Empowering Mobile Crowdsensing through Social and Ad Hoc Networking

Stefano Chessa, Antonio Corradi, Luca Foschini, and Michele Girolami

One of the biggest challenges in a real long-running MCS system lies in the capacity not only to attract new volunteers, but also, and most importantly, to leverage existing social ties between volunteers to keep them involved to build long-lasting MCS communities. In addition, the advent of high-performing devices and ad hoc communication technologies can help to further amplify the effect of sensing actions in proximity of the volunteer devices.

Abstract

Mobile crowdsensing (MCS) enables collective data harvesting actions by coordinating citizens willing to contribute data collected via their sensor-rich smartphones that represent sources of valuable sensing information in urban environments nowadays. One of the biggest challenges in a real long-running MCS system lies in the capacity not only to attract new volunteers, but also, and most importantly, to leverage existing social ties between volunteers to keep them involved to build long-lasting MCS communities. In addition, the advent of high-performing devices and ad hoc communication technologies can help to further amplify the effect of sensing actions in proximity of the volunteer devices. This article originally describes how to exploit these socio-technical networking aspects to increase the performance of MCS campaigns in the ParticipAct living laboratory, an ongoing MCS real-world experiment that involved about 170 students of the University of Bologna for more than two years. The article also reports some significant experimental results to quantify the effectiveness of the proposed techniques.

Introduction

Mobile crowdsensing (MCS) is a recent sensing paradigm that leverages the worldwide availability of smartphones. By installing a crowdsensing application, any smartphone can become part of a (large-scale) mobile sensor network, partially operated by the owners of the phones themselves. A crowdsensing application transforms a smartphone into a data sensing, collection, and sharing terminal to exploit the embedded sensors (cameras, microphones, accelerometers, barometers, etc.) and the mobility of the user carrying the phone to gather information about some events of common interest to users. Considering the widespread diffusion of smartphones and their density (typically in urban areas), the information from which a crowdsensing application draws can be rather dense and fine-grained [1]. This fact, along with the limited investments required to develop and maintain crowdsensing platforms and applications, made this paradigm particularly attractive to smart cities managers to improve the quality of their cities and citizens.

Despite its potential, MCS still faces some barriers to user acceptance [2]. Some of them are technological and will likely fade away with the progress of smartphones and crowdsensing technologies and standards: an example is the heterogeneity of hardware/software platforms that raises the costs for development and maintenance. Other barriers are more related to user acceptance, such as the concern of users for privacy, battery consumption, and communication costs. However, other obstacles will not disappear so easily: persistent examples are the difficulty of involving users and the high rate of dropoff due to users who lose interest in contributing to MCS campaigns. With increasing expansion of MCS, these last two obstacles are quite critical since an insufficient number of involved people can compromise the effectiveness of an MCS campaign. In fact, realistic scenarios may involve only a small fraction of the whole citizenship in MCS campaigns, while most citizens do not take part in it.

To overcome these limitations, we propose novel technical solutions to empower and enrich the whole MCS cycle by exploiting both user sociality and physical proximity to emphasize the effect of MCS campaigns. First, we gain higher user participation by leveraging communities of users, that is groups of users with high mutual attendance [3]. We introduce a new type of cooperative task, the *community-based task*, with multiple stages, and we ask people in the same community to participate to its completion. Second, we explore the possibility of increasing the number of data gathered during an MCS campaign by exploiting other users who are not part of the MCS loop. This capability, which we define as the *sensing amplification factor* [4], enables the smartphone of an MCS user to cooperate with other devices detected in proximity (owned by users not already involved), as happens already in mobile social networks (MSNs) [5, 6]. Finally, we report a selection of interesting experimental results that assess the proposed facilities in the ParticipAct MCS living laboratory,[1] by discussing and evaluating their strengths and weaknesses.

Mobility and Sociality in MCS

MCS services typically cross-cut and work at the overlapping of various different research areas that span from management of large socio-technical smart city systems for optimizing the MCS

[1] Additional information, tools, experimental results, and the ParticipAct MCS platform prototype code are available at http://participact.unibo.it

Stefano Chessa and Michele Girolami are with the University of Pisa and ISTI-CNR; Antonio Corradi and Luca Foschini are with the University of Bologna.

0163-6804/16/$25.00 © 2016 IEEE

process to crowdsourcing techniques that enlarge the number of people involved and aim to gain higher participation. Without claiming completeness, this section first proposes a general model for MCS, then briefly overviews the current state of the art in these very active research fields.

While there are already some good surveys about MCS systems (e.g., [7]), this article proposes a simplified MCS reference architecture by identifying its basic components: we claim that any MCS architecture must consist of the following building blocks. First of all, the MCS crowd is a set of *M* people, representing a fraction of the whole population, and we refer to them as *volunteer*s (drawn as light orange full points in Fig. 1), that deliberately express their availability to be involved in crowdsensing campaigns over the smart city.

Volunteers move within the smart city and establish meaningful social interactions (in the real world) that typically present two properties: their mobility and interaction patterns. In addition, volunteers are expected to carry their smartphones provisioned with the MCS app (i.e., *MCS client*) in charge of receiving MCS requests, also referred to as *tasks*. A task represents a data collection activity to be accepted and completed by volunteers involved in an MCS campaign. Each task may include multiple data gathering actions to complete, typically in a given time span and within an area of interest; these actions can occur either automatically by the MCS app without any other human interaction (apart the initial acceptance), or they may require some active participation of volunteers that should, for instance, provide their feedback about a focused night event. Finally, the *MCS server* is the central backend in charge of storing gathered data creating new MCS campaigns and sending tasks to MCS clients, and it is composed of a dashboard for configuring tasks and a database for the storage of sensed data.

In addition to the above basic MCS issues, we argue that the spatial and social dimensions of tasks can greatly enhance the effectiveness of MCS campaigns, and that potentially powerful aspect is often underutilized. Along that line, we claim the need for two extensions to the MCS reference architecture, in two directions:

- Enhancing the involvement of volunteers and supporting more complex tasks via the introduction of the concept of communities of volunteers
- Increasing the number of data gathered by MCS campaigns by exploiting devices of other users that lie in proximity of volunteers and can offer their data in a crowdsensing task

On one hand, the extended architecture includes the *discovering of communities of volunteers* with a meaningful and durable social interaction among themselves (in Fig. 1 these social ties among volunteers are depicted as solid lines) and the introduction of the new concept of a *community-based task*. Those tasks require that volunteers interact closely with one another toward the entire completion by contributing some different subtasks in their turn; the smaller tasks are called *community-based task stages* or simply stages, and volunteers can offer their work for specific stages,

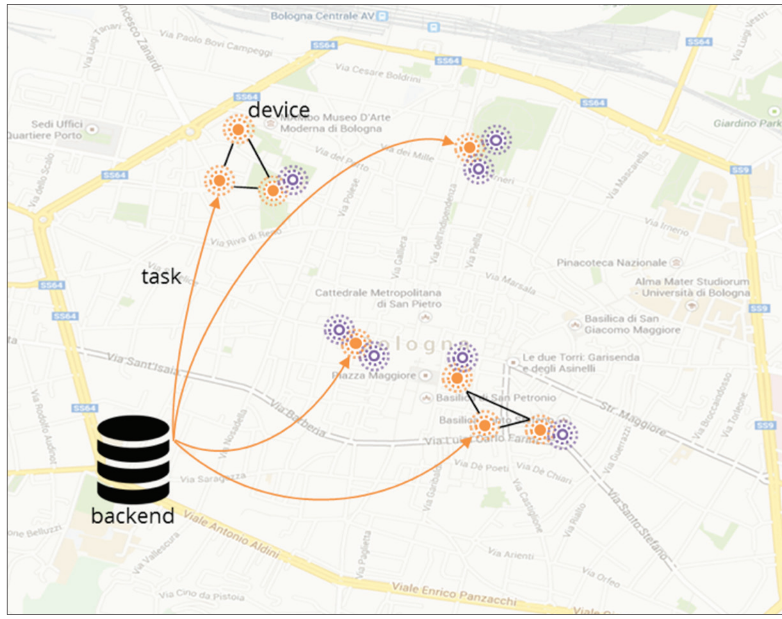

Figure 1. Reference architecture showing communities and devices in proximity.

depending on their capacities and availability. On the other hand, we add discovery of devices close to volunteer devices (referred to as *discovery of co-located device*) where the connection between devices in proximity typically rely on ad hoc communication protocols (devices in proximity are shown in Fig. 1 as pink blank spots, with the wireless transmission range corolla that overlaps with that of a volunteer device).

Focusing now on the literature, the availability of rich smartphone sensing platforms have enabled the creation of several MCS vertical services and apps for different domains including environment monitoring, intelligent transportation, urban dynamics sensing, healthcare, and so forth [7]. Some MCS systems, such as Vita and Medusa [8, 9], take a step further by introducing support functions to ease definition and automatic assignment of tasks to volunteers by also including complex (monetary and non-monetary) incentive mechanisms, which are out of the scope of this article and complementary, to increase participation [10].

More recently, various authors have agreed that one of the key open challenges in MCS is the possibility of exploiting socio-technical network effects to consolidate and extend the crowd by leveraging communities of volunteers [7, 11]. Accordingly, new research trends at the crossroads of MCS, MSNs, and ad hoc sensor networks share with our proposal the goal of facilitating community identification and formation to enable informed scheduling of MCS tasks not only to single volunteers, but also to communities of volunteers [12]. Moreover, some proposals explore the use of opportunistic interactions with nearby devices to further boost the performance of MCS data harvesting [4, 13].

What makes our research efforts unique is the fact that the ParticipAct living laboratory, with long duration (more than two years) and geographical width (the whole Emilia Romagna region in Italy), closely mimics a very realis-

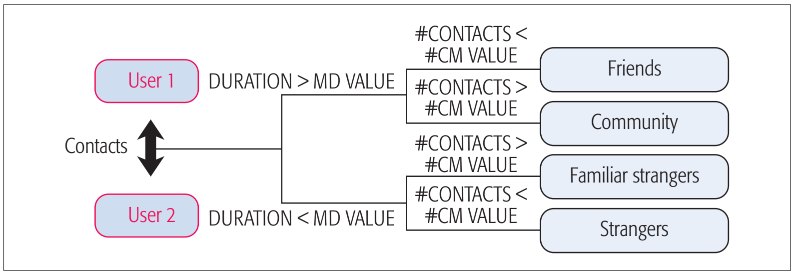

Figure 2. Definition of communities according to k-CLIQUE and assignment of community-based tasks.

tic MCS scenario where users can freely decide whether they accept a task or refuse it, while in several other similar large-scale MCS experiments, such as the Nokia Mobile Data Challenge (MDC), data collection is compulsory for all participants [14]. In addition, focusing on the number of involved users, ParticipAct with its 178 users is one of the largest datasets with respect to other ones available in the literature, such as Cambridge (around 35 users), MIT Reality (around 100 users), and MDC mentioned above (around 190 users).

Amplifying and Maintaining Users in MCS Platforms

This section details the solutions used to enhance the MCS process by leveraging existing communities of volunteers and enabling opportunistic communication with devices in proximity.

Amplifying MCS through Community-Based Tasks

The social interactions among humans are often referred to as social ties to imply the involved relationship among people. Moreover, as a matter of fact, people tend to move according to objectives and activities stemming from their social interactions and derived from specific opportunities and events [15]. Focusing on individual social relationships, they can have different strengths: they can be tight ones, as among friends, or weaker ones, as occur among strangers. The strength of social ties also depends on the time spent together by involved people: typically, friends are people who usually share lots of interests, and tend to meet often and spend much time together; differently, strangers tend to have fewer overlapping interests and stay in touch only for short and sporadic periods. According to the human tendency to spend time with people with similar interests, the MSN literature has classified the MSN topology to characterize how people move together in a smart city. In particular, it has defined four main categories of relationships to distinguish *communities* of people presenting different levels of sociality among themselves; from the most to the least social ones, these are *friends*, *community members*, *familiar strangers*, and *strangers*. Community detection algorithms aim to identify and extract from (volunteers') mobility traces the communities (namely, pinpointing all communities per category and the people in any one of them) belonging to each of those relationship categories.

In particular, the literature proposes many community detection algorithms motivated by different possible expected outcomes [5, 6]. Among all the available ones, we decided to use k-CLIQUE [3] to evaluate the social relationship between individuals. We choose the k-CLIQUE for some main reasons:

- As recognized in the literature, it is a general-purpose algorithm very easy to apply to a wide range of application scenarios.
- It does not require any previous profiling knowledge of volunteer mobility metrics.
- It has a relatively low computation complexity, $O(n^2)$ in the size of the explored network, and hence it is suitable for a periodical re-computation.

Going deep in the technical aspects, k-CLIQUE is a distributed spatio-temporal detection algorithm coming in many versions; the basic version of k-CLIQUE requires two parameters to decide whether to add a device to the community (called Familiar Set): the parameters are the *cumulative contact duration* (DURATION in Fig. 2) and the *number of contacts* (#CONTACTS in Fig. 2) [3]. In this article, we use k-CLIQUE to identify the four categories of relationship (friends, community members, familiar strangers, and strangers) typically used to quantify the strength of human relationship that can be measured by the time spent together and the frequency of encounters. Along that line, we determine different mean values of DURATION and #CONTACTS (MDVALUE and #CMVALUE, respectively) to identify and distinguish these four social categories. Indeed, the configuration of MDVALUE and #CMVALUE is an important decision, because it affects the cardinality of node communities and has to be tuned for the specific MCS datasets and mobility traces.

Finally, we use communities detected via k-CLIQUE to schedule tasks to communities; as already stated, we introduced community-based tasks because they favor wider participation and social involvement of volunteers, but let us also stress the importance of distinguishing different relationship categories. Once statistics on past community-based task completion rates are collected, it is possible, given a target community-based task completion ratio, to probabilistically decide the best relationship category to be involved. In other words, although "friends" communities will typically perform better (as also demonstrated by experimental results collected in the field in our living laboratory), a fair task management strategy suggests avoiding always involving the same category of community. Hence, a manager can more freely decide the community to be involved, also depending on the importance of the task itself. Along the same line, to decrease the MCS burden on single users, a volunteer who belongs to multiple communities is asked only for one community. We schedule the community-based task starting with the community with a stronger relationship, friends, first, then community members, and so forth, and we avoid sending the task multiple times, excluding all other communities of the same volunteer. To take a step further, it would be possible to profile single users, and eventually entire communities, according to their interests, and use these profiles to drive and further refine community-based task scheduling decisions. While that is part of our ongoing efforts on further evolving community-based task scheduling, we already explored those advanced scheduling strategies applied to single-user tasks considering both spatial and interest dimensions; for more detail, we refer interested readers to our previous works [1, 4].

Amplifying Sensing through Opportunistic Communications

Our *amplification factor* aims to exploit the sensing services that other (non-MCS) devices may offer to enlarge the number of sensed data. In particular, when an MCS client detects the presence of another device that can offer and publish data of interest for its present task, it can opportunistically pick up the foreign data to shorten its completion time.

This amplification approach requires cooperation among devices that is enabled by two current technologies available in the wearable device market: *short-range radio interfaces* and *applications for content sharing*. Short-range interfaces are Bluetooth or WiFi, for instance, now available on most smartphones, tablets, wristbands, and smart watches; they allow detection of nearby devices (i.e., 0 to 10 m) and interact with them without any broadband connections. Along the other line, the diffusion of *applications for sharing contents* has pushed a deeply novel way of device (and hence people) collaboration. Currently, all the most popular application markets offer apps for novel resource sharing techniques, for sharing either Internet connection (tethering), an instant messaging service, or sensors access.

The *amplification factor* f quantifies the benefit of exploiting the opportunistic communications with other devices in an MCS campaign. Given a task of an MCS campaign, let β be the amount of data that can be gathered by only considering devices of those volunteers who decide to accept the task, say $V \subset M$ (M is the total number of volunteers). We define f as the ratio between the amount of data η that the task can gather by opportunistically exploiting other devices and β:

$$f = \eta/\beta. \quad (1)$$

Thus, β is the lower bound of the number of retrievable data, while η represents the upper bound of it. Moreover, let us observe that data collected by different devices in the same place at the same time might overlap (unless some measurement error occurs); in that case they would account only for one data chunk. Hence, assuming that the sensor readings are requested at discrete time slots, we can provide an estimation of η by defining the *amplification set* G_i^t of device $i \in V$ at time slot t as the subset of the neighbor set N_i^t of i that provide i with additional data useful for the task. Letting q be the probability that a neighbor of i provides information useful for the task, we have $G_i^t = q|N_i^t|$. Hence, η can be written as

$$\eta = \Sigma_t \, |\cup_{i \in V} p^* G_i^t|. \quad (2)$$

Equation 2 provides an estimation (shown in Fig. 3) of the contribution given by one device in V to η, by varying the number of devices in V and

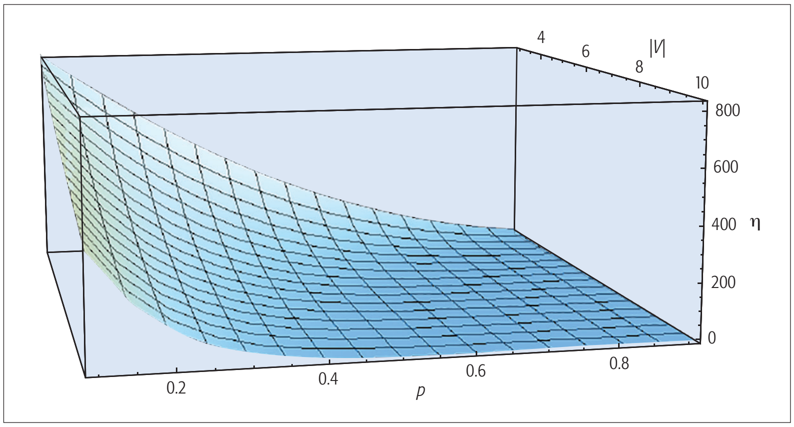

Figure 3. Contribution of η varying $|V|$ and p.

the probability p that in a time slot t, a device belongs to any pair of G_i^t and G_j^t for some $i, j \in V$. In particular, we observe that the value of η decreases for higher values of the cardinality of V. This because with a larger number of devices that receive the task, the chance that any two subsets G_i^t and G_j^t have a larger intersection is also higher. The same effect is also due to high values of p, which also result in small values of η. Therefore, in order to improve the amplification factor it is not sufficient to send the task to additional devices, but it is also important to choose the additional devices in such a way that their amplification sets do not overlap much with that of the other devices running the task in order to ensure that p is kept low, for example, by selecting users belonging to different communities such as strangers or familiar strangers.

Experimental Campaign

The experimental assessment of a large-scale crowdsensing system in a realistic scenario poses complex problems and opens up social, technical, and logistic challenges. The reported results have been collected within the long-running and still active ParticipAct living laboratory that we have maintained at the University of Bologna since January 2014. ParticipAct is a large crowdsensing deployment that now involves 178 volunteers, all of them students of our university from different courses, years, and campuses: the Bologna campus (123 students) and the Cesena campus (55 students). Although, as in any similar experiment, it is an open question whether obtained results and implications could be extended to a more general scenario, the ParticipAct dataset is large enough (in both time, almost two years, and space, with different cities involved) to draw some first important observations, rather realistic for urban setting scenarios. As another observation, our university campuses are not self-contained, but spread over some metropolitan areas, and the same is true for most universities in Italy. For this reason, volunteer paths and behaviors are not limited to a specific area, but relate to the whole urban territory that coincides with the same area of all citizens living in the same smart city. In the following, we present a selection of experimental results aimed at quantitatively assessing the functions originally presented in this article.

Our first experimental result shows the effectiveness of the newly introduced *community-based* task. In particular, we have compared volunteer involvement before and after the introduction of this feature to evaluate the impact of the employed technique in consolidating participation of volunteers for community-based tasks of different complexity. The analysis focused on periods of one month: ParticipAct mobility traces for one month all involved volunteers and typically contain about 7 million data location points that made it possible to extract an average of 41,570 contacts per month between our volunteers. First of all, we run some preliminary configuration tests to tune the k, MDVALUE, and #CMVALUE parameters; the goal was to find the best values to distinguish different relationship categories and balance the number of communities identified for each category. After some empirical tuning, we found that putting k = 4, MDVALUE = 144 h (20 percent of the total number of hours in one month of 30 days), and #CMVALUE = 5 contacts per day, it is possible to distinguish well an average number of 35 communities for each month with the distribution shown in Fig. 4a. In particular, communities are divided as follows: 52 percent *friends*, 35 percent *community members*, and the remainder distributed among 4 percent *familiar strangers* and 9 percent *strangers*. Let us stress that volunteers in strangers communities present very low sociality. Due to their very low responsiveness, after some first tentative community-task assignments to them, we decided not to use communities of this category for scheduling community-based tasks, but to simply consider single independent users as the volunteers in this category.

Then our second experimental result allowed us to automatically identify communities in September 2015, that is, 37 communities, and to use them in the scheduling of 16 community-based tasks in October 2015; these 16 community-based tasks present increasing complexities and a number of stages ranging from 1 to 5 (points in Fig. 4b are average values over all these tasks, and all presented measurements have exhibited a limited variance, always under 8 percent). All these tasks are very easy and fast to complete, such as ordering the first three soccer teams classified in the last World Cup championship and ordering the first N lyrics of a song. The first important result is that *the introduction of community-based tasks produced an increase in the acceptance rate* of the proposed tasks themselves: before the acceptance of a new task requests settled at 38 percent, the rate of volunteers ignoring new tasks settled at 55 percent, and those explicitly refusing at 7 percent; after the introduction of community-based tasks, instead, the acceptance rate improved to 51 percent, while the number of volunteers ignoring and refusing dropped to 45 and 4 percent, respectively. For the sake of fair comparison, we considered simple routine tasks that always required very simple action, that is, multiple choice questions to answer and usually taking less than one minute.

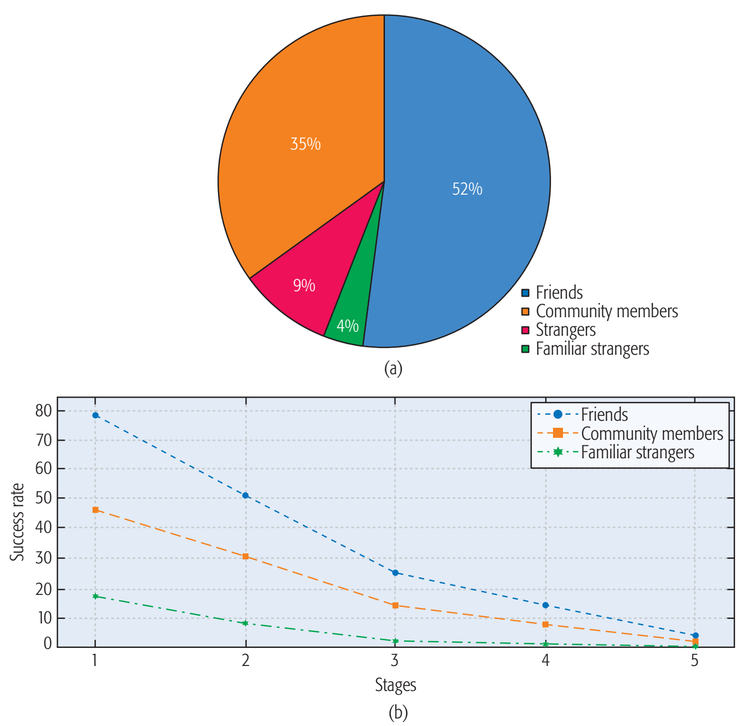

Figure 4. Evaluation of a) communities; b) community-based tasks success rate.

Afterward, we focused our analysis on community-based task success rate (expressed as percentage of success in Fig. 4b), and also on completion of all stages by involved volunteers; in particular, at each stage, we asked all participants in the group to answer part of a simple question by excluding those who had already replied at previous stages. As in Fig. 4b, there is a rather clear distinction between the performances obtainable with the different relationship categories: the higher the sociality, the better the completion ratio; that also allows, given a target completion ratio, (statistically) deciding on the category of communities to involve and those to exclude, thus lowering the load of requests sent to potential volunteers. Let us also add that an important guideline to follow in the design of community-based tasks is to *avoid a too complex task with too many stages*; in fact, over 4 stages the completion ratio drops below 20 percent for any category of community.

Finally, our third experimental test works on the amplification factor to evaluate the improvements in terms of additional data collected. For the sake of simplicity, respecting the general theoretical model of Fig. 3, we focused on the amplification factor for one single device receiving a task (hence $\beta = 1$), and we set probability $q = 1$ to represent the most optimistic case in which all encountered neighbors provide information suitable for the task. With such a setting, the presented results can be considered as the upper bound of the amplification factor f, as shown in Fig. 5, which reports the amplification factor for the whole crowdsensing campaign. On the x axis, we report the hours elapsed since the start of the experimentation for a total of approximately 9 months; on the y axis we report the amplification factor. The experimentation starts in December right before the Christmas holidays, during which very few students meet, resulting in very low amplification factor. From

January to June, the amplification increases and stays in the range [0 to 1.2]. The reason for such an *f* trend is the scheduling of many university lectures and examination breaks that attract students and force them to stay in touch for longer periods. As a result, the average cardinality of the neighborhood also increases. Summer is also significant as in Fig. 5, when few students meet each other, resulting in a low amplification factor. In fact, *the amplification factor increases during the work week* (typically Monday to Friday, see weekly pattern in Fig. 5) when people are more likely to meet and stay in contact for longer periods. Conversely, *the amplification factor decreases during late weekends* giving rise to a weekly pattern of encounters.

From the previous analysis, we observe that the number of expected results from a task is deeply affected by social activities of people. The ParticipAct dataset highlights that routinely cyclic aspects, and also other events forcing people to meet and to stay in contact, can increase the average neighborhood relationship and hence the amplification factor. Furthermore, *the knowledge of routine patterns like weekly patterns can be exploited in order to synchronize the submission of a task with the daily rhythm of a crowd*.

Let us conclude by summarizing some lessons we have learned toward new campaigns of the whole MCS process. First, even with a small number of volunteers, such as our 178 ParticipAct ones, a simple community detection solution is possible for community-based task scheduling. Second, the introduction of community-based tasks might be used as an indirect incentive [7] to increase volunteers' sense of belonging, so it induces an increase in acceptance rate. Third, MCS tasks should be as simple as possible, avoiding too many stages. Fourth, our experiments show a strong relationship between the amplification factor and the routine behavior of volunteers. Hence, fifth and finally, MCS systems should exploit that additional awareness to refine and synchronize task scheduling strategies with the rhythms of a crowd.

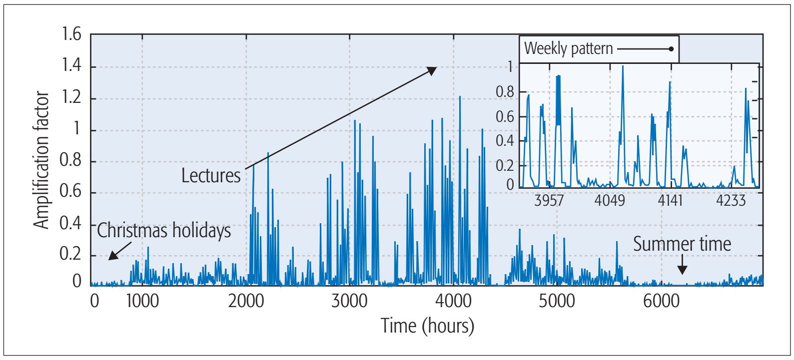

Figure 5. Results of the amplification factor with the ParticipAct dataset.

Conclusions

Mobile crowdsensing is a powerful tool for performing sensing campaigns with citizens; however, one of the most difficult barriers to the spread of MCS campaigns is recruiting volunteers. To overcome this barrier, the social context of people involved in the MCS can effectively increase performance. In particular, we propose two complementary solutions. The first one aims to keep users more involved by leveraging on their existing social ties, which is achieved by introducing community-based tasks that address groups of users sharing social relationships at different levels. The second solution aims to increase the number of data by opportunistically collecting relevant information already made available by non-MCS devices located in proximity of volunteers.

These two solutions represent a first attempt to exploit the knowledge of the social context to increase the results of an MCS campaign. That suggested three further lines of investigation. The first one deals with the detection of communities more accurately reflecting the social events shared by volunteers. The solution of this article exploits only spatial-temporal properties for detecting groups of people who are members of the same community. However, a step further can be taken by combining orthogonal sociological markers such as the physical activity of people, speech intensity, and similarity among visited places. All such markers can be mashed up together in order to identify strong and durable ties among volunteers. The second line of investigation relies on a deeper knowledge of volunteer profiles; information such as interests, social habits, and preferences may be exploited to decide more accurately the target volunteers for a specific task. Finally, along the third direction and complementary to the work presented in this article, we are also considering the possibility to exploit more direct incentive mechanisms to extend the core volunteer base. On one hand, we are considering the possibility of providing different of kinds of benefits and monetary micro-payments to motivate new users entering the MCS system; on the other hand, we are including in ParticipAct novel gamification and entertainment strategies to evolve MCS and sensing tasks into a game that offers virtual rewards to more active users and communities.

Acknowledgments

This research was supported in part by CIRI, Center for ICT Technology Transfer of the University of Bologna, funded by POR FESR Emilia-Romagna 2007-2013.

References

[1] G. Cardone *et al.*, "The Participact Mobile Crowd Sensing Living Lab: The Testbed for Smart Cities," *IEEE Commun. Mag.*, vol. 52, no. 10, Oct. 2014, pp. 78–85.

[2] Y. Xiao *et al.*, "Lowering the Barriers to Large-Scale Mobile Crowdsensing," *Proc. 14th ACM Wksp. Mobile Computing Systems and Applications*, article 9, 2013, 6 pages.

[3] P. Hui *et al.*, "Distributed Community Detection in Delay Tolerant Networks," *Proc. ACM/IEEE Int'l. Wksp. Mobility in the Evolving Internet Architecture (MobiArch'07)*, Kyoto, Japan, 27 Aug. 2007, pp. 1–8.

[4] M. Girolami *et al.*, "Social Amplification Factor for Mobile Crowd Sensing: The ParticipAct Experience," *Proc. IEEE Int'l. Symp. Computers and Communications*, Larnaca, Cyprus, 6–9 July 2015, pp. 117–23.

[5] M. Girolami, S. Chessa, A. Caruso, "On Service Discovery in Mobile Social Networks: Survey and Perspectives," *Computer Networks*, vol. 88, 2015, pp. 51–71.

[6] P. Bellavista, R. Montanari, and S. K. Das, "Mobile Social Networking Middleware: A Survey," *Elsevier Pervasive Mobile Computing*, vol. 9, no. 4, Aug. 2013, pp. 437–53.

[7] B. Guo *et al.*, "Mobile Crowd Sensing and Computing: The Review of an Emerging Human-Powered Sensing Paradigm," *ACM Computing Surveys,* vol. 48, no. 1, Article 7, Aug. 2015, pp. 1–31.

[8] H. Xiping *et al.*, "Vita: A Crowdsensing-Oriented Mobile Cyber-Physical System," *IEEE Trans. Emerging Topics in Computing*, vol. 1, no. 1, June 2013, pp. 148–65.

[9] M.-R. Ra *et al.*, "Medusa: A Programming Framework for Crowd-Sensing Applications," *Proc. 10th Int'l. Conf. Mobile Systems, Applications, and Services*, Low Wood Bay, United Kingdom; 25-29 June 2012, pp. 337–50.

[10] H. Xiong *et al.*, "CrowdTasker: Maximizing Coverage Quality in Piggyback Crowdsensing under Budget Constraint," *Proc. IEEE Int'l. Conf. Pervasive Computing and Communications*, St. Louis, MO, Mar. 2015, pp. 55–62.

[11] N. D. Lane, "Community-Aware Smartphone Sensing Systems," *IEEE Internet Computing*, vol. 16, no. 3, May–June 2012, pp. 60–64.

[12] V. Pejovic and M. Musolesi, "Anticipatory Mobile Computing: A Survey of the State of the Art and Research Challenges," *ACM Computing Surveys*, vol. 47, no. 3, article 47, Apr. 2015, pp. 1–29.

[13] B. Guo *et al.*, "Opportunistic IoT: Exploring the Harmonious Interaction Between Human and the Internet of Things," *Elsevier J. Network and Computer Applications*, vol. 36, no. 6, Nov. 2013, pp. 1531–39.

[14] J. K. Laurila *et al.*, "The Mobile Data Challenge: Big Data for Mobile Computing Research," *Proc. Mobile Data Challenge Wksp. in conjunction with Int'l. Conf. Pervasive Computing*, Newcastle, United Kingdom, June 2012, pp. 1–8.

[15] M. Conti *et al.*, "From Opportunistic Networks to Opportunistic Computing," *IEEE Commun. Mag.*, vol. 48, no. 9, Sept. 2010, pp. 126–39.

Biographies

Stefano Chessa (stefano.chessa@unipi.it) graduated from the University of Pisa, Italy, where he received a Ph.D. degree in computer science in 1999. Since 2015 he is an associate professor at the Computer Science Department of the University of Pisa. He has also been a research associate of the ISTI-CNR Institute since 2003. He has co-authored more than 100 papers published in international journals and conference proceedings. His current research interests are in the areas of Internet of Things, smart environments, ambient assisted living, activity recognition, and indoor localization.

Antonio Corradi [M] (antonio.corradi@unibo.it) graduated from the University of Bologna, Italy, and received an M.S. in electrical engineering from Cornell University, Ithaca, New York. He is a full professor of computer engineering at the University of Bologna. His research interests include distributed and parallel systems and solutions, middleware for pervasive and heterogeneous computing, infrastructure support for context-aware multimodal services, network management, and mobile agent platforms. He is a member of the Italian Association for Computing (AICA).

Luca Foschini [M] (luca.foschini@unibo.it) graduated from the University of Bologna, Italy, where he received a Ph.D. degree in computer engineering in 2007. He is now an assistant professor of computer engineering at the University of Bologna. His interests include distributed systems and solutions for system and service management, management of cloud computing, context data distribution platforms for smart city scenarios, context-aware session control and adaptive mobile services, and mobile crowdsensing and crowdsourcing.

Michele Girolami (michele.girolami@isti.cnr.it) graduated from the University of Pisa, where he received a Ph.D. degree in computer science in 2015. Currently, he is a member of research staff at CNR-ISTI with the Wireless Network Laboratory. His research interests include data forwarding and service discovery in mobile social networks, crowdsensing techniques, and context-aware middleware for smart environments.