hours of data per day, especially during the weekdays. The same set of subjects installed a Facebook application to participate in a survey, and gave permission for their friend lists to be collected when they installed this application.

4.1 Creating the Social Multigraph

The recorded social data was separated into three categories: pairs of users who were collocated for a total of at least 45 minutes (of which at least 15 minutes were consecutive) during the study ($CL.45$), pairs of users who were collocated for a total of at least 90 minutes (of which at least 15 minutes were consecutive) during the study ($CL.90$), and pairs of users who declared themselves friends on Facebook (FB). Each pair in the $CL.90$ data was also in the $CL.45$ data. There were 87 users in the FB data, 94 in the $CL.45$ data, and 81 in the $CL.90$ data. 81 of the 100 subjects appear in both Facebook and collocation data.

We created a multigraph based on the NJIT data where edges were labeled "facebook" with weight 0.1 for the FB data and "collocation" with weight 0.1 (or 0.2) for the $CL.45$ (or $CL.90$) data.

4.2 Testing GeoS Functionalities

To show how a social application could exploit the GeoS functionalities and the use of the weighted multigraph we consider a "photo-sharing" application scenario: The collocation of students in a summer camp is reported by Bluetooth sensors and recorded in GeoS as a "summer-camp"-labeled edge with different weights, corresponding to 0.1 for every 45 minutes of collocation. A Facebook sensor reports if two students are friends on Facebook.

Alice is interested in any photos taken by a fellow summer camp attendee that may contain her or her friends. She decides to use our "photo-sharing" application, which utilizes the collocation and Facebook edges maintained in GeoS, to find and include other attendees' photos in an album (depending on the attendees' privacy rules).

First, the application submits a neighborhood request for a given number of social hops using particular types of edges

Social Hops	1	2	3	4	5	6
FB	3(0.7)	11(0.5)	23	32	36	37
CL.45	8(0.7)	38(0.3)	72	84	87	87
CL.90	3(0.9)	8(0.6)	17	29	40	49
CL.45 or FB	9(0.6)	43(0.3)	78	90	92	92
CL.45 and FB	1(1.0)	5(0.6)	17	27	31	32
CL.90 or FB	5(0.8)	17(0.5)	34	49	60	67
CL.90 and FB	1(1.0)	5(0.8)	17	27	31	32

Table 1: Average number of contacts returned by a neighborhood request submitted by *ego* (for various combinations of edges and weights). Average social strength between *ego* and the contacts within two hops are in parentheses.

and combinations ("summer-camp" with a weight of 0.1 or 0.2 and "facebook" with a weight of 0.1). This results in a set of attendees connected directly or indirectly with Alice (Table 1). Alice decides to focus the search to find only the summer camp attendees she or her friends spent the most time with and are also friends with her or her friends on Facebook. To fulfill this search, the application performs a neighborhood inference request with a maximum of 2 social hops, returning 5 attendees. From these 5 attendees, the application requests to share their photos with Alice in her summer camp album.

Similarly, Alice receives requests to share her photos with other summer camp attendees that use the same application. She only wants to share her photos with the attendees that are the most socially connected to her. Thus, she specifies a high threshold on the social strength that connects her to the requestors. If Bob was the summer camp photographer, for example, he could specify a low threshold to allow more attendees to access his photos.

5. SUMMARY AND DISCUSSIONS

The position of this paper is that the recent dramatic exposure of social knowledge can be exploited for a variety of novel socially-aware applications and services. For this to happen we need a service that collects and manages social and location data reported from a diverse set of social sensors. We demonstrate the functionality of such a service on a number of hypothetical scenarios that require social inferences for serving users or applications. Our prototype, GeoS, currently supports a useful set of social inference functions and has been used by two socially-aware applications: silencing phone calls from relations that are not from the same type of social context as the called user's current context; and a socially-aware BitTorrent application on mobile devices [15].

What we did not present due to space limitations is our design of the decentralized GeoS, where social information is stored on distributed nodes and is available only to trusted users and services. In practice, our GeoS prototype runs on a DHT-based peer-to-peer infrastructure and uses ACLs and encryption to protect access to social data. We are currently running experiments on PlanetLab to evaluate the costs of social inferences running over a social graph distributed over Internet-connected resources, the effect of mapping social information of socially-connected users onto the same peer, and the costs of socially-aware ACL maintenance.

But other open research problems are introduced by the approach proposed in this paper, of which we briefly discuss two: the feasibility of developing accurate social sensors and more subtle aspects of privacy.

Even if not named or designed as such, a multitude of social sensors have already been deployed for recording both person-centric social ties (such as Facebook applications that collect social graphs [29, 10], co-location, and interactivity sensing) and object-centric ties (evident from system characterizations on YouTube [20], collaborative tagging sites [24], file sharing in scientific collaborations [14], and many others). The challenge raised in this paper is in accurately estimating weights and, in some cases, activity labels. Even if sociology studies could find a generalized formula for estimating weights of an activity, it must also adapt to the individual user's social behavior, since people can perceive the same interaction with different intensities. Accurate activity labels will likely rely on such things as physical locations being labeled, the ability to correctly determine the topic of text-based interactions, and speech recognition software that can run on a mobile device.

Perhaps the most important challenge introduced by aggregating information from many social sensors is preserving user privacy. On one hand, using and enforcing user policies on what data can be collected (particularly from mobile devices) is necessary for adoption. On the other hand, there are more subtle ways in which personal information can be exposed even if data is encrypted and well protected: aggregate or indirect measures (such as node degree in a graph) can be correlated with out-of-band information and lead to insights in a user's private information [13].

Along the same line there is another challenge: protecting privacy seen as contextual integrity [23]. Contextual integrity is a measure of how closely the flow of personal information conforms to context-relative informational norms. For example, medical information transmitted outside the health-care context to friends or co-workers can be seen as a violation of privacy. Using policy languages to formalize this concept (such as the one presented in [1]) and enforcing user-specified policies is a promising approach. In particular, this requires distinguishing between the different social contexts of a user (at a minimum, professional vs. personal) and prohibiting information from one social context to be used for a different context, even if the information itself is not directly exposed. The representation of the social network we chose, namely labeled activities, may be helpful: labels permit restricting the social ties that can be accessed and exploited by applications according to the context in which they run.

6. ACKNOWLEDGMENTS

This research was supported by the National Science Foundation under Grant No. CNS-0831785. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

7. REFERENCES

[1] A. Barth, A. Datta, J. C. Mitchell, and H. Nissenbaum. Privacy and contextual integrity: Framework and applications. In *IEEE Symposium on Security and Privacy*, pages 184–198, 2006.

[2] H. R. Bernard, P. Killworth, D. Kronenfeld, and L. Sailer. The problem of informant accuracy: The validity of retrospective data. *Annual Review of Anthropology*, 13:495–517, 1984.

[3] Y. Bian. Bringing strong ties back in: Indirect ties, network bridges, and job searches in china. *American Sociological Review*, 62(3), 1997.

[4] M. Bourgeois and N. E. Friedkin. The distant core: social solidarity, social distance and interpersonal ties in core-periphery structures. *Social Networks*, 23:245–260, October 2001.

[5] H. Chen, T. Finin, and A. Joshi. An ontology for context-aware pervasive computing environments. *Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review*, 18:197–207, 2003.

[6] H. Chun, H. Kwak, Y.-H. Eom, Y.-Y. Ahn, S. Moon, and H. Jeong. Comparison of online social relations in volume vs. interaction: a case study of Cyworld. In *Proceedings of the 8th conference on Internet Measurement*, 2008.

[7] N. Eagle and A. S. Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268, May 2006.

[8] J. Engeström. Why some social network services work and others don't: the case for object- centered sociality. http://bit.ly/engest05, 2005.

[9] N. E. Friedkin. Horizons of observability and limits of informal control in organizations. *Social Forces*, 62(1):57–77, 1983.

[10] S. A. Golder, D. Wilkinson, and B. A. Huberman. Rhythms of social interaction: Messaging within a massive online network. In *3rd Intl. Conference on Communities and Technologies*, 2007.

[11] T. Gu, H. K. Pung, and D. Q. Zhang. A service-oriented middleware for building context-aware services. *J. Netw. Comput. Appl.*, 28(1):1–18, 2005.

[12] K. P. Gummadi, A. Mislove, and P. Druschel. Exploiting social networks for internet search. In *Proc. 5th Workshop on Hot Topics in Networks*, pages 79–84, Irvine, CA, 2006.

[13] M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Weis. Resisting structural re-identification in anonymized social networks. *Proceedings VLDB Endowment*, 1(1):102–114, 2008.

[14] A. Iamnitchi, M. Ripeanu, E. Santos-Neto, and I. Foster. The small world of file sharing. *IEEE Trans. Parallel Distrib. Syst.*, To appear.

[15] Z. King, J. Blackburn, and A. Iamnitchi. Battorrent: A battery-aware bittorrent for mobile devices. In *Proc. of the 11th Intl. Conference on Ubiquitous Computing, Poster Session*, 2009.

[16] L. M. Koehly. Social network analysis: A new methodology for counseling research. *Journal of Counseling Psychology*, 45:3, 1998.

[17] J. S. Kong, B. A. Rezaei, N. Sarshar, V. P. Roychowdhury, and P. O. Boykin. Collaborative spam filtering using e-mail networks. *Computer*, 39(8):67–73, 2006.

[18] K. Lewis, J. Kaufman, M. Gonzalez, A. Wimmer, and N. Christakis. Tastes, ties, and time: A new social network dataset using Facebook.com. *Social Networks*, 30(4):330 – 342, 2008.

[19] J. Li and F. Dabek. F2F: reliable storage in open networks. In *Proceedings of the 5th International Workshop on Peer-to-Peer Systems*, 2006.

[20] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *Proc. of the 5th Internet Measurement Conference*, 2007.

[21] A. Mislove, A. Post, P. Druschel, and K. P. Gummadi. Ostra: leveraging trust to thwart unwanted communication. In *Proceedings of the 5th Symposium on Networked Systems Design and Implementation*, pages 15–30, 2008.

[22] NetworkX. Overview. http://networkx.lanl.gov/.

[23] H. F. Nissenbaum. Privacy as contextual integrity. *Washington Law Review*, 79, 2004.

[24] E. Santos-Neto, D. Condon, N. Andrade, A. Iamnitchi, and M. Ripeanu. Individual and social behavior in tagging systems. In *The 20th Conference on Hypertext and Hypermedia*, 2009.

[25] B. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *Proceedings of the 1st Workshop on Mobile Computing Systems and Applications*, 1994.

[26] D. Siewiorek, A. Smailagic, J. Furukawa, A. Krause, N. Moraveji, K. Reiger, J. Shaffer, and F. L. Wong. Sensay: A context-aware mobile phone. In *Proc. of the 7th IEEE International Symposium on Wearable Computers*, 2003.

[27] D. N. Tran, F. Chiang, and J. Li. Friendstore: cooperative online backup using trusted nodes. In *Proceedings of the 1st Workshop on Social Network Systems*, pages 37–42, 2008.

[28] B. Wellman. *Structural analysis: From method and metaphor to theory and substance.* Social structures: A network approach. 1988.

[29] C. Wilson, B. Boe, A. Sala, K. P. N. Puttaswamy, and B. Zhao. User Interactions in Social Networks and their Implications. In *Proc. of the 4th European Conference on Computer Systems*, 2009.

[30] S. A. Yahia, M. Benedikt, L. V. S. Lakshmanan, and J. Stoyanovich. Efficient network aware search in collaborative tagging sites. *Proceedings of VLDB Endowment*, 1(1):710–721, 2008.