Dynamic Data Driven Crowd Sensing Task Assignment

Layla Pournajaf, Li Xiong, and Vaidy Sunderam

Emory University, Atlanta, Georgia, U.S.A.
lpourna@emory.edu, lxiong@emory.edu, vss@emory.edu

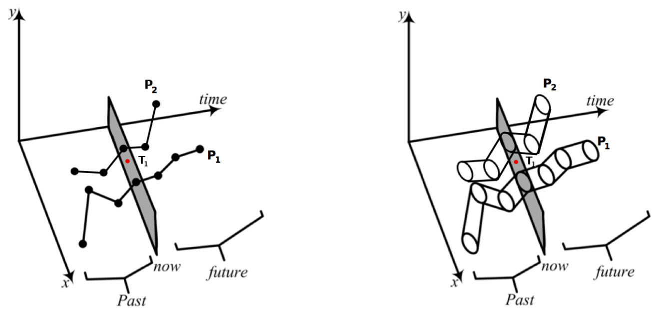
Abstract
To realize the full potential of mobile crowd sensing, techniques are needed to deal with uncertainty in participant locations and trajectories. We propose a novel model for spatial task assignment in mobile crowd sensing that uses a dynamic and adaptive data driven scheme to assign moving participants with uncertain trajectories to sensing tasks, in a near-optimal manner. Our scheme is based on building a mobility model from publicly available trajectory history and estimating posterior location values using noisy/uncertain measurements upon which initial tasking assignments are made. These assignments may be refined locally (using exact information) and used by participants to steer their future data collection, which completes the feedback loop. We present the design of our proposed approach with rationale to suggest its value in effective mobile crowd sensing task assignment in the presence of uncertain trajectories.

Keywords: Mobile crowd sensing, Dynamic task assignment, Uncertain trajectories, Feedback loop

1 Introduction

Mobile crowd sensing [7] enables individuals to participate in a collective data sensing paradigm using their smartphones or other computing devices (e.g., contributing pictures, videos, audios, location, or speed measurements). To fully utilize such a paradigm, a task assignment module could be used to recruit a set of qualified individuals to perform the sensing tasks. A general task assignment framework in mobile crowd sensing includes three main entities:

1. *Participants* are individuals who use a sensor to obtain or measure the required data about a subject of interest.
2. *Applications* or end users are the entities that request data through tasks and then utilize the information acquired by participants.
3. *Tasking entities* are responsible for distribution of tasks to participants who meet the requirements of applications. In certain architectures, end users and participants can also act as tasking entities.



(a) Exact Trajectories (b) Uncertain Trajectories

Figure 1: Effect of uncertainty on spatial task assignment with two participants P_1 and P_2, and a task T_1.

An interesting and valuable class of crowd sensing applications is geospatial crowd sensing, in which the goal is to collect data about specific targets that could be objects, events, or phenomena at particular locations. Examples of such geospatial sensing tasks include instant event-based news reporting at the place of event, traffic monitoring, or trail condition updates after storms. In geospatial task assignment, participants' location information are often used in order to coordinate sensing tasks based on the participants' proximity to the sensing targets. Tasks are distributed to participants based on their locations which can be optimized for coverage-based goals (e.g. maximum coverage or partial coverage) combined with cost-based goals (e.g. minimum cost in terms of distance that needs to be traveled to the sensing target or budgeted cost) [11, 23, 18]. However, in many applications, the exact location of participants might not be available to the task assignment server due to error in location-detection devices, noisy transmissions, or explicit location perturbation due to privacy concerns. In addition, participants may be constantly moving (e.g. commuters) and sensing tasks may be updated (e.g. moving crowd in an event).

Figure 1 illustrates an example of the spatial task assignment problem with two participant trajectories indicated as P_1 and P_2 and the location of a spatial task shown as T_1 at current time (i.e. labeled as now in the figure). Figure 1a includes exact trajectories as the sequence of locations and time stamps while Figure 1b only contains uncertain trajectories as the sequence of location areas at each time point. Assume that a task assignment server has access to these trajectories along with the location of the task, and an assignment goal which requires tasks to be assigned to the closest participants at each time point. While it is intuitive to assign T_1 to P_1 at current time in Figure 1a using exact locations, it might not be as straightforward in Figure 1b which deals with uncertain trajectories. Thus there is an urgent need for dynamic task assignment in mobile crowd sensing for maximized sensing coverage and minimized sensing cost, while adapting in real-time to the dynamic and uncertain locations of participants and the sensing requirements of the applications.

In this paper, we propose a dynamic data driven framework for spatial task assignment in mobile crowd sensing with dynamic and uncertain participant locations. Our approach is based on the DDDAS (Dynamic Data Driven Application Systems) [4] paradigm. The DDDAS concept is crucial to address the big data problem in such crowd sensing applications in order to steer and assign the data collection tasks in targeted ways, adapting dynamically to application needs and the dynamic and uncertain locations of participants. The task assignment entails a synergistic feedback loop between application simulations and data collection: 1) based on assigned tasks, participants report the collected data and possibly their current (uncertain) locations to the application; 2) the collected data are dynamically integrated into an executing simulation to augment or complement the application model (e.g. flood movement), 3) the reported (uncertain) locations are dynamically integrated into an executing mobility model to accurately track participants' moving locations, and 4) conversely the executing simulations update the data collection targets and requirements as well as participants' locations which are then used by the task assignment module to make new task assignments and steer future data collection. Through model-based prediction and filtering, the DDDAS feedback loop is essential to dynamically steer future data collection, adapting in real time to data dynamics, moving participants, and application needs.

An overview of our project PREDICT (Privacy and secuRity Enhancing Dynamic Information Collection and moniToring) is given in [25] which develops a framework for dynamic data collection, aggregation, and analysis with feedback loops enhanced by privacy mechanisms (i.e. when personal information is being collected), while in this paper, we focus on task assignment for data collection in crowd sensing applications. The remainder of this paper is organized as follows. In Section 2 we give an account of previous work. In Section 3 we present an step by step overview of our proposed framework followed by a comprehensive description of each module. Finally, Section 4 gives the conclusions and our future research direction.

2 Related Work

2.1 Mobile Crowd Sensing Task Assignment

Task management in mobile crowd sensing can be categorized based on the task allocation scheme as *i) autonomous task selection* or *ii) coordinated task assignment* [17]. In *autonomous task selection* scheme, participants select their tasks autonomously from a pool of globally available tasks. Examples of these approaches may be found in [22, 6]. Since the selected tasks are not optimized globally, these approaches tend to be inefficient with respect to sensing cost or global utility. On the other hand, *coordinated task assignment* recruits qualified participants and efficiently allocates available sensing resources (i.e. participants) to tasks to meet the goals of applications. Applications might aim for different criteria such as high task coverage, data credibility and quality, or low cost. Examples of this approach can be found in [19, 21, 5, 11, 12]. In this paper, we are interested in a coordinated task assignment approach, however, our work differs from these previous works since we deal with uncertain trajectories of participants and do not have access to exact locations. In addition, we apply the DDDAS concept to account for dynamic data and application needs.

2.2 Mobility Modeling and Prediction

Recent mobility studies aim at mining individual's mobility patterns and constructing mobility models using high resolution positioning data such as GPS [1, 15, 16]. In contrast to synthetic mobility models which are based on randomly generated movements, trace-based mobility models fit a statistical model of individual's mobility using real-world trajectories. These models can be constructed based on the trajectories under the walk mode [13, 20] or may consider other modes of transportation [14] (i.e. such as driving or public transportation) which may introduce different statistical properties. Several techniques have been proposed in literature for clustering locations or trajectories, mining their dependencies, and modeling the probability of transition between locations which can be found in a recent survey [15]. Variants of Markov

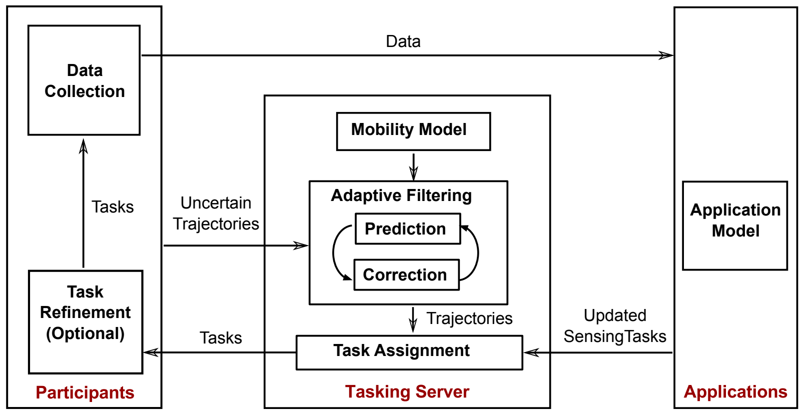


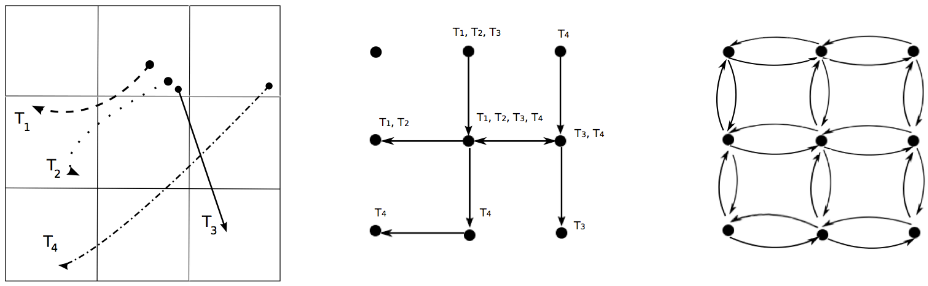
Figure 2: Adaptive dynamic data driven framework for uncertain spatial task assignment

model are widely used to model the transition probability between locations by individuals and predict their next destination [2, 9, 14]. Another popular framework for destination prediction using historical trajectories is Bayesian inference [26]. In this paper, we utilize these mobility models in addition to the reported (uncertain) locations to augment the mobility models in order to get more accurate estimates of participants' locations for task assignment.

3 Dynamic Data Driven Framework For Task Assignment

We propose a novel DDDAS based framework for dynamic spatial task assignment with uncertain trajectories illustrated in Figure 2. Our framework includes an offline learning process which builds a mobility model by mining the public trajectories to be used as the process model in the filtering module. The framework entails a feedback loop composed of the following key components:

- Participants, based on their assigned tasks, report collected data and voluntarily their current (uncertain) locations to the application.
- The collected data are dynamically integrated into an executing simulation to augment or complement the application model (e.g. flood movement), which updates the sensing targets for future data collection,
- The reported (uncertain) locations are dynamically integrated into a filtering component to augment or complement a mobility model. The prediction step computes or simulates a participant's current location based on her historic trajectory and the mobility model (i.e. prior estimates). The correction step integrates the reported (uncertain) location and predicted location into a more accurate location (i.e. posterior estimate).
- The updated sensing targets and requirements as well as participant information with posterior estimates of their locations are then fed to the uncertain task assignment module which assigns tasks to participants using a probabilistic model while globally optimizing sensing coverage and cost. The output of this module, which is a mapping of tasks to participants, will be returned back to participants.



(a) A 3×3 grid representation (b) A graph representation (c) A Markov model

Figure 3: An example map with a grid, graph, and Markov model representation

- Once each participant receives a set of tasks from the tasking server, if she has access to her exact location, a local task refinement step (i.e. a second-stage tasking module) can be used to further optimize her set of tasks.

- The final assigned tasks are then used to steer the future data collection as well as possibly participants' future trajectories since they might need to travel to the location of sensing targets.

Since the application model is dependent on specific applications and sensing tasks, we focus on the general task assignment module in this paper and explain each component in detail below.

3.1 Learning Mobility Models

To learn a mobility model, we use a set of publicly available trajectories as a historical data set to calculate transition probabilities between adjacent locations. To formulate the location transition process, Bayesian inference and Markov model are two popular methods which are examined in this paper. Note that, we do not consider modes of transport or location-based activities to build these models, however this module can be easily extended to include them. We note that we can learn mobility models for specific individuals if we have sufficient historical personalized trajectories. Of course, additional privacy mechanisms may be required to protect the individual's privacy when learning such personalized models.

3.1.1 Bayesian Inference

A grid-based road network with mapped trajectories can be used to build a Bayesian inference framework for next location prediction [26]. Generally, a map is a two-dimensional $g \times g$ grid with granularity of a cell (i.e. all the locations within a single cell are considered to be the same). A graph is build based on the grid where each cell corresponds to a node in the graph, then trajectories are mapped to sequences of these nodes. An example of a 3×3 grid is given in Figure 3a, where trajectories T_1, T_2, T_3, and T_4 are shown in the map. Figure 3b represents the trajectories mapped in a graph which is created based on the grid. T_1 and T_2 are identical in the graph because of the granularity which is a grid cell.

After mapping the set of all trajectories T, given a partial trajectory T_j, the probability of a node n being the next destination of T_j can be computed based on the Bayes rule as

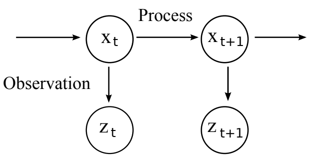


Figure 4: A Simple State-Space Model

Equation 1, which is the probability that node n contains the next location in given trajectory T_j, conditioning on the query trajectory T_j.

$$P(next(T_j) = n \mid T_j) = \frac{P(T_j \mid next(T_j) = n)P(next(T_j) = n)}{P(T_j)} \tag{1}$$

On the other hand, the probability that node n is the next destination of trajectory T_j can be calculated as Equation 2 which is the number of trajectories containing the sequence of nodes in T_j followed by node n divided to the total number of trajectories.

$$P(next(T_j) = n) = \frac{|\{T_i \mid (T_j, n) \in T_i\}|}{|T|} \tag{2}$$

3.1.2 Markov Model

Using the same graph as Figure 3b, a Markov model can be constructed which assumes a state for each node of the graph as in Figure 3c. For each pair of adjacent nodes, both transition directions are considered and the probability of each directed edge p_{rs} and p_{sr} are calculated as the probability of location transition between n_r and n_s and vice versa. In a first-order Markov model, only the current state determines the probability of transiting to the next state, so the probability p_{rs} is calculated as the number of trajectories which have the sequence of two nodes (n_r, n_s) divided to the number of trajectories which have n_r as shown in Equation 3.

$$p_{rs} = \frac{|\{T_i \mid (n_r, n_s) \in T_i\}|}{|\{T_i \mid n_r \in T_i\}|} \tag{3}$$

After calculating all the probabilities between the nodes, a $g^2 \times g^2$ transition matrix M is created which can be used as a process model in a state-space model as described in Section 3.2. Moreover, other advanced Markov models such as higher order or hierarchical Markov models can be built to feed more variables to the model which are not considered in this paper, but can be easily plugged into the framework.

3.2 Adaptive Filtering

The filtering component in our framework provides estimates of noisy locations in order to improve the accuracy of location information per time stamp to be used in task assignment module. First, given the mobility model as described in Section 3.1, we can create a linear state space model as shown in Figure 4 and formulated in Equation 4. In Figure 4, x represents true states while z shows observed states.

$$\begin{aligned} x_{t+1} &= Ax_t + \omega_t \\ \omega_t &\sim \mathcal{N}(0, Q) \end{aligned} \tag{4}$$

where A is the time-invariant, linear coefficient and ω_k represents the noise of the linear model. Intuitively, the process model linearly relates the current location state x_{t+1} to the previous state x_t as described in Markov model, except for a white Gaussian noise ω, called the process noise with variance Q. The observed state z_t is also obtained from the true state x_t at each time point t and contains additive measurement noise. we can build a observation model as Equation 5.

$$\begin{aligned} z_t &= Hx_t + \nu_t \\ \nu_t &\sim \mathcal{N}(0, R) \end{aligned} \tag{5}$$

where H is the linear coefficient and ν_k represents the additive measurement noise. In our context, this noise can originate from device inaccuracy (e.g. GPS noise) or a perturbation method (e.g. differential privacy) and be modeled differently according to the measurement equipment or process. In this paper, we assume a Gaussian noise with variance R.
Given the process model and the measurement model, a filtering algorithm is used for posterior estimation of true state to minimize the measurement error. Two popular filtering algorithms in literature are Kalman Filter [10] and Particle Filter [8]. Kalman filter is optimal for linear processes with a Gaussian noise, while Particle filter makes no assumption about the process model or noise in the state-space model but can be computationally expensive.

3.3 Uncertain Spatial Task Assignment

The output of the filtering module for the given set of noisy participant locations at time t is a set of filtered uncertain locations to be used in spatial task assignment. Given a set of updated spatial tasks and assignment goals by applications, and the set of participants with uncertain locations (i.e. output of filtering module), we have developed methods and algorithms to handle location uncertainty and optimize assignment process to achieve required goals of the applications [18]. In our work, we considered coverage-based assignment goals with a distance-based cost model, however, any assignment goal can be adopted in our task assignment module in the dynamic framework. General definitions of task assignment, spatial task and distance-based cost model is given as follows.

A *task assignment* is a mapping of participants to tasks. Each participant-task pair can be considered also an individual assignment. A cost might be set for each assignment.

A *spatial task* (i.e. location-based task) includes a target (i.e. an object, event, or phenomena), a location, a time-frame for sensing, and other specific instructions or sensor requirements to perform sensing.

A *distance-based cost model* defines the cost of each assignment as a function of distance between the participant and the task location. In the simplest definition which considers the exact distance as cost, closest participants to each task will form least costly assignments.

Task coverage could be defined in different ways.

- *Single-coverage* model means each task needs to be performed by only one participant to be considered as covered.
- *K-coverage* model requires each task to be assigned to k participants to be considered as covered. This model can be used in untrustworthy or uncertain environments to avoid faulty or missing data.

Considering these task coverage models, coverage-based assignment goals might fall in one of the following categories.

- *Maximum coverage assignment* aims at maximizing the coverage of tasks by participants. In a single-coverage model, the assignment goal can be summarized as maximizing the number of assigned tasks.
- *Minimum-cost coverage-based assignment* aims at achieving a task coverage goal with minimum cost. In a single-coverage model, the assignment goal might require only a portion of tasks to be assigned with minimum cost.
- *Minimum-cost maximum-coverage assignment* aims at maximizing a task coverage goal with minimum cost.

Task Assignment with Uncertain Locations In spatial task assignment with uncertain locations, since exact locations of participants are not provided, the distance information between task locations and participants is unavailable to the tasking server, therefore the server is required to deal with location uncertainty to achieve assignment goals. Spatio-temporal queries over uncertain data have been extensively studied with many algorithms to handle queries such as nearest neighbors, top-k, and range queries [24]. These queries mostly consider the results for one query object and not a set of objects, therefore cannot be directly adopted in our work or would be very inefficient. Among other spatio-temporal queries, K closest pairs query (K-CPQ) [3] which is the problem of finding the K closest pairs between two spatial datasets is the most related query type to our problem, but is not studied for uncertain datasets.

Knowing the set of uncertain locations of participants as a set of minimum bounding area of each uncertain location with a probability density function, we can apply two simple methods to calculate the expected distances between this set and the set of exact locations of the tasks. As a naive method, for each participant, we calculate the centroid point of all possible location samples in each uncertainty area and use it to calculate the expected distances between this point and all locations of the task set. As a more accurate alternative method, we first apply a geometric pruning algorithm to remove the task-participant pairs with zero probability of being accessible (i.e. a participant can not travel to the place of a task) and shrink the uncertain areas to only contain the accessible location samples. Then, for the remaining pairs with shrinked areas, we calculate the probabilities of the task locations being accessible by participants as well as the expected distance between task locations and shrinked areas. Finally, the set of expected distances and the accessibility probabilities can be used in our probabilistic task assignment methods proposed in [18].

Local Task Refinement Once the task assignment is done at the tasking server, the assignments are sent to individual participants. The goal of the local task refinement is to further optimize task assignment results of the global assignments by each participant using her exact location. This approach requires the participants to know their actual locations. This assumption is reasonable when spatial noise is added by participants for privacy-preserving purposes, but can not be applied to other types of uncertainty. Hence, we have considered an optional task refinement module in participant side of our proposed framework.

The final assigned tasks are then used by the participants to steer their future data collection, which completes the feedback loop. Our main insights are that given this feedback loop, data will be collected in a targeted way, adapting dynamically to application needs and data dynamics. The data collection process is adapted according to participants' moving trajectories and the sensing tasks in such a way that sensing cost and resource utilization are optimized.

4 Conclusions and Future Work

The emergence of a variety of crowd sensing applications on top of a platform based on moving individuals (i.e. carrying smart devices) prompts an urgent need for effective management models such as spatial task assignment. Such crowd sensing platform comes with some inherent challenges including a dynamic environment (i.e. moving participants or task targets) with uncertainty (i.e. noisy or imprecise location information) which are addressed in our dynamic spatial task assignment framework based on the DDDAS (Dynamic Data Driven Application Systems) paradigm. Our approach includes 1) a dynamic data driven framework with feedback loops to steer data collection, adapting in real time to moving participants, data dynamics, and application needs, 2) a state-space modeling of participant mobility using public trajectories which continuously adjusts to the dynamic and uncertain locations of participants exploiting an adaptive filtering module, and 3) a spatial task assignment approach to recruit the best set of participants to achieve application-specific goals such as maximum coverage or minimum cost. As a next step, we plan to evaluate our framework with real-time data by implementing a real-world crowd sensing application using our dynamic task assignment approach. We are also interested in extending our approach to take into account other assignment criteria such as data credibility and quality.

Acknowledgments

This research is supported by the Air Force Office of Scientific Research (AFOSR) DDDAS program under grant FA9550-12-1-0240.

References

[1] Nils Aschenbruck, Aarti Munjal, and Tracy Camp. Trace-based mobility modeling for multi-hop wireless networks. *Computer Communications*, 34(6):704–714, 2011.

[2] Daniel Ashbrook and Thad Starner. Using gps to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, 7(5):275–286, 2003.

[3] Antonio Corral, Yannis Manolopoulos, Yannis Theodoridis, and Michael Vassilakopoulos. Algorithms for processing k-closest-pair queries in spatial databases. *Data & Knowledge Engineering*, 49(1):67–104, 2004.

[4] Frederica Darema. Dynamic data driven applications systems: A new paradigm for application simulations and measurements. In *Computational Science-ICCS 2004*, pages 662–669. Springer, 2004.

[5] T. Das, P. Mohan, V. N. Padmanabhan, R. Ramjee, and A. Sharma. Prism: Platform for remote sensing using smartphones. pages 63–76. in Proceedings of the 8th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys), 2010.

[6] Dingxiong Deng, Cyrus Shahabi, and Ugur Demiryurek. Maximizing the number of worker's self-selected tasks in spatial crowdsourcing. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 314–323. ACM, 2013.

[7] Raghu K Ganti, Fan Ye, and Hui Lei. Mobile crowdsensing: Current state and future challenges. *Communications Magazine, IEEE*, 49(11):32–39, 2011.

[8] Neil J Gordon, David J Salmond, and Adrian FM Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 107–113. IET, 1993.

[9] Ramaswamy Hariharan and Kentaro Toyama. Project lachesis: parsing and modeling location histories. In *Geographic Information Science*, pages 106–124. Springer, 2004.

[10] Rudolph Emil Kalman et al. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[11] Leyla Kazemi and Cyrus Shahabi. Geocrowd: enabling query answering with spatial crowdsourcing. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 189–198. ACM, 2012.

[12] Leyla Kazemi, Cyrus Shahabi, and Lei Chen. Geotrucrowd: trustworthy query answering with spatial crowdsourcing. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 304–313. ACM, 2013.

[13] Kyunghan Lee, Seongik Hong, Seong Joon Kim, Injong Rhee, and Song Chong. Slaw: A new mobility model for human walks. In *INFOCOM 2009, IEEE*, pages 855–863. IEEE, 2009.

[14] Lin Liao, Donald J Patterson, Dieter Fox, and Henry Kautz. Learning and inferring transportation routines. *Artificial Intelligence*, 171(5):311–331, 2007.

[15] Miao Lin and Wen-Jing Hsu. Mining gps data for mobility patterns: A survey. *Pervasive and Mobile Computing*, 2013.

[16] Mirco Musolesi and Cecilia Mascolo. Mobility models for systems evaluation. In *Middleware for Network Eccentric and Mobile Applications*, pages 43–62. Springer, 2009.

[17] Layla Pournajaf, Li Xiong, Daniel A. Garcia-Ulloa, and Vaidy Sunderam. A survey on privacy in mobile crowd sensing task management. Technical Report TR-2014-002, Department of Mathematics and Computer Science, Emory University, 2014.

[18] Layla Pournajaf, Li Xiong, Vaidy Sunderam, and Slawomir Goryczka. Spatial task assignment for crowd sensing with cloaked locations. In *Proceedings of the 2014 International Conference on Mobile Data Management (MDM)*. IEEE, to appear.

[19] S. Reddy, D. Estrin, and M. Srivastava. Recruitment framework for participatory sensing data collections. pages 138–155. in Proceedings of the 8th International Conference on Pervasive Computing, Springer Berlin Heidelberg, May 2010.

[20] Injong Rhee, Minsu Shin, Seongik Hong, Kyunghan Lee, Seong Joon Kim, and Song Chong. On the levy-walk nature of human mobility. *IEEE/ACM Transactions on Networking (TON)*, 19(3):630–643, 2011.

[21] Mehdi Riahi, Thanasis G. Papaioannou, Immanuel Trummer, and Karl Aberer. Utility-driven data acquisition in participatory sensing. EDBT/ICDT, ACM, March 2013.

[22] M. Shin, C. Cornelius, D. Peebles, A. Kapadia, D. Kotz, and N. Triandopoulos. Anonysense: A system for anonymous opportunistic sensing. *Journal of Pervasive and Mobile Computing*, 7(1):16–30, 2010.

[23] Houtan Shirani-Mehr, Farnoush Banaei-Kashani, and Cyrus Shahabi. Efficient viewpoint assignment for urban texture documentation. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 62–71. ACM, 2009.

[24] Yijie Wang, Xiaoyong Li, Xiaoling Li, and Yuan Wang. A survey of queries over uncertain data. *Knowledge and Information Systems*, pages 1–46, 2013.

[25] Li Xiong, Vaidy Sunderam, Liyue Fan, Slawomir Goryczka, and Layla Pournajaf. Predict: Privacy and security enhancing dynamic information collection and monitoring. *Procedia Computer Science*, 18:1979–1988, 2013.

[26] Andy Yuan Xue, Rui Zhang, Yu Zheng, Xing Xie, Jin Huang, and Zhenghua Xu. Destination prediction by sub-trajectory synthesis and privacy protection against such prediction. In *Proc. IEEE 29th International Conference on Data Engineering, ICDE*, 2013.