Journal of Artificial Intelligence Research 61 (2018) 433-474 Submitted 02/17; published 03/18

Coordinating Measurements for Environmental Monitoring in Uncertain Participatory Sensing Settings

Alexandros Zenonos AZ2G13@ECS.SOTON.AC.UK
Sebastian Stein SS2@ECS.SOTON.AC.UK
Electronics and Computer Science,
University of Southampton,
Southampton, UK

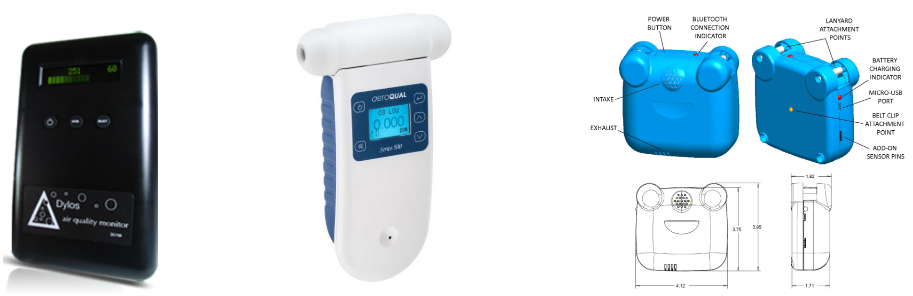
Nicholas R. Jennings N.JENNINGS@IMPERIAL.AC.UK
Department of Computing,
Department of Electrical and Electronic Engineering,
Imperial College London,
London, UK

Abstract

Environmental monitoring allows authorities to understand the impact of potentially harmful phenomena, such as air pollution, excessive noise and radiation. Recently, there has been considerable interest in participatory sensing as a paradigm for such large-scale data collection because it is cost-effective and able to capture more fine-grained data than traditional approaches that use stationary sensors scattered in cities. In this approach, ordinary citizens (non-expert contributors) collect environmental data using low-cost mobile devices. However, these participants are generally self-interested actors that have their own goals and make local decisions about when and where to take measurements. This can lead to highly inefficient outcomes, where observations are either taken redundantly or do not provide sufficient information about key areas of interest. To address these challenges, it is necessary to guide and to coordinate participants, so they take measurements when it is most informative. To this end, we develop a computationally-efficient coordination algorithm (adaptive Best-Match) that suggests to users when and where to take measurements. Our algorithm exploits probabilistic knowledge of human mobility patterns, but explicitly considers the uncertainty of these patterns and the potential unwillingness of people to take measurements when requested to do so. In particular, our algorithm uses a local search technique, clustering and random simulations to map participants to measurements that need to be taken in space and time. We empirically evaluate our algorithm on a real-world human mobility and air quality dataset and show that it outperforms the current state of the art by up to 24% in terms of utility gained.

1. Introduction

Applications involving the placement of sensors for monitoring environmental phenomena, especially noise and air pollution, are receiving considerable attention (Jutzeler, Li, & Faltings, 2014; Seinfeld & Pandis, 2012; Stevens & D'Hondt, 2010), as it is a subject that concerns many, from environmental organisations to policymakers to the general public. Noise pollution can cause heart conditions, loss of sleep and changes in brain chemistry (Chepesiuk, 2005). Poor air quality can have short-term effects on health, such as headaches, asthma, eye irritations and lack of concentration (Mabahwi, Leh, & Omar, 2014; Seaton,



(a) Dylos (b) Aeroqual (c) AirBeam

Figure 1: Examples of portable devices that measure air quality in terms of atmospheric particulate matter (PM).

Godden, MacNee, & Donaldson, 1995). More importantly, however, air pollution is responsible for a range of heart-related diseases and leads to approximately 7 million deaths per year (WHO, 2014). This costs the global economy hundreds of billions of pounds in terms of lost labour income and trillions in welfare losses (World-Bank, 2016). Given this, understanding the situation and predicting how it is going to change, in the long term as well as on a daily or even hourly basis, is crucial in allowing decision makers to take action. For example, in terms of urban planning, city councils can make decisions about where to build parks and plant trees to minimise the effect of highly polluted areas in cities (Paoletti, Bardelli, Giovannini, & Pecchioli, 2011). They can also make planning decisions about new roads so as to handle traffic efficiently based on air pollution measurements (Ministry of Housing Communities, 2014). Furthermore, it can help doctors link environmental factors with symptoms and thus affect potential patients' treatment (Burke, Estrin, Hansen, Parker, Ramanathan, Reddy, & Srivastava, 2006).

At present, the problem of monitoring air pollution is mainly tackled with networks of static stations. These are often funded and operated by government authorities collecting measurements on a continuous basis, and they are controlled by a number of experts. These stations are very costly to acquire and maintain, resulting in the collection of limited information (Jutzeler et al., 2014). However, an alternative way to monitor environmental phenomena is to exploit *participatory sensing*, which is a promising paradigm for data collection (Burke et al., 2006). Instead of making use of expensive equipment and employing a number of experts to work for hours to collect data, the burden is divided between a higher number of individuals (not necessarily experts), carrying affordable sensor devices such as mobile phones and 'Dylos'[1], 'Aeroqual'[2] or 'AirBeam'[3] as shown in Figure 1. In a typical participatory sensing application, users take a measurement or a reading using the air quality device and transfer it to the smartphone via Bluetooth or USB. Then the measurements are uploaded to the server via the internet. This paradigm enables the public to gather and share local knowledge. The benefits of this approach are firstly that it is cheaper than the traditional approach of using static stations (Jutzeler et al., 2014) by

[1] http://www.dylosproducts.com/

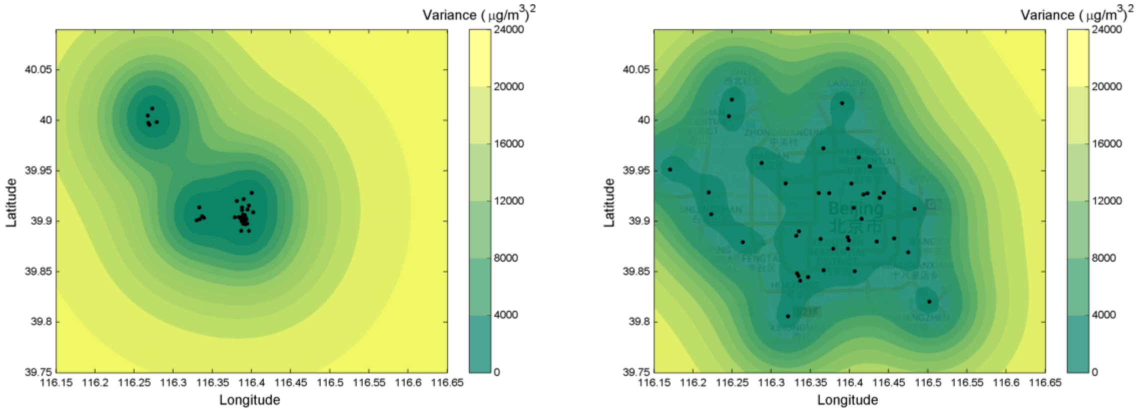
[2] http://www.aeroqual.com/product/series-500-portable-air-pollution-monitor

[3] http://www.takingspace.org/aircasting/airbeam/

more than a factor of ten in some cases (WAQI, 2015). Secondly, static sensors are often located away from streets and emission sources in order to reflect the average pollution over an area (Jutzeler et al., 2014). Consequently, that approach might underestimate the true exposure of people to air pollution. Participatory sensing alleviates this by enabling people to directly take measurements at places they frequent during their daily routine and which may be near sources of air pollution.

Participatory sensing has already been employed for noise pollution monitoring. In particular, Noisetube (2008-2014) was a proof-of-concept trial that allowed people to use their smartphone devices to take measurements in order to produce a noise heatmap over specific cities around the world (Stevens & D'Hondt, 2010). The latest participation count stated that the project had approximately 1300 registered users from 650 cities in 75 countries (Stevens, 2012; D'Hondt, Stevens, & Jacobs, 2013). In a different application, following the Fukushima Daichi Nuclear Power Plant disaster in 2011, people volunteered to assist authorities in measuring the radiation levels in the environment. Specifically, an open platform, Safecast, was utilised to allow people to submit measurements taken using specialised equipment (Geiger tubes). A total of approximately 1000 devices were used globally, and environmental radiation data collection via Safecast has seen an exponential growth since 2011, leading to a current total of over 50 million measurements in 2017. Averaging over that period, this corresponds to about 7.1 million measurements per year, 20 thousand per day, or approximately 815 measurements per hour. This was a significant milestone in participatory sensing, as this was the first time it was successfully employed in the wild on such scale (Brown, Franken, Bonner, Dolezal, & Moross, 2016).

While delivering impressive results (D'Hondt et al., 2013; Whitney & Richter Lipford, 2011; Brown et al., 2016), existing participatory sensing campaigns lack an important element. They do not provide a coordination system that can efficiently guide or suggest to participants when and where to take measurements in order to fill coverage gaps over time in the area of interest. Thus, there is no system that outputs a mapping of users to specific locations at specific times. This is a major problem, because some areas may remain unexplored, which leads to a false or partial picture of the situation over the entire environment that the campaign initiator is interested in (Stevens & D'Hondt, 2010). Also, people may provide redundant information by taking measurements at the same time and place, which can waste participants' effort as well as communication and processing resources. By way of illustration, Figure 2 shows in a city simulation of air pollution monitoring the contrast in reduction of variance (which is a measure of uncertainty) in air quality over an entire city (Beijing) between a system in which users take measurements in an uncoordinated way compared to one where there is effective coordination. In this case, Figure 2(a) shows people taking measurements in an uncoordinated way. For example, since there are a lot of people in the city centre and near popular sightseeing destinations in Beijing, most people take measurements at those locations. Figure 2(b) shows the same number of people taking measurements in a coordinated way. People are still mainly in the city centre, but they are instructed to take measurements on their way to other places where there is not much coverage. Therefore, we observe that the variance over a larger area is reduced. This in turn shows that coordinating measurements results in the exploration of a larger area and thus more information about the environment is gained.



(a)

(b)

Figure 2: Uncoordinated measurements (a), coordinated measurements (b)

A key challenge in participatory sensing is that people participate for a variety of reasons (Gao, Liu, Wang, Zhao, Song, Su, Crowcroft, & Leung, 2015). Specifically, some people participate for monetary incentives (extrinsic incentives) (Jaimes, Vergara-Laurens, & Labrador, 2012). This can take the form of micro-payments or coupons (Albers, Krontiris, Sonehara, & Echizen, 2013). For example, a micro-payment scheme was used as an incentive to promote realtime participation in a university campus garbage monitoring campaign (Reddy, Estrin, Hansen, & Srivastava, 2010). Likewise, SenseUtil is a model where the consumer who needs data pays the producers who carry out sensing tasks and report the data. The price is determined based on the concept of demand and supply (Thepvilojanapong, Tsujimori, Wang, Ohta, Zhao, & Tobe, 2013), where the price changes dynamically according to the sensing frequency, quantity of nearby sensing locations and user preferences. Others volunteer for social reasons, for example, to gain public recognition or a high position on a leader-board. In some systems, volunteers compete against friends for points or badges (Anderson, Huttenlocher, Kleinberg, & Leskovec, 2013). Finally, some people volunteer because of their personal interest in a social cause, altruism, or as a hobby (intrinsic incentives) (Jennings, Moreau, Nicholson, Ramchurn, Roberts, Rodden, & Rogers, 2014). For example, in addition to Noisetube and Safecast described above, where people are interested in the cause of the campaign, an application for finding an endangered species of insects in the UK relies on the excitement of the visitors of a particular area on the South coast of England that the insect is believed to inhabit (Zilli, Parson, Merrett, & Rogers, 2013).

Crucially, in either case it cannot be assumed that participants will provide an unlimited number of measurements; they should not be seen as robotic entities that behave exactly as instructed all the time. Instead, it is better to view them as self-interested agents that have their own personal goals and limited information about the environment[4]. For instance, they might believe that taking a measurement in the city centre is more useful than elsewhere,

[4] Note we do not model agents as self-interested in the game-theoretic sense, as we cannot in general assume detailed information about their specific goals and utility functions. Instead, we use the terminology to refer more broadly to agents who cannot directly be controlled (as is common in the AI and multi-agent literature).

since more people are potentially affected, or they might not take a measurement at all since they are too busy with their daily routine and might believe that the impact of a single measurement is low at their location. Thus, the act of taking measurements should be viewed as incurring inconvenience costs or even requiring financial compensation. In our work, we capture this by assuming a budget for taking measurements. Specifically, each user is assumed to have a different budget in order to capture the different preferences of people (heterogeneity) in terms of their willingness to take measurements. This budget might be an actual supply of money or a notional supply of good will.

In some cases, such as when people carry GPS-enabled phones with them or their phones have Internet connectivity, knowledge about how they tend to move at particular times might be available to the participatory sensing campaign initiator. However, to date, none of the existing work in this area has looked at harnessing this knowledge. This information could be available by learning those patterns (Baratchi, Meratnia, Havinga, Skidmore, & Toxopeus, 2014a), as people tend to be predictable in their daily routine and thus their daily routine can be probabilistically modelled (McInerney, Stein, Rogers, & Jennings, 2013b). A participatory sensing system could exploit the fact that participants are likely to be at specific locations at specific times in order to prepare a plan for suggesting measurements. However, even if people are at their predicted locations, it is uncertain whether or not they will actually take a requested measurement. Crucially, we cannot assume that participants will always be willing or able to take a measurement when requested, even if they initially agreed on contributing a number of measurements. For example, related work has shown that only 83% of smartphone users engage with notifications on their device within five minutes of receiving them (Sahami Shirazi, Henze, Dingler, Pielot, Weber, & Schmidt, 2014), which implies some desired measurements will be missed.

Against this background, we are interested in monitoring environmental phenomena using the participatory sensing paradigm. In particular, we focus on *intelligently* collecting data (measurements/observations) with the assistance of people in order to maximise the information we learn about the environment over a period of time. This, in turn, will enable us to build a fine-grained air quality map, covering an area of interest, which will lead to better situational awareness (Endsley, 1988). In so doing, we take into consideration the real-life constraints discussed above. These include the number of measurements each individual is willing to take, the uncertainty associated with their willingness to take a given measurement when asked and the uncertainty about the users' location in the future.

Specifically, we propose a novel stochastic coordination algorithm (adaptive Best-Match) for large-scale monitoring of dynamic environmental phenomena using the participatory sensing paradigm. The algorithm adaptively selects observations to be taken at each timestep[5] and maps individuals to measurements both in space and time in order to maximise the information learned about the environment over a given time period. The algorithm is able to deal with hundreds of participants at the same time[6], incorporates probabilistic knowledge of the mobility patterns of humans and assumes that people have a daily limit/budget on the number of measurements they are willing to take. Also, our algorithm

[5] We discretise time in our model, and each timestep could represent a different time interval. In our work, a timestep denotes 1 hour.

[6] It can deal with tens of thousands of people over the period of a day. This figure is larger than any existing participatory sensing campaign.

deals with the inherent uncertainty associated with participants taking a measurement when suggested to do so (reliability). Our algorithm makes use of artificial intelligence techniques such as clustering to group spatially close people and thus scale up in terms of the number of participants it can deal with at every timestep. Also, it uses heuristic search (local greedy search) to search through a number of possible combinations of clusters and random simulations to deal with uncertainty. In particular, the contributions of this paper are:

- We propose a new participatory sensing coordination framework (Section 3) that captures the architecture of our vision for a real participatory sensing application for environmental monitoring that intelligently coordinates participatory sensing campaigns.
- We are the first to define the problem of coordinating measurements for participatory sensing applications in the presence of uncertainty (Section 4).
- We develop a novel stochastic coordination algorithm that is able to handle hundreds of participants at every timestep. The algorithm consists of an offline component, which is responsible for simulating participatory sensing campaigns and choosing the best measurements to be taken, as well as an online component that adapts the measurements based on real-time information. The algorithm considers each individual's budget, incorporates probabilistic knowledge about human mobility patterns and deals with the uncertainty related to the willingness of people to take a measurement when notified by the system (Section 6).
- We empirically evaluate our algorithms on real human mobility and air quality sensor data (Section 7) from the Urban air and Geolife datasets in Beijing and show that our algorithm significantly outperforms the state of the art by up to 23.91% in terms of utility gained.

The remainder of this work is organised as follows: In Section 2, we discuss related work and give a background on relevant techniques used in the literature. In particular, we divide related work into three parts (agent coordination, task allocation and sensor placement), which represent the key research areas that our research draws on. Also, we provide the background on the techniques on how an environmental phenomenon can be modelled. In Section 3, we present our vision of how a real participatory sensing system could be used in practice such that it utilises an intelligent coordination system. In Section 4, we formally introduce the coordination problem for participatory sensing settings using an agent-based formalism. In Section 5, we describe how we model the environment using a non-linear non-parametric technique and quantify informativeness of the measurements taken. In Section 6, we present our adaptive Best-Match algorithm for coordinating measurements in uncertain participatory sensing settings. The algorithm uses heuristic search to find a mapping between participants and measurements should be taken that are most informative. It is adaptive, as it chooses which measurements to be taken given the uncertainty in the users' behaviour. In Section 7, we explain how we evaluated our algorithms, describe the benchmarks we have used and present our findings. In particular, we compare our algorithm to the benchmarks in terms of total utility gained and runtime. In Section 8, we conclude and determine the potential future steps to be taken to extend our work.

2. Background and Related Work

In this section we present related work and give a background on the techniques used. Specifically, our work draws on the intersection of three main research areas: agent coordination, task allocation in the context of crowdsourcing and sensor placement in environmental monitoring. In the sections below we show the relationship with each one of those and present related work in that area. Finally, we provide the technical background on how environmental phenomena are modelled in order to provide a method for quantifying information gained by taking measurements and building a heatmap of the spatio-temporal phenomenon.

2.1 Agent Coordination

First of all, our work is related to the agent coordination domain, as the overall purpose of our system is to coordinate the measurements taken by users acting as self-interested agents. Users typically have limited information about the environment and follow their own personal agendas. In the agent coordination literature, mobile agents, such as autonomous ground vehicles (AGVs), autonomous unmanned aerial vehicles (UAVs) or unmanned underwater vehicles (UUVs) are often used to explore an environment or perform specific tasks in an area. Typically, coordination of teams of such agents is computationally intensive and the focus is on finding informative paths for a single autonomous agent (Marchant & Ramos, 2012; Binney, Krause, & Sukhatme, 2010). In order to scale up, domain specific heuristics and clustering approaches are utilised to group spatially close sensing locations and thus reduce the search space (Singh, Krause, Guestrin, & Kaiser, 2009; Stranders, Fave, Rogers, & Jennings, 2010). However, coordination is shown only for a small team of agents (i.e., up to a dozen of autonomous robots (Singh et al., 2009; Ouyang, Low, Chen, & Jaillet, 2014; Low, Dolan, & Khosla, 2011a; Stranders et al., 2010; Schwager, Dames, Rus, & Kumar, 2017; Tiwari, Honoré, Jeong, Chong, & Deisenroth, 2016). Thus, existing work does not scale to the settings we are interested in. Also, these techniques cannot easily be extended to consider probabilistic knowledge about the mobility patterns of participants or the willingness of the users to take a measurement, since the agents are robotic entities that do not have their own agendas but rather follow computed paths.

In other related work, Stranders, Farinelli, Rogers, and Jennings (2009) deal with path finding for mobile sensors considering both the spatial correlations of a phenomenon, as well as the temporal ones. They implement an adaptive receding horizon algorithm in a decentralised manner, which means that there is no central system that controls these sensors but instead they autonomously decide what to do based on the information available to them by exchanging messages with other mobile sensors. The focus of that work is on decentralised approximations and dealing with reliability in the communication network between agents and permanent failure of agents' hardware, which is not a concern in this work. Rather, in our work an agent has a probability of being unavailable at a specific time, in which case users might ignore a notification to take a specific measurement at a given time, as well as a probabilistic model of their mobility patterns. Also, Stranders et al. (2009) assumes that each agent has a specific radius within which it collects information and no underlying model exists, while in environmental phenomena the effect of a measurement can be captured by a probabilistic model that depends on the nature of the phenomenon. In other words, a probabilistic model can capture the development of the phenomenon in space and time and thus is able to create heatmaps by interpolating between measurements as well as predicting into the future (see Section 2.4 for more details). Consequently, this enables us to be more accurate about the information gained when taking each measurement, as well as deciding when and where to take measurements, given the information collected by earlier measurements. Also, in many cases, the measurements taken are noisy, which can be captured by using a probabilistic model, creating accurate heatmaps about the phenomenon monitored. This can lead to better coordination of measurements in order to maximise the information about the environment. Furthermore, even though the algorithm uses a number of approximations and heuristics, it is evaluated only on ten agents, which highlights the complexity of the solution.

Building on this, in other work Stranders, Munoz De Cote, Rogers, and Jennings (2013) approach the problem of continuous multi-agent coordination by modelling the problem of space exploration by a team of mobile agents as a Markov Decision Process (MDP). They show that an approximation algorithm for solving MDPs can be used to continuously coordinate only a small team of mobile agents (up to ten agents) for an infinite time horizon. However, it is not shown to work for larger teams of agents.

In other research, Partially Observable Markov Decision Process (POMDP) algorithms have been used in the context of agent coordination (Pineau, Gordon, & Thrun, 2006; Hollinger & Singh, 2008). Firstly, POMDPs handle uncertainty in both action effect and state observability. Plans are expressed over information states instead of world states, since the world state is not observable. POMDPs form plans by optimising a value function, thus allowing the agent to numerically trade off between alternative ways to satisfy a goal, compare actions with different costs/rewards, as well as plan for multiple interacting goals. Also, instead of producing a sequence of actions, POMDPs produce a full policy for action selection. However, the state space grows exponentially with the number of variables that are considered in the selection problem. Also, the complexity of planning for POMDPs grows exponentially with the cardinality of the state space (Pineau et al., 2006). Thus, multiple agents, and multiple potential spatio-temporal locations where they can take measurements from exponentially increase the state space of the problem, which makes the use of POMDPs infeasible for the settings we consider.

Drawing these together, even though the aforementioned algorithms solve problems that are related to coordinating measurements in participatory sensing settings, they are not applicable in our work mainly because they are not scalable to hundreds of participants. Also, in environmental monitoring of dynamic phenomena, the Markov property might not hold. In particular, taking a measurement at a timestep might provide enough information such that no other measurement is required in the near future, given that the phenomenon is changing slowly over time. Therefore, it might be better for some people to wait many timesteps, given that they have a limited budget of measurements they can take in the future before taking a measurement that would be of greater value in terms of providing more information about the phenomenon. Put differently, for a number of different measurements in space and time, we obtain different amounts of information about the phenomenon. This is not compatible with the Markov property, which requires that the future of the process depends only on the current state, i.e., measurements taken at a given timestep and not on the ones in the past. Consequently, the decision about when to take a measurement needs to be taken given the history of measurements taken so far in space and time and not just the last one.

Overall, in our work we transfer the agent-based problem formulation applied in the multi-agent coordination problem used by Stranders et al. (2013) to the coordination of measurements for environmental phenomena in the participatory sensing setting. This enable us to apply artificial intelligence techniques and exploit domain-specific knowledge to develop an efficient algorithm.

2.2 Task Allocation

Our work is also related to ongoing research in task allocation in the context of spatial crowdsourcing. In particular, it is relevant to task allocation, as the purpose of our system is to allocate tasks to users, i.e., which measurements to take. However, there are substantial differences that are unique to environmental monitoring. Concretely, recent work (Chen, Cheng, Gunawan, Misra, Dasgupta, & Chander, 2014; Chen, Cheng, Lau, & Misra, 2015) uses mobility patterns to effectively coordinate agents in crowdsourcing. The focus of these papers is assigning agents to tasks based on their mobility patterns, so as to maximise the payoff of the tasks within a given time limit. However, no budget is associated with each user to correspond to the inconvenience or the incentive needed to execute the task. This is unrealistic in participatory sensing, because people cannot provide an unlimited number of measurements. Moreover, the tasks are assumed to be independent of each other and once they are executed, they are no longer available. This is not the case when monitoring environmental phenomena, where it is often important to revisit locations in order to keep up with the temporal variations of the phenomenon. Also, the reward gained when taking measurements of an environmental phenomenon is not easy to quantify. It cannot be captured by a fixed reward value as in task allocation. It should be calculated based on the model about the environment (which is examined in Section 5), since each measurement may be different in terms of the information it conveys. In other words, the utility in environmental monitoring is associated with what measurements are taken globally in space and time by the crowd. On the other hand, the reward for a particular task in crowdsourcing is usually independent of what tasks other people are executing, since each task has different characteristics, such as difficulty and type of task, which are not affected by other available tasks. Furthermore, even though simplified assumptions are made, such as the fact that users will always accept and perform the requested tasks or users have up to two alternative routes, in these papers, the complexity of allocating people to tasks is still NP-hard (Chen et al., 2014). Thus, a range of offline greedy approaches including a greedy construction heuristic and iterated local search are utilised. Specifically, they first construct an initial solution as fast as possible by using a greedy heuristic and the quality of the initial solution is improved iteratively by employing an iterated local search (ILS), which is part of the stochastic local search (SLS) algorithms (Hoos & Stützle, 2004). These are a family of high-performance local search algorithms that make use of randomised choices in generating or selecting candidate solutions for a given combinatorial problem instance. In particular, their algorithm performs four main actions: it swaps two agents with two task nodes if that improves the total remaining detour time for both agents. Next, it moves a task from an agent to another with the highest remaining detour time. Then, an unassigned task is chosen with the highest reward and the agent with the highest remaining detour time is selected to do it. Finally, an assigned task is replaced by an unassigned one with higher reward. All possible insertions are examined until the process exceeds a predefined number of iterations. That algorithm, however, is not applicable in our situation as the problem we are addressing is different in the ways described above. However, similar to the work by Chen et al. (2014), we also build on heuristic approaches, and in particular, we propose a novel Stochastic Local Greedy Search (SLGS) algorithm.

Generally, in the area of crowdsourcing, it is a common practice to use algorithms that are first and foremost fast to use, while at the same time achieving their objective (i.e., complete available tasks) (Musthag & Ganesan, 2013). In particular, practical deployments of mobile crowdsourcing often utilise the Pull-based (Proximity-driven) algorithm (Musthag & Ganesan, 2013). This algorithm selects those tasks that are closest to a user's current spatial location, and then lets them choose the task that they think is best suited for them. This approach is fast to utilise and does not require any intelligent system for coordination. However, as we argued above, people have a limited budget. Thus, doing a task at a future timestep might be more rewarding than doing one at the present. Therefore, this approach can lead to a suboptimal total reward. However, since this algorithm is used in practice, it will be adjusted for environmental monitoring settings and used as a benchmark in this work. Specifically, since there are no specific tasks to be executed in environmental monitoring, users will be asked to take measurements when the utility gain is above a threshold. This algorithm is discussed in more detail in Section 7.1.

2.3 Sensor Placement

Finally, our work is related to the sensor placement problem in the context of environmental monitoring. Specifically, it can be viewed as the task of placing a number of sensors, that equals the number of users, in a dynamic environment where specific constraints are associated with the sensors. For instance, the number of sensors to be placed in the environment is changing every timestep (depending on whether a user has some budget left or not), and the location of the sensors is constantly changing as humans follow their daily routine. Importantly, each sensor is associated with uncertainty about their future location and whether they will actually be able to take a measurement when instructed to do so. Since the nature of this problem is combinatorial, finding the optimal solution is computationally infeasible. In a seminal paper, Krause, Singh, and Guestrin (2008) show that the sensor placement problem is NP-hard. They also prove that the sensor placement problem has a desirable property, submodularity, that allows a greedy algorithm to provide specific guarantees about the approximation ratio of the solution provided. In particular, building on the work of Nemhauser, Wolsey, and Fisher (1978), they show that using a greedy algorithm, the solution is always at least $1 - \left[\frac{(K-1)}{K}\right]^K$ times the optimal value and has a limiting value (i.e., as $K \to \infty$) of $(1 - \frac{1}{e})$, where K is the number of sensors placed. In the context of monitoring spatial phenomena, the same property is exploited to produce a polynomial-time approximation algorithm, which is within $(1-\frac{1}{e})$ of the optimum (Guestrin, Krause, & Singh, 2005; Golovin & Krause, 2011). Specifically, Guestrin et al. (2005) greedily deploy a fixed number of sensors in an environment such that a submodular function, and in particular, mutual information between the chosen locations and the locations which are not selected is maximised. The algorithm, at each iteration, adds the sensor which results in the maximum increase in mutual information until the desired number of sensors is reached. This is representative of a large class of algorithms that greedily select the next measurement that maximises an entropy-based criterion until a given budget is exhausted. Given the ubiquity of this approach, we will use the greedy algorithm described by Guestrin et al. as benchmark to our approach.

2.4 Modelling Environmental Phenomenon

A key challenge in monitoring environmental phenomena is to identify any spatio-temporal patterns in the observations that have been made. These patterns are used to make predictions (such as noise and air quality) about the locations where no observations have been made and about the future state of the world.

Regression is commonly used to accomplish this (Stranders et al., 2013; Tiwari et al., 2016; Schwager et al., 2017). This is a statistical process for estimating the relationship among variables and in particular understanding how the value of a variable will change by varying another variable. Two types used for environmental monitoring are Piecewise Linear Regression (Section 2.4.1) and Gaussian Processes (Section 2.4.2).

In our work, we use Gaussian Processes, as it is a non-linear non-parametric regression technique that can identify potential complex spatio-temporal patterns in noisy observations. Also, Gaussian Processes have been used successfully in modelling spatio-temporal phenomena as shown in related work (Guestrin et al., 2005; Krause, Guestrin, Gupta, & Kleinberg, 2006; Low, Dolan, & Khosla, 2011b; Garg, Singh, & Ramos, 2012; Ouyang et al., 2014). Specifically, they provide uncertainty estimations alongside the predictions, which can be used as a basis for utility functions.

In this section, we introduce Piecewise Linear Regression and Gaussian Processes in the context of modelling an environmental phenomenon. We keep the description brief here, but we provide further details about GPs in Appendix A, as it is the model of our choice. The model described there is key for representing the environment as well as valuing information from measurements taken, which are part of the framework proposed in Section 3.

2.4.1 Piecewise Linear Regression Background

Linear regression is commonly used in many practical applications because of its simplicity and computational performance. Environmental phenomena however, exhibit non-linear behaviour over space and time (Stranders et al., 2013). Thus, linear regression is not suitable for modelling the environment. However, Padhy, Dash, Martinez, and Jennings (2010) proposed the use of a variation of linear regression, called Piecewise Linear Regression, as an alternative that could be used in environmental monitoring. In particular, in order to model temperature and pressure, which have a non-linear relationship over time, they used Bayesian inference to decide whether each data point can be sufficiently explained by the current regression model or whether a new linear model is required.

Consequently, that environment is separated into a number of regions such that each region can be modelled by a linear regression. However, the parameters used in this approach increase with the number of linear regressions used, thus making it difficult to estimate them, which causes the model to be computationally expensive. Also, it is not certain where to start and stop in each linear regression, as these parameters need to be estimated again, which might result in an inaccurate model. Moreover, standard piecewise linear regression does not provide the confidence intervals over its estimates, which is useful in order to measure the information gained at each spatio-temporal location.

2.4.2 Gaussian Process Background

Gaussian Processes (GPs) (Rasmussen & Williams, 2006) are a class of nonparametric probabilistic models that are used in modelling spatio-temporal phenomena. For these kinds of phenomena, the interest is not only on the value of the phenomenon (e.g., noise level or air pollution level) at the sensed location but also at locations where no observations were taken. In such problems, regression techniques are used to perform these predictions. Although Piecewise Linear Regression can sometimes capture these relationships, as we have seen above, it is not flexible and does not model the uncertainty about its predictions (Guestrin et al., 2005; Krause et al., 2006; Low et al., 2011b; Garg et al., 2012; Ouyang et al., 2014). In contrast, Gaussian Processes can capture more complex non-linear relationships and also provide a way to measure the uncertainty of those predictions through the notion of variance. Also, they are flexible in the sense that they can model different phenomena by using different covariance functions. These make Gaussian Processes an ideal tool for our work. Appendix A provides a brief introduction to Gaussian Processes and explain their properties in more detail, which are used in the rest of this paper and specifically to quantify informativeness of the measurements taken (Equation 2). In the next section, we show where this model fits in our participatory sensing framework.

3. The Participatory Sensing Framework

Our framework shows how our coordination algorithm fits into the broader context of participatory sensing campaigns for environmental monitoring. In particular, it describes how to efficiently monitor an environment by coordinating measurements, taking into consideration available knowledge about the participants. Figure 3 shows the overall architecture of the framework and illustrates how the components interact with each other. In particular our framework consists of five core components:

- The main component is the *coordination algorithm*, which is the main contribution of this work. This component decides when and where each participant should take a measurement to maximise information about the environment. In other words, it produces a mapping from users to spatio-temporal locations.

- The *human mobility patterns prediction* component is a system for making predictions about the mobility patterns of the participants. This component provides probabilistic information about the future locations of the users, which is used to make decisions about when and where to take a measurement.

- The *budget* component captures the number of measurements an individual is willing to take, since users have a limited number of measurements they can take.

- The *reliability model* component captures the uncertainty related to individuals about whether they will actually take a measurement when asked to do so, as seen in the work

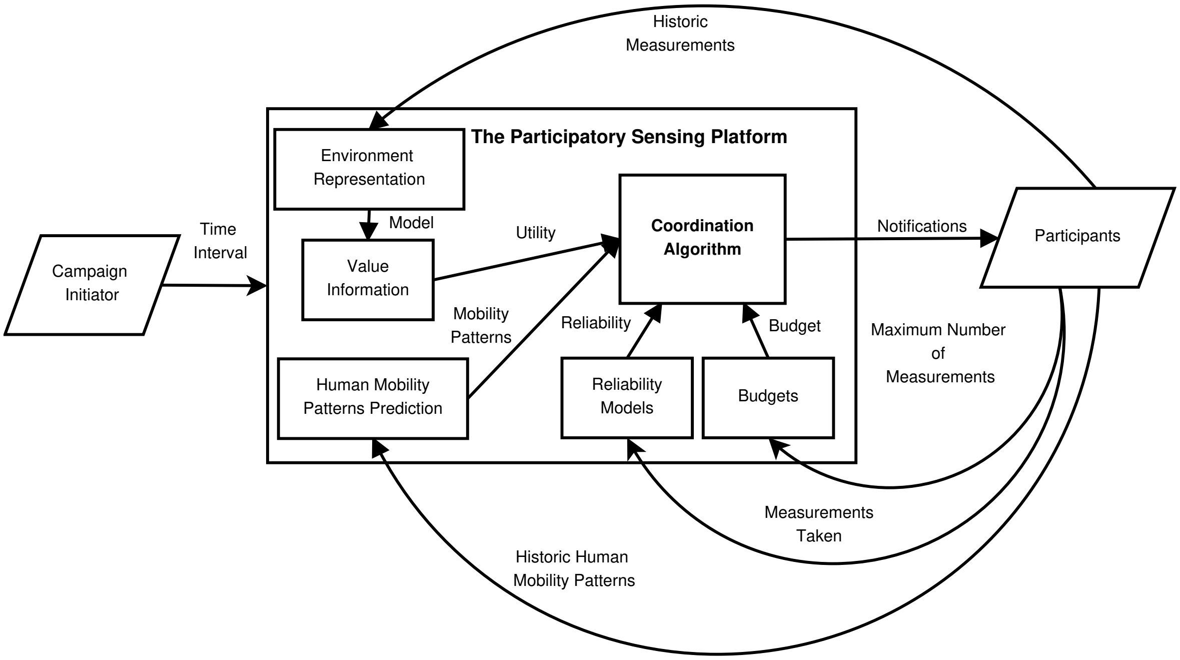


Figure 3: A conceptual architecture of an intelligent participatory sensing platform

of Ramchurn, Mezzetti, Giovannucci, Rodriguez-Aguilar, Dash, and Jennings (2009). This model is built based on information collected from users based on their past behaviour in participatory sensing campaigns.

- The components related to the environmental phenomenon supply the system with information about the environment being monitored. In particular, *historic measurements* from multiple users are fused together using a model that creates a representation of the environment (*environment representation* component). This, in turn, is used to value future measurements taken, in terms of their information value (*value information* component) as in the work of Guestrin et al. (2005).

A participatory sensing campaign is initiated by a person, group or organisation interested in understanding an environmental phenomenon for a particular area (campaign initiator). The campaign initiator is not a core part of the framework, as the campaign is not affected by them. However, the initiator is responsible for the recruitment of the participants and setting up the time interval of the campaign, i.e., the starting and ending date and time.

Anyone willing to take part in the campaign would own or be provided with a smartphone with Internet connectivity and specialised equipment, depending on the phenomenon and environment to be monitored. The participatory sensing platform is responsible for contacting the participants over the Internet in real time or in advance in order to suggest which measurements they need to take and at what time. This would take the form of notifications on their smartphone.

The participants are in a feedback loop, where they provide the platform with the measurements taken, as well as their mobility patterns and their budget. In our work, we assume that participants have to explicitly take a measurement using their mobile device. This is reasonable, because although there are devices that are able to continuously take measurements, this is usually associated with a high energy cost. Moreover, continuous measurements decrease the need for coordination and so we focus on settings where measurements are taken explicitly and may be constraint by a budget. Concerning their mobility patterns, intelligent agents on participants' devices can monitor their behaviour and provide the platform with the mobility patterns as per the work of Sánchez-González, Pérez-Romero, Agustí, and Sallent (2016). The human mobility pattern prediction system infers their future mobility patterns for a specific time horizon. That system is able to produce a number of possible routines and allocate probabilities for each alternative (McInerney et al., 2013b; Baratchi, Meratnia, Havinga, Skidmore, & Toxopeus, 2014b). This could potentially raise privacy concerns, which is an active research topic in participatory sensing. Specifically, Gao et al. (2015) present a number of privacy-aware approaches in participatory sensing. However, this is not the main focus in our work. Instead, we focus on the coordination algorithm which suggests to participants when and where to take measurements, in order to efficiently monitor the environment for a time period.

Also, each participant is associated with a budget (Chon, Lane, Kim, Zhao, & Cha, 2013), which, in our framework, can be given directly by participants or learned from their participation in previous campaigns with the assistance of the intelligent agents. In particular, the aforementioned work started with no monetary incentives for users' contribution for the duration of a month and then rewarded contribution for the next one (which was shown to significantly increase the number of measurements taken). However, this depended on a number of factors, such as the cause of the campaign and specific requirements. Nevertheless, it was shown that people typically use their phone in a participatory sensing setting to take between 2-11 measurements per day. Thus, in our work, we assign a low average budget to users.

The framework also considers that people are not guaranteed to take the measurement requested. In particular, 83% of users check their smartphone notifications within 5 minutes of receiving them[7] (Sahami Shirazi et al., 2014). Thus, the intelligent agents on participants' phones can monitor the behaviour of the participants, as is commonly done in the crowdsourcing domain (DiPalantino, Karagiannis, & Vojnovic, 2010), to provide a model for their reliability (shown in Figure 3) with respect to the system. Specifically, this model can be used to estimate the probability that a user will take a measurement when notified to do so.

In this paper, we do not focus on a complete implementation of the aforementioned framework, since each of the components is an active research area on its own. We rather focus on the the algorithmic challenge of developing the main component, which is an efficient coordination algorithm the maps users to spatio-temporal locations in the environment. Our algorithm, however, is able to exploit probabilistic knowledge about participants' mobility patterns and consider budget and reliability constraints of each participant, as provided by the other components.

[7] Even though this an average, there is great variance in response time and this is affected by a number of factors, including the frequency of notifications, their cause and purpose.

4. Coordination of Measurements Problem Definition

This section formally introduces the problem of coordinating measurements in participatory sensing for environmental monitoring. In particular, we focus on the problem that the *coordination algorithm* has to solve (shown in Figure 3), subject to budget constraints and the reliability of users.

First of all, an environmental campaign is initiated to collect as much information about a particular phenomenon in an environment as possible. An environment $\mathcal{E}$ is a continuous set of spatio-temporal locations (L, T) that the *campaign initiator* is interested in. This is defined by the spatial and temporal boundaries of the area and *time interval* of interest up to time E. A set of *participants* $\mathbf{A}= \{\mathcal{A}_1, \ldots, \mathcal{A}_M\}$ can take a set of discrete measurements (also called observations) within the spatial boundaries of this environment within the time period of the campaign ($O = L \times T$). The set of observations made before or at time t is denoted as $\mathbf{O}_t \subseteq O$, while the set of observations made at time t is denoted as $O_t \subseteq \mathbf{O}_t$.

A utility function $u : 2^O \rightarrow \mathbb{R}^+$ assigns a utility value to a set of observations. The value assigned by this function is based on the entropy, which is a way to measure information (the value information component in the framework in Section 3). This is based on the formulation by Guestrin et al.(2005), and it is further discussed in Section 5. Here, it is sufficient to say that the goal is to maximise the sum of utilities over the time period of the environmental campaign. Intuitively, this captures the total information gained when a set of measurements are taken. Moreover, citeauthorlargecrowdsourcingstudy, (2013) show that people tend to contribute a limited amount of information in participatory sensing campaigns. Hence, we cannot assume that people can take an unlimited number of measurements but they rather have a budget. Formally, each individual $\mathcal{A}_i$ has a specific *budget*, i.e., $B_i \in \mathbb{N}^+$, which is the maximum number of measurements that it can take within a day. We represent the budgets of all users with $B = \{B_1, B_2, \ldots, B_M\}$. Finally, in this work, we assume that people only take measurements without deviating from their daily routine.

A function $r : \mathbf{A} \rightarrow \{v \in \mathbb{R} |\ 0 \leq v \leq 1\}$ assigns a real number between zero and one to users, representing their reliability (the reliability models component in Figure 3). This is the probability that they actually take a suggested measurement when requested to do so by the system. Each user has a personal reliability that is independent of other users. We represent the reliability for all users with $R = \{r(\mathcal{A}_1), r(\mathcal{A}_2), \ldots, r(\mathcal{A}_M)\}$. In our system, even if a user fails to take the measurement suggested, their budget is reduced, so as to avoid suggesting measurements to be taken by the same user if they are not willing to contribute. Intuitively this implies that the users will not be continuously notified to take measurements if they keep ignoring them.

We denote by $\mathbb{U}$ the total utility earned by all the measurements taken by the agents. Thus,

$$\mathbb{U}(\mathbf{O}_E) = \sum_{t=1}^{E} u(O_t) \tag{1}$$

where $u(O_t)$ is the utility gained from a set of observations made by participants at the timestep t, given the effect of all the measurements taken before that. The coordination algorithm needs to decide on a policy, i.e., when and where the citizens should make these observations to maximise this function given a probability distribution over people's possible locations at each timestep and constraints of budget as well as user reliability.

Given this notation, the optimisation problem solved by this algorithm can be formulated as follows: map a set of participants to a set of measurements to maximise the expected utility over the period of the campaign, subject to individual budget constraints of participants. Formally, a policy $S^* = arg\, max_s\, \mathbb{E}(\mathbb{U}(\mathbf{O}_E))$, where $s : \mathbf{A} \rightarrow 2^O$.

5. The Environment Model

Given the introduction on Gaussian Processes in Section 2.4.2 and Appendix A, and the definitions in Section 4 we now focus on probabilistically modelling the environmental phenomenon. This enables us to quantify the informativeness of measurements used in our utility function (Equation 1). In order to model the environmental phenomenon, we first discretise the environment in a way such that a two-dimensional grid is created over space and the time is divided into hourly measurements (timesteps). Consequently, we say that locations $\mathcal{L} \subset L$ are the intersections of the grid and $\mathcal{T} \subset T$ are the timestep. In our work, we convert longitude and latitude into UTM format, i.e., meters, so as to be able to make calculations in the Euclidean space.

Each location $l \in \mathcal{L}$ and time $t \in \mathcal{T}$ is associated with a random variable $X_{l,t}$, that describes an environmental phenomenon, such as noise or air pollution. We use $X_{l,t} = x_{l,t}$ to refer to the realisation of a random variable at a particular spatio-temporal coordinate, which becomes known after an observation is made. In order to describe the phenomenon at time t over the set of locations ($\mathcal{L}$), given that some observations have been made in the past ($\mathbf{O}_{t-1}$), we use $X_{\mathcal{L},t|\mathbf{O}_{t-1}}$. Similarly, we denote by the random variable $X_{\mathcal{L},t|O_t}$, the environmental phenomenon over the set of locations $\mathcal{L}$ at time t given that a set of observations (their spatio-temporal locations) are made at time t (O_t). For simplicity in the notation, and unless stated otherwise, we use $X_y = X_{\mathcal{L},t|\mathbf{O}_{t-1}}$ and $X_A = X_{\mathcal{L},t|O_t}$. Similarly, the realisation of the measurements over the set of locations $\mathcal{L}$ given a set of observations is denoted by $X_A = x_A$. Given the nomenclature above, we can now model the phenomenon.

As explained in Section 2.4.2 and Appendix A, the measurements of an environmental phenomenon can have a multivariate Gaussian joint distribution over all of their locations $\mathcal{L}$ and timesteps $\mathcal{T}$. The main advantages of GPs in environmental monitoring are that they can capture structural correlations of a spatio-temporal phenomenon as well as providing a value of certainty on the predictions, i.e., predictive uncertainty. Crucially, it is sufficient to know the locations of the observations but not the actual value of the measurement, to get the variance over the environment.

Gaussian Processes provide the mathematics of the utility function we need to maximise, as shown in Section 2.4.2 and Appendix A. Similar to the work by Guestrin et al. (2005), we want to maximise the sum of information obtained over time which is captured by the entropy over the entire environment at a specific timestep minus the entropy that can be obtained by taking specific measurements in the next time step over the entire environment.

In other words, our utility function measures the reduction of entropy at all locations of the environment (global metric) by making a set of observations and it is proportional to the uncertainty without making any observations minus the uncertainty when observations are made. This is given by:

$$I(X_y; X_A) = H(X_y) - H(X_y|X_A) \tag{2}$$

In terms of Gaussian Processes, the conditional entropy of a random variable X_y given a set of variables X_A is expressed as follows:

$$\begin{aligned} H(X_y|X_A) &= \frac{1}{2}\log(2\pi e\sigma^2_{X_y|X_A}) \\ H(X_y|X_A) &= \frac{1}{2}\log(\sigma^2_{X_y|X_A}) + \frac{1}{2}(\log(2\pi)+1) \end{aligned} \tag{3}$$

Using a GP to model the environment, we develop an algorithm to exploit predictive uncertainty and the information metric designed.

6. The Coordination Algorithm

Our algorithm is designed to work with hundreds of participants at any given time, with probabilistic information about their mobility patterns and heterogeneity in their budget and reliability.

As discussed in Section 2, finding the optimal solution is computationally infeasible for realistic settings. In this work, we focus on designing an efficient algorithm that outperforms the state of the art. The challenges are the probabilistic nature of human mobility patterns and human reliability as described above, the budget constraints, and the large number of participants. Thus, our algorithm must be able to adapt under uncertainty and be scalable. In this section, we first intuitively explain how our algorithm works and then we provide the formal details.

Our approach, the adaptive Best-Match algorithm or 'aBM' (Algorithm 1), consists of two main components, the offline component, i.e., the Simulations for Scalable Searching (SiScaS) algorithm (Algorithms 2, 3 and 4), and the online component, i.e., the Matching algorithm (Algorithm 5). The offline algorithm is responsible for searching through the space of potential candidate solutions in order to produce a number of mappings of participants to spatio-temporal locations. Specifically, the algorithm makes small changes to the candidate solution (local search), in terms of when and where each user should take a measurement, and evaluates its performance by simulating the environmental campaign. The algorithm is explained in detail in Section 6.1. This algorithm, however, treats spatial clusters of people as a single entity, which speeds up the searching process. The Adapt algorithm, which is part of the offline component and is presented in Section 6.1.2, deals with finding people within a particular cluster who should take a measurement, in order to maximise the expected utility while at the same time saving budget for future iterations.

The next part of the algorithm (presented in Section 6.2) is responsible for acting in real time, matching the simulated output with the current situation. In particular, given the uncertainty in human mobility patterns, users are not guaranteed to be at the locations used in the simulations. Also, offline simulations can typically take a significant amount of time to complete, depending on the number of participants and timesteps. Consequently, there might not be enough time to suggest to people when and where to take measurements. Thus, an algorithm that handles the real time situation is necessary. Our algorithm finds the best match between the simulation output from the offline algorithm to the real-time situation.

Our decision to exploit both offline and online components is due to the fact that the offline algorithm can find good solutions by making assumptions about the uncertainty related to mobility patterns. However, in real-time, the actual location of the users can be observed. Thus, an online algorithm is required to adapt the measurements to be taken, which are produced by the offline component, in order to match the real-time situation and increase the total utility gained. For instance, if the offline component determined that a user who will be in a specific location should take a measurement, but in real-time the user is not there, a nearby available user could potentially take the required measurement instead.

In this section, we present a high-level overview of our algorithm and then focus on each component and subcomponent of it. In particular, the high level structure of our coordination algorithm (aBM) is shown in Algorithm 1. This algorithm shows that given the number of timesteps, the budget of the people and their reliability (line 1), a number of offline simulations (N) are made (line 2). For each simulation (N), a different mapping of users to spatio-temporal locations is produced (S), which we represent with $S_{1,\dots,N}$. Also, a number of spatio-temporal clusters are produced, depending on the spatial locations of people, each of which is associated with the users that belong to it, their coordinates as well as the coordinates of the centroid of the cluster. Formally, $C = \{\mathcal{C}_{1,1}, \dots, \mathcal{C}_{E,m}\}$, where $m \leq M$ and every $\mathcal{C}_{i,j}$ is associated with a number of users $A \subseteq \mathbf{A}$. In other words, it is a set of spatio-temporal clusters that include information about each participant's location that belongs in that cluster, their reliability and budget as well as the centroid in terms of coordinates of each cluster. Then, in real time (represented in lines 3-7), i.e., every timestep i, the Matching algorithm is called to find the best match between simulations and the real-time situation (line 4). Next, the selected users are notified to take the measurements required by the system (line 5). Finally, the environment is updated (line 6) with the information provided by the users.

Algorithm 1 Adaptive Best-Match (aBM) Algorithm

```
1: input: $E$ (timesteps), $B$ (budget), $R$ (reliability)
2: $S_{1,\dots,N}, C_{1,\dots,N} \leftarrow$SiScaS$(E, B, R)$ {Simulations running offline}
3: for $i = 1$ to $E$ do
4:     $S_i^* \leftarrow$ Matching$(E, i, B, S_{1,\dots,N}, C_{1,\dots,N})${Online mapping of users to measurements}
5:     Notification$(S_i^*)${Notify selected users to take measurement}
6:     $\mathcal{E} \leftarrow$ Update$(S_i^*)${The environment is updated with the new information obtained by
       the measurements taken by users at this timestep.}
7: end for
```

6.1 Simulations for Scalable Searching (SiScaS)

The Simulations for Scalable Searching (SiScaS) algorithm is a critical component in our work, as it is responsible for a number of functions including calling the Stochastic Local Greedy Search (SLGS) algorithm, which is described in Algorithm 3. The SiScaS algorithm is shown in Algorithm 2. In particular, this algorithm is responsible for sampling from the human mobility patterns distributions provided by the human mobility prediction component in Figure 3 (line 4), in order to get the possible locations for each of the participants. It also clusters people in spatially correlated groups for all the timesteps using a well-known

Algorithm 2 Simulations for Scalable Searching (SiScaS) Algorithm

```
1: input: $E$ (timesteps), $B$ (budget), $R$ (Reliability)
2: $Simulations = N$ {Number of simulations to run}
3: for $s = 1$ to $Simulations$ do
4:     $A, l \leftarrow SAMPLEHMPs$ {Sample from human mobility patterns distribution where
       $A \subset \mathbf{A}$, and $l \subset L$ their corresponding locations}
5:     $C_s \leftarrow DBSCAN(A, l, E)$
6:     $S_s \leftarrow SLGS(E, C_s, B, R)$
7: end for
8: return: $S_{1,\dots,N}, C_{1,\dots,N}$
```

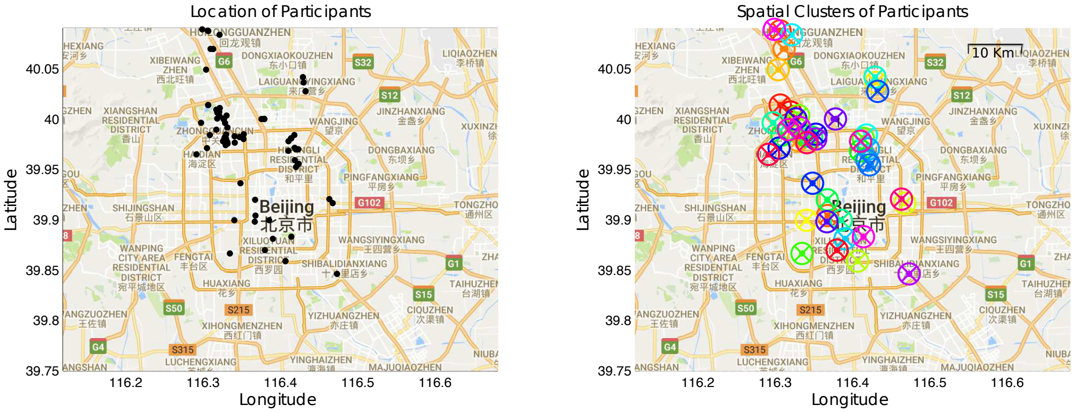
clustering technique called DBSCAN (Ester, Kriegel, Sander, & Xu, 1996)[8] (line 5). DBSCAN enables the grouping of people based on the Euclidean distances between each other at each timestep and is independent of the shape of the cluster. Also, DBSCAN, in contrast to other clustering techniques, does not require an explicit input of the number of clusters that should be formed. Rather, it requires the minimum number of points needed to form a cluster, as well as a distance threshold that prohibits points far apart from each other to belong to the same cluster. Consequently, people close to each other are said to belong to the same cluster and thus can be treated as a single entity, which is crucial in scaling up the number of participants in the campaigns. This is feasible since, in our case, measurements taken at the same spatio-temporal location contribute the same information to the campaign. Since, at each timestep people can be in different locations, the algorithm produces a different set of clusters for each timestep. For example, Figure 4 (a) shows an example of how a hundred people are scattered in an area, which is part of the real human mobility dataset we use for our experiments later on. Figure 4 (b) shows the same 100 people clustered in 47 spatial groups. On average there are 2 people per cluster in this occasion. However, isolated people are in their own cluster and people in more populated areas are grouped together.

Finally, SLGS is called (line 6) and the human mobility patterns as well as the spatio-temporal clusters are passed to it. For each iteration of the algorithm, SLGS will produce a different mapping of participants to measurements, since it will keep sampling from the mobility patterns and forming clusters for a number of times $Simulations = N$.

6.1.1 Stochastic Local Greedy Search (SLGS) Algorithm

The Stochastic Local Greedy Search Algorithm (SLGS) algorithm is the core component of SiScaS (called on line 6 of Algorithm 2). The idea of SLGS is to stochastically evaluate a number of policies (set of spatio-temporal locations of a set of agents), according to the utility function defined in Equation 2, and greedily proceed to a neighbouring policy by applying local changes in order to maximise that function. Thus, given a set of spatio-temporal clusters, the budget of people and a number of timesteps, SLGS finds a mapping between clusters and possible measurements, such that the information about the environ-

[8] Other clustering algorithms such as K-means (MacQueen, 1967), Gaussian Mixture Model (McLachlan & Peel, 2000) or Hierarchical Clustering (Johnson, 1967) could be used here.



(a)

(b)

Figure 4: *Spatial locations of 100 participants in Beijing, showing the locations of individual users (a) and the locations of the means of the clusters created (b)*

ment is maximised. SLGS is able to simulate how the information about the environment is changing over time by exploiting the property of Gaussian Processes that requires only the location of the measurement and not the actual value of it in order to provide the magnitude of uncertainty over the environment.

A key feature of SLGS is that each cluster is treated as a single entity and can only take a single measurement at a time, which is assumed to be taken from its centroid. The reason for this is to avoid using individuals' locations to make our algorithm more efficient. However, the algorithm needs to decide who should actually take the measurement and reduce the budget of the participants accordingly.

To do so we use a greedy algorithm within each cluster (Section 6.1.2), choosing the users that provide the best expected utility, while taking into account their reliability. Intuitively, our approach requires the most reliable people to take the most important measurements. However, calculating the exact utility is intractable for a large number of users. This is because we would have to consider all the combinations of users in a cluster to get the expected utility. To overcome this problem, we calculate the probability that at least one of the selected users in the cluster will take the measurement. This is easy to calculate as it is one minus the product of probabilities of all users not taking a measurement when notified:

$$R^* = 1 - \prod_{j=1}^{W} (1 - r(A_j)) \tag{4}$$

where W is the number of people instructed to take a measurement within a cluster and $A_j \in \mathbf{A}$ the user to take the measurement.

Since the number of spatio-temporal clusters can be large (up to a maximum of the number of participants times the number of timesteps, i.e., $M \cdot E$), we sample again through space and time. That is, we select a random number of clusters $c \subseteq C$ for every timestep. Consequently, we are left with a smaller number of spatio-temporal clusters. We greedily select measurements that maximize the total information. However, in order to save com-

Algorithm 3 Stochastic Local Greedy Search (SLGS)

```
**input**: E (timesteps), C (clusters), $B$ (budget), $R$ (Reliability)
$maxU' = 0$, $S^* \leftarrow null\ matrix(|C|)$
**for** $k = 1$ to $|C|$ **do**
    { For each $k$, an extra spatio-temporal cluster is taking a measurement.}
    **if** $max_i(B_i) == 0$ **then**
        **return**: $S^*$
    **end if**
    $c \leftarrow RANDOMSAMPLE$ {Take a random sample per timestep from the set of clusters available where people have some budget left such that $c \subseteq C$}
    $sz \leftarrow |c|$
    **for** $l = 1$ to $sz$ **do**
        { For each $l$, a different spatio-temporal cluster is taking a measurement.}
        $O' \leftarrow O \cup o_l$ {Where $o_l$ is the extra observation to be taken}
        $\mathbb{U}(\mathbf{O}_E) \leftarrow \sum_{t=1}^{E} u(O'_t)$ {Calculate the utility for every timestep $t$, where $O'$ includes the spatio-temporal measurements selected so far, including a new measurement $l$.}
        $s_l \leftarrow$ getMappings($\mathbb{U}(\mathbf{O}_E), c$) {A function that associates the users in the spatio-temporal cluster with the utility of the measurement taken, i.e., $s_l : c \rightarrow \mathbb{U}(\mathbf{O}_E)$}
    **end for**
    Keep maximum $\mathbb{U}(\mathbf{O}_E)$ of $s_l$ in $maxU$ variable
    Set $S_l$ to be the best configuration of all $s_l$
    $S^* \leftarrow$ Adapt($S_l, R, E$) {Get the subset of users that will be notified to get a measurement}
    Reduce budget from users selected in $S^*$
    $\delta = (maxU - maxU')/maxU$
    **if** $\delta <$ threshold **then**
        **return**: $S^*$
    **end if**
    $maxU' \leftarrow maxU$
**end for**
**return**: $S^*$
```

putation time, we stop the process when the increase in information by taking a specific measurement is below a predefined threshold.

A simple example of how the SLGS algorithm works is presented on Figure 5. Here, for simplicity, we assume that there is a two timestep campaign (t_1 and t_2) with two clusters (C_1 and C_2). The algorithm starts by evaluating the null matrix (no measurements at all) and then adds the single best measurement that produces the highest total utility. However, at this stage the algorithm is unaware who in particular will take the required measurements. A zero value means that no measurement is taken and a one means that a measurement is taken by users in the cluster at that timestep. The algorithm evaluates a number of candidate solutions at each iteration (k), selects (denoted with the bold arrow line) the one that produces the highest utility (U), calculated by Equation 1, and proceeds to the next

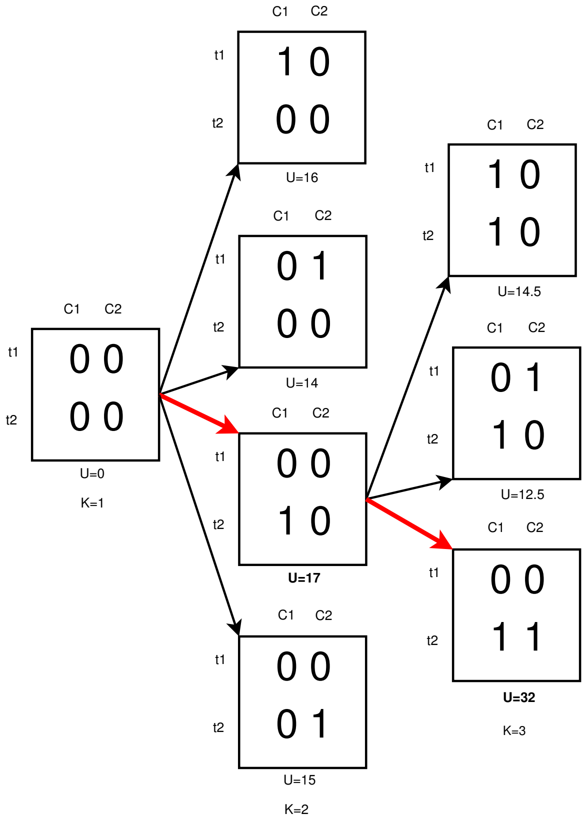


Figure 5: *Schematic representation of an SLGS Algorithm example*

iteration (k), where an additional measurement is added. For instance, when $k = 2$ the maximum utility is gained ($U = 17$) by cluster 1 (C_1) taking a measurement on timestep 2 (t_2). Similarly, when $k = 3$ the maximum utility is gained ($U = 32$) when an additional measurement is taken by cluster 2 (C_2) on timestep 2 (t_2).

Now, the SLGS algorithm, shown in Algorithm 3, is described in more detail. The algorithm accepts the locations of people spatially clustered per timestep (C), as well as the budget of each individual (B), their reliability (R) and the total number of timesteps (E) as shown in line 1. Given that there is sufficient budget left for at least one person in the cluster, it randomly selects a cluster per timestep (line 8). It then checks what the utility would be when adding a measurement from the centroid of each cluster (lines 10-15). This is achieved by forwarding the campaign in time to check what the final utility would be (line 13). This enables the simulations to run fast since not every single position in the cluster is considered by the Gaussian Process. Next, the utility produced by the specific combination of measurements is stored as a mapping from users to spatio-temporal locations in s_l (line 14). Then, the algorithm finds the cluster that produced the highest marginal increase (δ) in utility, given the set of candidate solutions s_l, and selects it (line 16). Since the algorithm is greedy, this measurement can no longer be removed and thus it is not considered in the following iterations. At this point Adapt is called (line 18) in order to select who, within the selected cluster, will actually take the measurement (see Section 6.1.2 for more details).

At the same time, the budgets of people in the cluster selected are adjusted accordingly, i.e., the budget of the people selected within the cluster is reduced by one (line 19). The algorithm iterates until the marginal increase is below a percentage threshold or everyone's budget is depleted (line 21 and line 5 respectively).

In order to speed up our algorithms, we reuse some of the results already calculated by partially evaluating policies in the SLGS algorithm. In particular, at each iteration of policy evaluation in time (line 13), i.e., when forwarding the campaign in time, we store the utility earned from that part of the policy. When this part of the policy appears again, we reuse the utility without the need to re-evaluate it.

6.1.2 Adapt Algorithm

Algorithm 4 is responsible for selecting the people within the cluster that should take the measurement, as explained above. It is called on line 18 of the SLGS algorithm (Algorithm 3).

Algorithm 4 Adapt

```
input: $S_l$ (users in space and time), $r_{1,\dots,M}$ (reliability), E (timesteps)
$maxU' = 0$, $S^* \leftarrow null$
for $f = 1$ to $|S_l|$ do
  $sz = |S_l| - |S^*|$ {People not yet selected}
  for $l = 1$ to $sz$ do
    $R_l = 1 - \prod_{j=1}^{|S^*|+1} (1 - r(A_j))$ {Calculate the probability that at least one user takes a
      measurement according to Equation 4, where $A_j$ are the people selected within the
      cluster including the new measurement $o_l$.}
    $\mathbb{U}(\mathbf{O}_E) \leftarrow R_l \cdot u(O_t)$ {Calculate the overall utility of a set of observations (with and
      without the observations from the given cluster)}
    $s_l \leftarrow$ getMappings($\mathbb{U}(\mathbf{O}_E), A$) {A function that associates the users with the utility
      of the measurement taken, i.e., $s_l : c \rightarrow \mathbb{U}(\mathbf{O}_E)$.}
  end for
  Keep maximum $\mathbb{U}(\mathbf{O}_E)$ of $s_l$ in $maxU$ variable
  Set $S^*$ to be the best configuration of all $s_l$
  $\delta = (maxU - maxU')/maxU$
  if $\delta <$ threshold then
    return: $S^*$
  end if
  $maxU' \leftarrow maxU$ {Update the highest utility}
end for
return: $S^*$
```

This is a greedy algorithm that estimates the utility if at least one of the users that are requested to take the measurement in a particular cluster will actually take the measurement. In other words, it selects a subset of people within the cluster to take a measurement, saving measurements for future iterations. Figure 6 shows an example of how this algorithm works. Specifically, in this example, the utility gained from the cluster taking measurement is

$U = 17$ and reliabilities of the agents are $r(\mathcal{A}_1) = 0.6, r(\mathcal{A}_2) = 0.9, r(\mathcal{A}_3) = 0.8, r(\mathcal{A}_4) = 0.7$. Initially, the measurement to be taken is by cluster 1 (C_1) on timestep 2 (t_2), which is the result of iteration $K = 2$ in Figure 5. The algorithm selects iteratively who in that cluster should take the measurement. For instance, in the first iteration, user 2 ($\mathcal{A}_2$) is selected (bold arrow line) since the total utility, calculated by Equation 4, is greater than any other choice, since it has the highest reliability. Similarly, in the next iteration, the probability of at least one user taking a measurement is calculated. The algorithm adds measurements greedily until utility gained is less than a threshold.

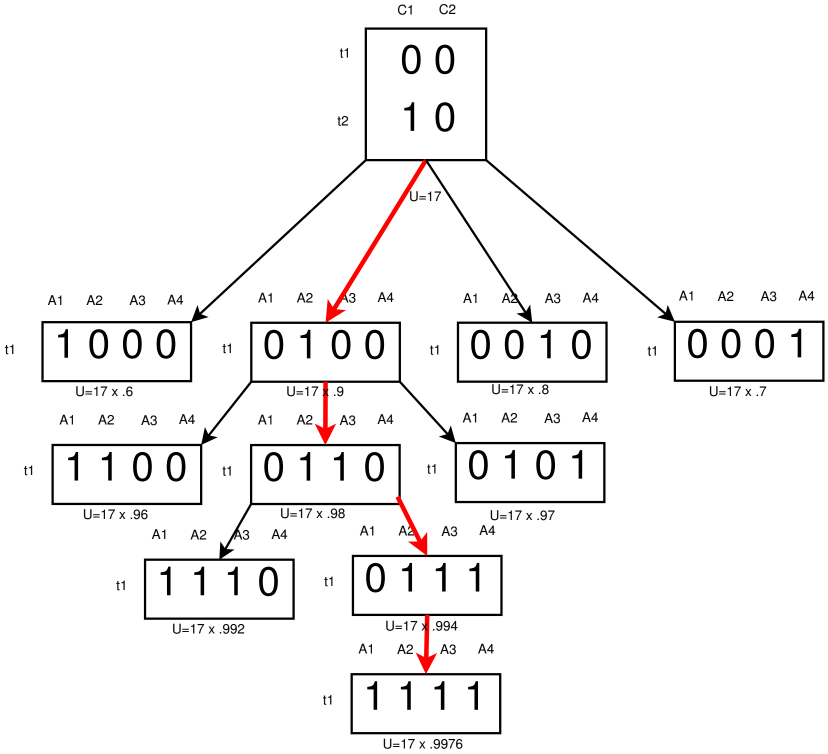


Figure 6: *Adapt Algorithm Example*

Now, Algorithm 4 is explained in more detail. This algorithm iterates through the number of users (line 3-17), who belong in the cluster selected in Algorithm 3. Then, it iteratively adds a user to the list of selected users (line 5-9), calculating the probability (reliability) that at least one of the selected users will actually take the measurement required (line 6). Next, the total utility is calculated (line 7) and this is stored as a mapping from users to spatio-temporal locations in s_l (line 8). Then, the algorithm finds the user that produced the highest marginal increase (δ) in utility, given the set of candidate solutions s_l, and selects it (line 10). Since the algorithm is greedy, this measurement can no longer be removed and thus it is not considered in the following iterations. The algorithm iterates until the marginal increase is below a percentage threshold (line 13).

6.2 The Matching Algorithm

SiScaS (presented in Section 6.1) will produce a number of mappings (N) of participants to measurements depending on the samples taken from human mobility patterns, as well as the clusters that are formed. However, in real time, participants can actually be in a different location or they may not be available at all. Also, people might not take the measurement even if they are actually at the desired position. The idea of the Matching algorithm is to decide who to notify in real time, given the output of SiScaS ($S_{1,\ldots,N}$) and the state of the world at each timestep.

Algorithm 5 Matching algorithm

```
1: **input**: E (timesteps), current (current timestep), B (budget), $S_{1,\ldots,N}, C_{1,\ldots,N}$
2: $A \times l \leftarrow GetHumanLocations$ {Get GPS coordinates of users where $A \subset \mathbf{A}$ and $l \subset L$}
3: $\widehat{C}^{current} \leftarrow DBSCAN(A, l, current)$\{$\widehat{C}^{current}$ are the clusters formed at the current timestep in real-time}
4: **for** $s = 1$ to $N$ **do**
5:     Find nearest neighbour from $\widehat{C}^{current}$ to $C_s^{current}$ {Find the best match between the cluster in real-time ($\widehat{C}^{current}$) and a number of simulations ($C_s^{current}$) at a specific timestep ($current$)}
6:     $D_s \leftarrow$Calculate Euclidean distance of $\widehat{C}_{current}$ nearest neighbour
7: **end for**
8: $ind \leftarrow$ arg $\min_s D_s$ {Get the index of the minimum distance.}
9: $P \leftarrow C_{ind}^{current}${Get people from the best simulation}
10: $\widehat{P} \leftarrow \widehat{C}_{current}${Get people in real time}
11: Find $S^{'} = S^*_{ind} \cap \widehat{P}\{S^*_{ind}$ is the best match between clusters formed in simulations in advance and real-time clusters. $S^{'}$ is a subset of those mappings that includes only those people that are actually available in real time.}
12: Get the people not taking measurements within selected cluster $S^{''} = \widehat{P} \backslash S^{'}$
13: Select $X = |S^*_{ind}| - |S^{'}|$ measurements
14: Append $X$ random measurements to $S^{'}$ from $S^{''}$
15: $M \leftarrow$ Get people with budget left in $ind$ simulation
16: $totalBudget \leftarrow$ Sum the budget left in $ind$ simulation
17: $O = totalBudget/(E - current)$
18: Choose $O$ random measurements from $M$
19: Append new measurements to $S^{'}$
```

Concretely, the Matching algorithm (Algorithm 5) gets human locations (line 2) in real time and clusters them using the DBSCAN algorithm (line 3). Then, the algorithm finds the best match between the measurements that are most informative, as calculated in advance, and the actual positions of participants in real time. Specifically, we find the nearest neighbours from the real-time clusters to the clusters produced in SiScaS (line 5) and then the Euclidean distance between them is calculated (line 6). The smaller the distance, the more similar the clusters are. Then, the simulation that best matches the current situation is found by selecting the smallest distance (D) from all the simulations (line 8). Given what measurements were selected in the simulations in advance, the corresponding people in the cluster are selected (line 9). Then, the people within the real-time cluster are selected (line 10). Since not everyone in the cluster should take a measurement, the algorithm finds whether there are actually users selected in simulations in that cluster (line 11). Given that not everyone in the cluster would have been in the simulation, we randomly select people from the cluster to match the number of users instructed to take the measurement (line 12-14). Next, the people whose budget have not been depleted in the best simulation are retrieved (line 15) and the total budget left is calculated (line 16). In order to evenly distribute the remaining budget, we divide the total budget by the timesteps left (line 17). Then, the algorithm randomly selects measurements to be taken by people whose budget was not depleted in the simulations (line 18). Finally, the randomly added measurements are appended to the previous ones (line 19).

7. Empirical Evaluation

In this section, we evaluate the algorithm developed using real human mobility patterns and air quality sensor data. In the first part, we introduce our benchmarks and give a description of the experiments performed. Finally, we discuss our findings.

7.1 Benchmarks

The algorithm developed was benchmarked against the state-of-the-art algorithms which are introduced below:

- **Greedy:** This algorithm is based on the work by Krause et al.(2008) discussed in Section 2.3. It iterates through possible measurements available at each timestep, finding the one that produces the highest utility. It keeps adding measurements until a budget k is met. In our setting, k is derived from the total budget of people available at each timestep. In particular, we divide the total budget that is available by the number of timesteps left.

- **Best-Match:** This is an algorithm presented in an earlier version of this paper (Zenonos, Stein, & Jennings, 2015a). The Best-Match algorithm works similarly to adaptive Best-Match. However, it is conservative in terms of the measurements taken. Specifically, when a cluster is selected in the simulations, all of the people belonging to that cluster are instructed to take a measurement. In real time, the people belonging to the cluster that matches the offline simulations are again instructed to take the measurement. In doing so, this algorithm does not take in consideration the reliability of users and may exhaust its budget more quickly than our approach.

- **Proximity-driven (Pull-Based):** This algorithm is often used in practice to let people execute tasks based on their spatial location, as described in Section 2.2. In environmental monitoring this can be interpreted as taking measurements when people are in an area of high uncertainty or when the measurement they take has a high utility. This approach is used by the state-of-the-art mobile crowdsourcing applications, such as FieldAgent[9] and GigWalk[10], and it is outlined by Chen et al. (2014).

[9] http://www.fieldagent.co.uk

[10] http://www.gigwalk.com/

- **Random:** This algorithm randomly selects measurements to be taken by people until no budget is left. Specifically, at each timestep, each participant is requested to take a measurement with probability p.
- **Patrol:** The Patrol algorithm takes measurements at all timesteps until everyone's budget is depleted. This algorithm draws on the agent coordination literature (Section 2.1) and in particular on the work by Stranders et al. (2013), where agents continuously take measurements for environmental monitoring.

Also, since the optimal algorithm is computationally infeasible, we developed an upper bound to the algorithm that can be easily calculated. The upper bound is described below:

- **Upperbound:** We relax the assumption that people have a limited budget, we assume full knowledge of human mobility patterns and assume that people are reliable. Thus, all participants are assumed to take measurements at every timestep and the total utility can be trivially calculated.

7.2 Experimental Setup

In order to empirically evaluate our algorithm, we compare its performance against the six algorithms described above. In particular, we focus on air quality in terms of fine particulate matter (PM2.5) in Beijing, where the levels of air pollution are known to be high and thus it is of considerable interest to both the authorities and the people living there. Table 1 shows the air quality index for air quality in Beijing. We use an air quality dataset (Zheng, Liu, & Hsieh, 2013), which contains one year's (2013-2014) fine grained air quality data from static air quality monitoring stations in Beijing. We use this data to train our GP model, and in particular learn the hyperparameters. These include the dynamism of the phenomenon (l_3) and smoothness over latitude and longitude (l_1, l_2). The sensors are scattered in Beijing and take measurements every hour. Figure 7 shows the stations and the state of the environment represented by a GP for a particular timestep. Air quality exhibits spatial variations (PM2.5 is different depending on where you are in Beijing) as well as temporal variations (it is different depending on the time of the day).

Ideally, at the same time the human mobility patterns are learned using a human mobility prediction system. In this work, however, we use data from the Geolife trajectories dataset (Zheng, Zhang, Xie, & Ma, 2009; Zheng, Li, Chen, Xie, & Ma, 2008; Zheng, Xie, & Ma, 2010), which contains sequences of time-stamped locations of 182 people in Beijing over a period of 5 years (2007-2012), as reported by portable GPS devices. We preprocess the dataset, and take the location of each user every ten minutes. We also take patterns of different weeks or months from the same pool of participants' trajectories. This is used as the ground truth to compute the upper bound in our experiments. In order to make the system more realistic, we provide a probability distribution of the users' potential future locations. This is to simulate the behaviour of a real human mobility prediction system that is able to provide us with these probabilities over possible locations. In particular, in this work, we assume that the correct locations have a high probability of being assigned a higher probability than the rest of the locations. Specifically, we create the probability distribution of the locations such that 80% of the time the true location of the people will be allocated a higher probability than the alternative locations. At the same time, 20% of the time the correct location is assigned less probability than a random location from the user's mobility patterns[11]. This is in line with evidence from the human mobility prediction literature (Song, Qu, Blumm, & Barabási, 2010; McInerney, Rogers, & Jennings, 2013a; McInerney et al., 2013b; Baratchi et al., 2014a). In particular, Song et al. (2010) claim that the predictability of human mobility patterns varies very little. Their results show that predictability peaks at 93%, and no users were observed whose predictability was under 80%. However, people have a limited budget of measurements they are willing to take per day. In our work, we assume that people have an average budget of two measurements per day, which is consistent with findings in real participatory sensing systems (Chon et al., 2013)[12]. Also, people may not take the measurement they are requested according the their reliability, as described in Section 4.

The next section presents the results of our experiments. Our experiments involve comparing the execution time of the algorithms and the performance in terms of utility gained (Equation 1 in Section 4) in campaigns with different numbers of participants (up to 1000 per timestep), different time-scales, which affect the dynamism of the phenomenon and different user reliabilities. We compare algorithms in terms of execution time, as the problem we address is NP-hard (Section 2) and thus no optimal solution is tractable. On the other hand, the utility measures the information collected about the phenomenon over time, which is important to understand how the phenomenon changes over time. We also experimented with another evaluation metric, namely Root Mean Square Error (RMSE). Due to the similarity in the results, we do not include those results here. Moreover, different phenomena might have different dynamics. Thus, we examine how algorithms perform under different degrees of dynamism (or time-scale (l_3) in terms of GPs shown in Equation 8). Also, the more people, the more complex the problem becomes in terms of finding the best solution. However, the more people, the less the contribution of each one in terms of information to the overall campaign. Also, as mentioned in Section 3, people are associated with uncertainty about whether they will actually take a measurement when they are asked to do so. In order to examine the robustness of our algorithm, we vary the average reliability of the people between zero and one.

Moreover, in order to obtain statistical significant in our results, we performed two-sided t-test significance testing at the 95% confidence interval. Our experimental platform is the IRIDIS High Performance Computing Facility with 2.6 GHz Intel Sandybridge processors and 64GB RAM per node[13].

7.3 Results

Figure 8 shows results of the performance of the algorithms coordinating a varying number of participants when varying the time-scale, which controls the dynamism of the phenomenon. The smaller the time-scale, the more dynamic the phenomenon. Consequently, as the time-scale approaches zero, the phenomenon rapidly changes over time. In these environments, the adaptive Best-Match algorithm is better, in terms of total utility gained than the rest of the algorithms. The adaptive Best-Match algorithm saves measurements in the simulations

[11] We also experimented with different distributions and we got broadly the same results.

[12] We also experimented with different budgets and we got broadly the same results.

[13] `http://cmg.soton.ac.uk/iridis`

AQI Category	PM2.5 Level	Associated Health Impacts
0-50	Excellent	Little or no risk.
51-100	Moderate	Few hypersensitive individuals should reduce outdoor exercise.
101-150	Unhealthy for Sensitive Groups	Slight irritations may occur.
151-200	Unhealthy	Everyone may begin to experience health effects.
201-300	Very unhealthy	Healthy people will be noticeably affected.
300+	Hazardous	Healthy people will experience reduced endurance in activities.

Table 1: Air Quality Index (AQI) for air pollution (`http://airnow.gov/index.cfm?action=aqibasics.aqi`)

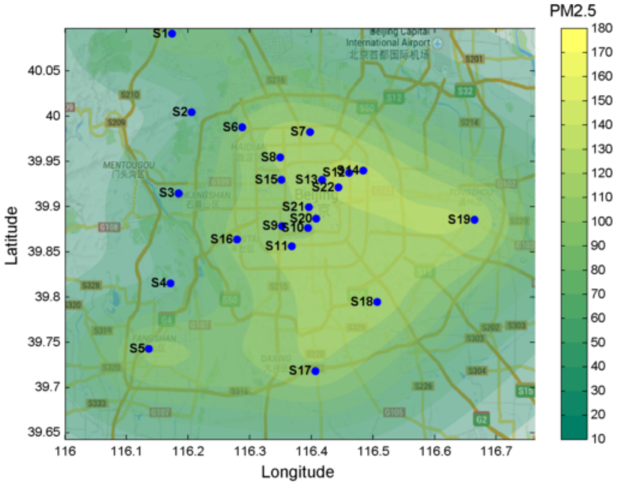
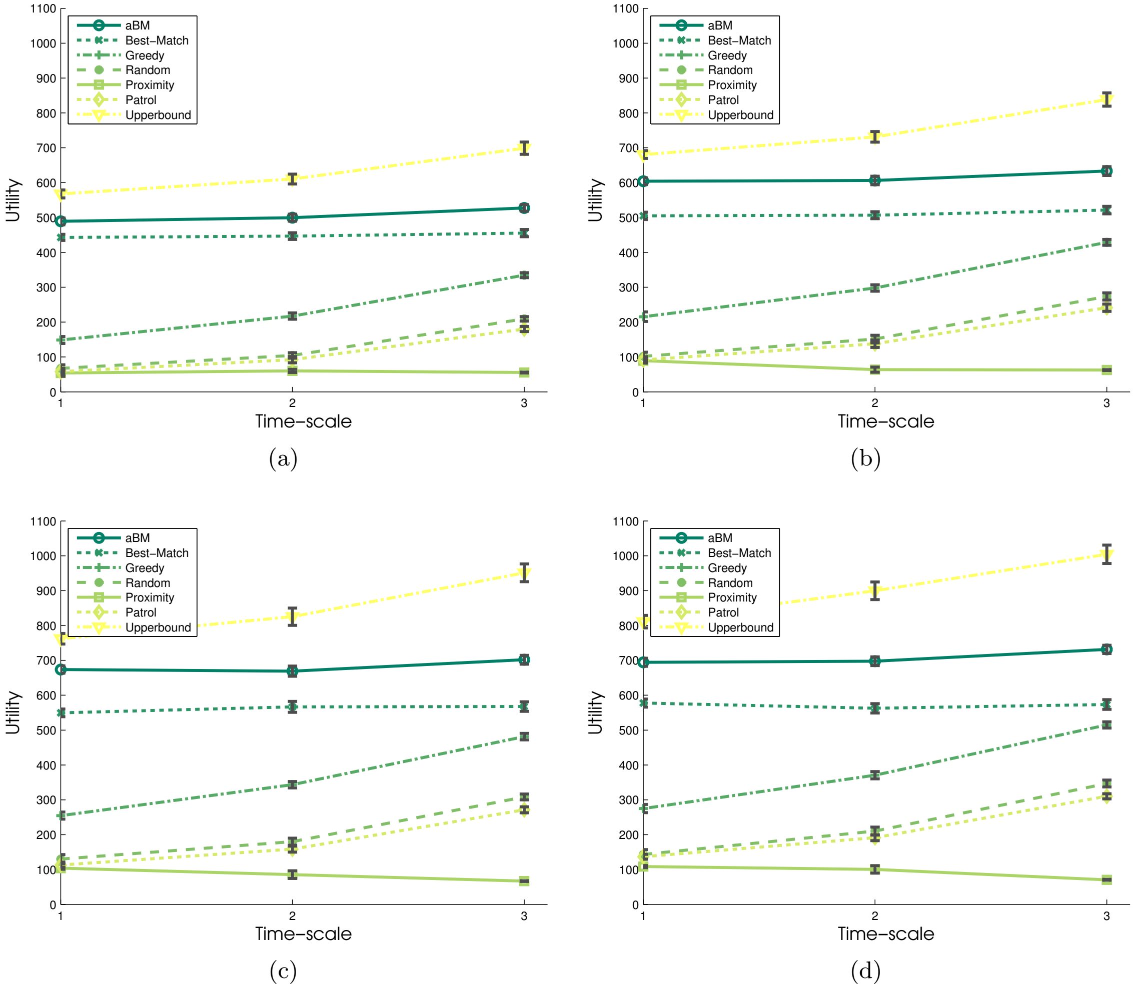


Figure 7: Air quality measurement stations in Beijing overlayed by air quality measurements predicted by GP.

by choosing who specifically should take measurements within the cluster while at the same time maximising the total utility. This allows the algorithm to take extra measurements in real time, which increases the total utility and thereby leads to a higher performance than the Best-Match algorithm. Next is the greedy algorithm. This algorithm is able to choose individuals to take measurements that increase the total utility and that could potentially be in different clusters. However, as will be discussed in detail later, this comes at a great computational expense, as the algorithm needs to consider all the participants one by one until the k best observations are found at each timestep. Also, since it cannot look ahead in time, the algorithm struggles in highly dynamic environments. Specifically, it is possible that some future measurement is more informative if no measurement was taken at that location in the past. The adaptive Best-Match algorithm is designed to produce reasonable outcomes in dynamic environments and it is shown to outperform all the benchmarks in these environments. The proximity algorithm chooses measurements that are informative, since they are above a threshold, but it does not perform well. As we mentioned before, a future measurement might be more informative than the current one. Thus, taking a measurement, which is above the threshold at a timestep, might not be as informative as taking some other measurement in the future. If the threshold is very high, taking only that future measurement might not be as informative as taking a lot of measurements over



(a)

(b)

(c)

(d)

Figure 8: *Total utility gained for 24 timesteps when (a) run with 250, (b) 500, (c) 750 and (d) 1000 participants. The error bars indicate the 95% confidence intervals.*

time. Moreover, it is difficult to define which measurements are informative as the threshold needs to be determined empirically. Patrol is an algorithm that instructs all the users to take all the measurements whenever possible. This means, measurements are taken as early as possible until budgets are depleted. This is not a good strategy as no budget is left later in the campaign. Even a random algorithm is better than patrol since only a random subset of people are taking measurements at each timestep. However, there is no intelligent component that determines how those measurements are taken, and thus uninformative measurements are taken.

In particular, adaptive Best-Match is 12.69% better than the Best-Match algorithm for 250 agents and 2.39 times better than Greedy, 20.29% for 500 agents and 2.11 times better than greedy, 21.45% for 750 agents and 2 times better than greedy. Finally, it is 23.93% for 1000 agents and 94.27% better than greedy. It is consistent for different participants and the results are significant to a 95% confidence level in a two-tailed t-test significance test.

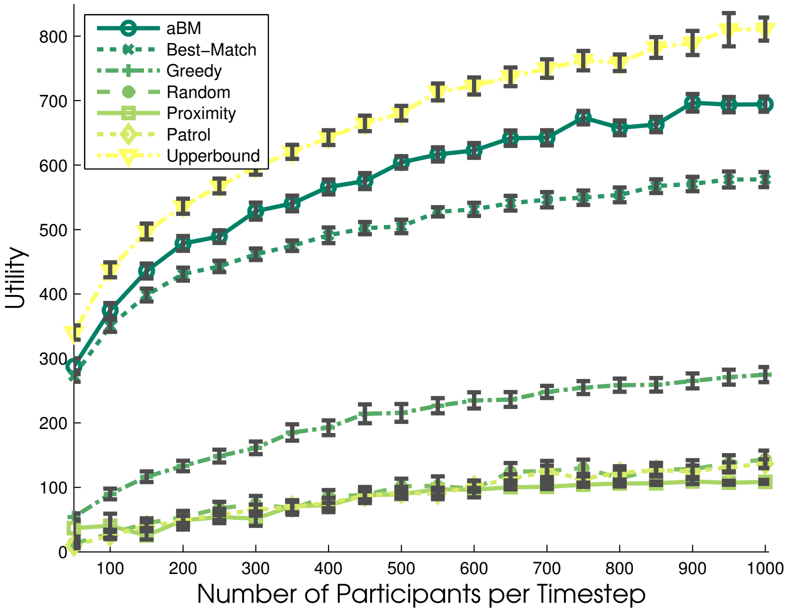


Figure 9: *Total utility for 24 timesteps and a varying number of participants at a constant time-scale of 1. The error bars indicate the 95% confidence interval.*

Figure 9 shows the results of the performance of the algorithms in terms of utility gained when we vary the number of participants (M) in the campaign. The dynamism in this experiment is fixed at 1, to show the performance of the algorithms in a highly dynamic phenomenon. We can observe that adaptive Best-Match is 12.74% better than the Best-Match algorithm and 3.3 times better than the Greedy algorithm for 250 participants. It is 20.31% better than Best-Match and 2.8 times than Greedy with 500 participants. It is 21.43% better than Best-Match and 2.6 times than Greedy for 750 participants. Finally, it is 23.91% better than Best-Match and 2.5 times than Greedy for 1000 participants. Also, we observe that the adaptive Best-Match algorithm is up to 60 better than no coordination, even for small-scale scenarios (50 participants). The results are significant to a 95% confidence level in a two-tailed t-tests significance test. Overall, we can observe that adaptive Best-Match algorithm is significantly better in most scenarios and at least as good as the Best-Match up to 150 users. Crucially, the upperbound is on average only 13.14% better than adaptive Best-Match, which highlights the good performance of our algorithm.

Figure 10 shows the performance of the algorithms in terms of the total runtime when varying the number of participants per timestep. The results show that the adaptive Best-Match algorithm is faster than the Greedy and Proximity-driven algorithms and it is comparable to Best-match. Specifically, it is not significantly different up to 250 agents, but is 40.5% slower for 1000 agents and 30.02% on average. It is evident that adaptive Best-Match and Best-Match algorithms grow linearly with the number of agents. The Proximity-driven and Greedy algorithms require much more time, because as the number of users increases, the number of possible measurements that could be taken is greatly increased. So, the algorithms need to consider one measurement at a time for an increasing set of possible measurements. In fact, the Greedy algorithm is about 50 times slower than the aBM algorithm for 1000 agents. Depending on the number of measurements to be taken at each timestep (k), the Greedy algorithm attempts to find the best measurements by iteratively adding the next single best measurement to the list of measurements to be taken. Similarly, the proximity-driven algorithms evaluates all the possible measurements to decide whether or not that measurement would lead to a higher gain than the pre-defined threshold.

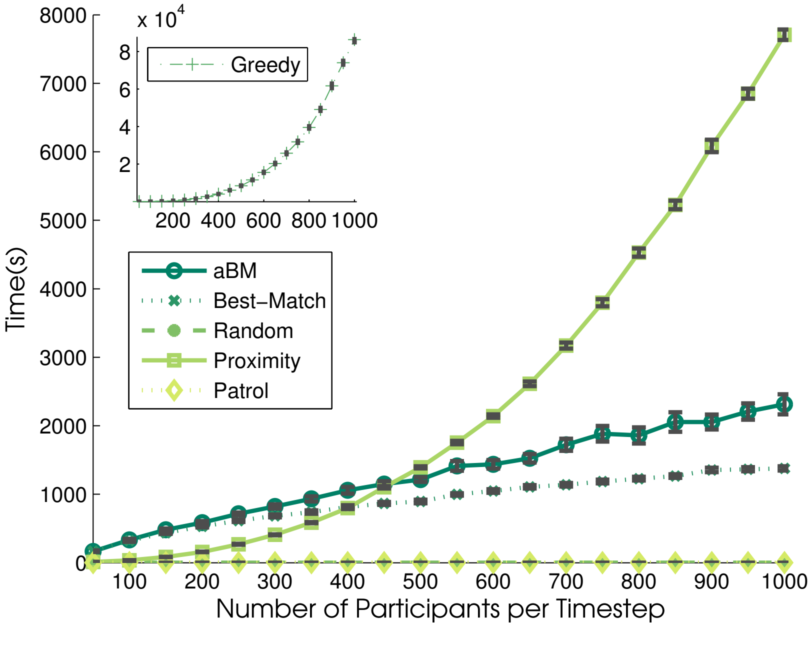


Figure 10: *Average runtime for 24 timesteps and a varying number of participants. The error bars indicate the 95% confidence interval.*

Figure 11 shows the performance of the algorithms in terms of utility when we vary the average reliability of the users. The dynamism is fixed at 1, i.e., a highly dynamic phenomenon and the agent number to 500, which is a representative number such that all

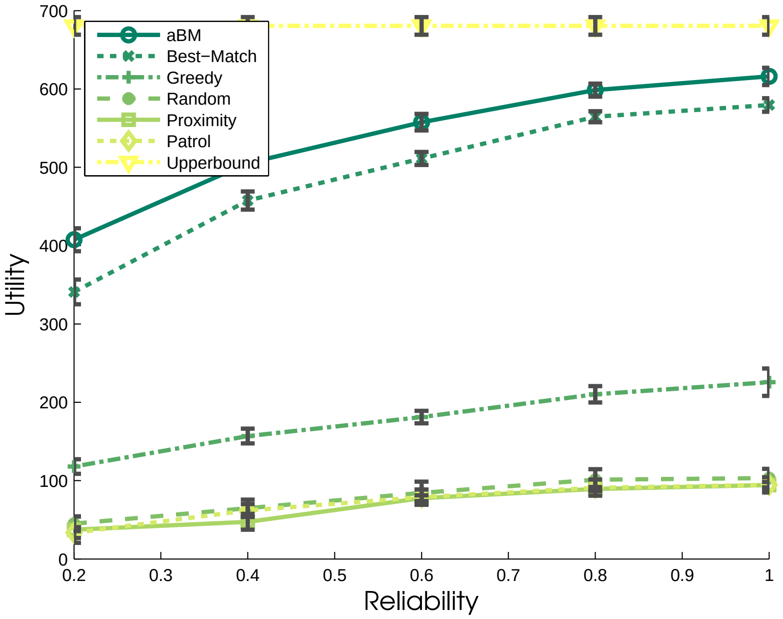


Figure 11: *Total utility for 24 timesteps for 500 agents with a varying reliability at a constant time-scale of 1. The error bars indicate the 95% confidence interval.*

algorithms work efficiently, given the runtime of the algorithms in Figure 10. We observe that adaptive Best-Match is 19.55% better than the Best-Match when reliability is 0.2 and 6.3% when reliability is 1 and 10.27% on average. This is because the offline component of the adaptive Best-Match is able to select more people when reliability is low by choosing the most important measurements in the simulations to be taken by people with the highest reliability. Also, in real time, the online component of the algorithm selects a number of available participants who still have some budget left to take measurements randomly, evenly distributed across the time domain. On the other hand, Best-Match selects all the participants in the best cluster. This makes sure that the most important measurements are taken, but at the cost of using the budget of some people that could potentially have taken other measurements at a different location and time.

8. Conclusions

In this article, we introduced the problem of coordinating measurements in participatory sensing settings under high uncertainty. This uncertainty includes both uncertainty of human mobility patterns and unreliability with respect to taking measurements when requested to do so. We developed a novel algorithm that maximises the total utility gained over a period of time constrained on the number of measurements each user is willing to take. In particular, we demonstrated how efficient the adaptive Best-Match algorithm is compared to the state-of-the-art Best-Match and Greedy algorithms. An empirical evaluation on real data showed that (a) adaptive Best-Match is significantly better than the Best-Match and Greedy algorithms in terms of total utility gained, (b) adaptive Best-Match is significantly faster than the Greedy approach and comparable to the Best-Match one, (c) dynamic environments affect the performance of the adaptive Best-Match algorithm and the total utility gained, but it still outperforms the benchmarks in all scenarios, (d) adaptive Best-Match is significantly better than Best-Match and Greedy algorithm in all scenarios with different degrees of user reliability.

This work constitutes a significant advancement in the area of artificial intelligence, as our algorithm can be used in other applications beyond environmental monitoring. In particular, this work focuses on an entropy-based criterion as a utility function, which is the difference in the information between two timesteps. However, a new utility function for other applications can be devised. For example, in a crowdsourcing classification system, users could be asked to verify objects or events (e.g., traffic jams, vandalism or littering), which are classified from a machine vision algorithm, by physically visiting those locations. The utility in this scenario could capture how valuable human input is. For instance, verifying a rare event of vandalism at a specific location could be more important than verifying a traffic jam in a usually busy area. Our algorithm could be used to decide which users to ask to increase the overall system's efficiency.

In the future, we plan to extend our algorithm to deal with corrupted or false user readings. The current system values information only based on when and where measurements are taken, without considering the trustworthiness of each user. However, in realistic settings, there might be a tendency for malicious users to falsify their contribution for their own selfish reasons. For instance, coordinated malicious participants might falsify air quality readings near a factory to hide a potential air pollution issue or provide higher readings in another area, where a competitor's factory is located. Moreover, more sophisticated attacks could attempt to falsify their readings in some areas to steer urban planning decisions like the construction of new roads and parks. We have started addressing this setting in related work (Zenonos, Stein, & Jennings, 2017), but there are still open research challenges that need to be addressed. Specifically, that work does not consider sophisticated, systematic attacks, which will be the focus of our future work in this space.

Also, it would be interesting to explore other variations of our algorithm. Specifically, if the cluster distances are small, then instead of choosing users based on the highest expected utility, it could be better to choose based on those who have the largest budget. Also, we can further explore the insights we can get from human mobility patterns, as they can potentially be clustered in space and time, which could be exploited to speed up the algorithm. Moreover, we will consider running a real-world user trial as it could provide additional insights to guide the algorithm.

Acknowledgments

The authors would like to acknowledge the use of the IRIDIS High Performance Computing Facility, and associated support services at the University of Southampton, in the completion of this work. Also, they acknowledge the funding from the UK Research Council for the ORCHID project, grant EP/I011587/1. Preliminary findings of this work appeared at AAAI (Zenonos et al., 2015a) and AAMAS (Zenonos, Stein, & Jennings, 2015b).

Appendix A. Gaussian Processes Regression

The purpose of this Appendix is to provide the details of Gaussian Processes (GPs) in relation to the environmental monitoring. First, we provide the nomenclature, and then a technical description of GPs.

Let $\mathbf{x}_*$ be the input vector (test data) and y_* its corresponding output value (prediction). In an environmental context, $\mathbf{x}_*$ represents a single location on the map described by the spatial (x_1, x_2) and temporal coordinates (x_3). The output value (y_*) is the prediction for the actual value (e.g., the air or noise pollution level) at that specific spatio-temporal location represented by $\mathbf{x}_*$. Also, let $y = f(x)$ be a process that denotes the relationship of a $n \times D$-dimensional input vector $\mathbf{x} \in \mathbb{R}^D$ and an output variable $y \in \mathbb{R}$. When representing spatio-temporal phenomena, we have $D = 3$ to capture longitude, latitude and time. In addition, let $\{(\mathbf{x}_i, y_i) \mid i = 1 \ldots n\}$ denote a set of input-output pairs which represents past observations of the process f (training data). In terms of environmental phenomena, the training data is the set of known spatio-temporal locations where measurements were taken in the past with the corresponding value of the measurement at that time. Finally, we denote the collection of n-dimensional output vectors y_i as $\mathbf{y}$. In other words, $\mathbf{y}$ is the output for n locations. Also, we denote the D-dimensional input matrix as X which is a collection of n $\mathbf{x}_i$ (i.e., n rows).

Gaussian Processes are defined as a collection of random variables, any finite number of which have a joint Gaussian distribution. In practice, a GP is completely specified by its mean function and covariance function (or kernel). A mean function $m(\mathbf{x})$ and a covariance function $k(\mathbf{x}\text{,}\mathbf{x}')$ of a real process $f(\mathbf{x})$ are defined as follows:

$$\begin{aligned} m(\mathbf{x}) \quad &= \mathbb{E}[f(\mathbf{x})], \\ k(\mathbf{x}\text{,}\mathbf{x}') &= \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))] \end{aligned} \tag{5}$$

where $\mathbb{E}[\mathbf{X}]$ is the expectation of a random variable $\mathbf{X}$. Thus, we can write a Gaussian Process as follows:

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}\text{,}\mathbf{x}')) \tag{6}$$

The kernel k plays a critical role in Gaussian processes. It determines the covariance between $f(\mathbf{x})$ and $f(\mathbf{x}')$. In other words, it specifies the relationship between two outputs with respect to their associated input. This enables GPs to identify the covariance between the outputs of training data, test data and the combination of both, which gives the predictive power of GPs as shown below. When $m(\mathbf{x})$ and $k(\mathbf{x}, \mathbf{x}')$ are known, they function as a prior over function f. However, when new observations are made, a GP can be updated to fit these data, increasing the prediction accuracy at the unobserved locations.

In GPs a key assumption is that data can be represented as a sample from a multivariate Gaussian distribution. This is expressed as:

$$\begin{bmatrix} \mathbf{y} \\ y_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X) & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right) \tag{7}$$

where $K(\cdot,\cdot)$ are obtained by evaluating the covariance function k for all pairs of columns. X represents the input vector of training data and X_* the input vector of test data. For simplicity in notation, we set $K(X,X) = K$, $K(X,X_*) = K_*^T$, $K(X_*,X) = K_*$ and $K(X_*,X_*) = K_{**}$.

Essentially, K is the covariance of the training points and K_* for all the pairs of training and test points. For the purposes of environmental monitoring, we are interested in the conditional probability $p(y_*|\mathbf{y})$. In other words, given a set of observations $\mathbf{y}$, how likely is a certain prediction for y_*. Using the properties of the Gaussian distribution, we obtain:

$$y_*|\mathbf{y} \sim \mathcal{N}\left(K_*K^{-1}\mathbf{y}, K_{**}K^{-1}K_*^T\right) \tag{8}$$

Thus, the best estimate for y_* is the mean of the distribution and the uncertainty about the estimation is the variance as shown below:

$$\begin{aligned} \mu &= K_*K^{-1}\mathbf{y} \\ \Sigma &= K_{**} - K_*K^{-1}K_*^T \end{aligned} \tag{9}$$

An important property that we exploit in this work is that the covariance of the prediction outputs $\mathbf{y}_*$ does not depend on the actual value of the observations $\mathbf{y}$ made, but rather only on the input vectors X, which are the spatio-temporal locations of those observations. This enables us to run simulations forward in time as it is not necessary to know the actual value of the measurements to estimate the variance at a future timestep. We will come back to this property when dealing with the algorithm developed that exploits it (see Section 6.1), since it is crucial to the evaluation of our utility function in our simulations.

As we have already mentioned, the covariance function is of critical importance. However, we have not yet examined how it is mathematically expressed. A popular choice of such a function is the *Matérn*, which is commonly used for spatial statistics and geostatistics (Jutzeler et al., 2014; Ouyang et al., 2014; Minasny & McBratney, 2005). Importantly, this kernel has free parameters which control the smoothness of the function, or in our context the dynamism of the phenomenon, as well as its sensitivity to measurements and noise. Matérn is defined as follows:

$$k(x,x') = \sigma_f^2(1+\sqrt{3}r)\exp(-\sqrt{3}r) + \sigma_n^2\delta_{x,x'} \tag{10}$$

where $r = \sqrt{(x-x')^T\mathbf{P}^{-1}(x-x')}$, $\mathbf{P} = \begin{bmatrix} l_1 & 0 & 0 \\ 0 & l_2 & 0 \\ 0 & 0 & l_3 \end{bmatrix}$

and $\theta = \{l_1, l_2, l_3, \sigma_f^2, \sigma_n^2\}$ are the hyperparameters that need to be learned. Specifically, l_1 is the length-scale that controls the smoothness of the regression function over the x-axis (longitude), l_2 over the y-axis (latitude) and l_3 over time. Intuitively, (l_1, l_2, l_3) captures the dynamism of the phenomenon in both the spatial and the temporal dimension. Also, σ_f^2 is the signal variance that controls the uncertainty about predictions made further away from the observed points and σ_n^2 is the noise variance that controls the percentage of the data variation that can be attributed to noise.

However, there is no standard kernel that should be used in all applications. Rather, domain-specific knowledge should be taken into consideration when a specific kernel is chosen. For example, an ideal kernel for air pollution would be a non-stationary[14] one

[14] Non-stationary covariance functions allow the model to adapt to functions whose smoothness varies with the inputs. A stationary kernel is one where covariance only depends on distances between points (Paciorek & Schervish, 2004).

that considers air dispersion and mathematically captures the dynamics of air pollution particles. However, often this is difficult to capture and, worse, such kernels require many parameters and are computationally expensive to compute (Ouyang et al., 2014). Designing a kernel for a particular application is outside the scope of this work. Therefore, our choice of kernel is the Matérn described above, which is able to capture the potential non-linear relationship of the phenomenon and allows for smooth interpolation between points. Also, it is empirically shown to work well in related work (Jutzeler et al., 2014).

Estimating θ is equivalent to finding a value for θ that results in a high $p(\theta|\mathbf{x},\mathbf{y})$. In practice, it is achieved by maximising the log marginal likelihood $\log p(\theta|\mathbf{x},\mathbf{y})$. This is given by:

$$\log p(\theta|\mathbf{x},\mathbf{y}) = -\frac{1}{2}\mathbf{y}^T K^{-1}\mathbf{y} - \frac{1}{2}\log|K| - \frac{n}{2}\log 2\pi \tag{11}$$

References

Albers, A., Krontiris, I., Sonehara, N., & Echizen, I. (2013). Coupons as monetary incentives in participatory sensing. In Douligeris, C., Polemi, N., Karantjias, A., & Lamersdorf, W. (Eds.), *Collaborative, Trusted and Privacy-Aware e/m-Services*, Vol. 399 of *IFIP Advances in Information and Communication Technology*, pp. 226–237. Springer Berlin Heidelberg.

Anderson, A., Huttenlocher, D., Kleinberg, J., & Leskovec, J. (2013). Steering user behavior with badges. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pp. 95–106, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Baratchi, M., Meratnia, N., Havinga, P. J. M., Skidmore, A. K., & Toxopeus, B. A. K. G. (2014a). A hierarchical hidden semi-markov model for modeling mobility data. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '14, pp. 401–412, New York, NY, USA. ACM.

Baratchi, M., Meratnia, N., Havinga, P. J. M., Skidmore, A. K., & Toxopeus, B. A. K. G. (2014b). A hierarchical hidden semi-markov model for modeling mobility data. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '14, pp. 401–412, New York, NY, USA. ACM.

Binney, J., Krause, A., & Sukhatme, G. S. (2010). Informative path planning for an autonomous underwater vehicle. In *Robotics and automation (icra), 2010 IEEE international conference on*, pp. 4791–4796. IEEE.

Brown, A., Franken, P., Bonner, S., Dolezal, N., & Moross, J. (2016). Safecast: successful citizen-science for radiation measurement and communication after fukushima. *Journal of Radiological Protection*, *36*(2), S82.

Burke, J., Estrin, D., Hansen, M., Parker, A., Ramanathan, N., Reddy, S., & Srivastava, M. B. (2006). Participatory sensing. In *Workshop on World-Sensor-Web: Mobile Device Centric Sensor Networks and Applications*, pp. 117–134.

Chen, C., Cheng, S.-F., Gunawan, A., Misra, A., Dasgupta, K., & Chander, D. (2014). Traccs: Trajectory-aware coordinated urban crowd-sourcing. In *Second AAAI Conference on Human Computation & Crowdsourcing (HCOMP)*, pp. 30–40.

Chen, C., Cheng, S.-F., Lau, H. C., & Misra, A. (2015). Towards city-scale mobile crowdsourcing: Task recommendations under trajectory uncertainties. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pp. 1113–1119. AAAI Press.

Chepesiuk, R. (2005). Decibel hell: The effects of living in a noisy world. *Environmental health perspectives*, A35–A41.

Chon, Y., Lane, N. D., Kim, Y., Zhao, F., & Cha, H. (2013). A large-scale study of mobile crowdsourcing with smartphones for urban sensing applications. *Proc. of ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp'13), Zurich, Switzerland.*

D'Hondt, E., Stevens, M., & Jacobs, A. (2013). Participatory noise mapping works! an evaluation of participatory sensing as an alternative to standard techniques for environmental monitoring.. *Pervasive and Mobile Computing*, *9*(5), 681–694.

DiPalantino, D., Karagiannis, T., & Vojnovic, M. (2010). Individual and collective user behavior in crowdsourcing services. Tech. rep. MSR-TR-2010-59. `https://www.microsoft.com/en-us/research/publication/individual-and-collective-user-behavior-in-crowdsourcing-services/`.

Endsley, M. R. (1988). Situation awareness global assessment technique (sagat). In *Proceedings of the IEEE 1988 National Aerospace and Electronics Conference*, pp. 789–795 vol.3.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pp. 226–231. AAAI Press.

Gao, H., Liu, C. H., Wang, W., Zhao, J., Song, Z., Su, X., Crowcroft, J., & Leung, K. K. (2015). A survey of incentive mechanisms for participatory sensing. *IEEE Communications Surveys Tutorials*, *17*(2), 918–943.

Garg, S., Singh, A., & Ramos, F. (2012). Efficient space-time modeling for informative sensing. In *Proceedings of the Sixth International Workshop on Knowledge Discovery from Sensor Data*, SensorKDD '12, pp. 52–60, New York, NY, USA. ACM.

Golovin, D., & Krause, A. (2011). Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *J. Artif. Int. Res.*, *42*(1), 427–486.

Guestrin, C., Krause, A., & Singh, A. P. (2005). Near-optimal sensor placements in gaussian processes. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pp. 265–272, New York, NY, USA. ACM.

Hollinger, G., & Singh, S. (2008). Proofs and experiments in scalable, near-optimal search by multiple robots. *Proceedings of Robotics: Science and Systems IV, Zurich, Switzerland*, *1*.

Hoos, H., & Stützle, T. (2004). *Stochastic Local Search: Foundations & Applications*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Jaimes, L., Vergara-Laurens, I., & Labrador, M. (2012). A location-based incentive mechanism for participatory sensing systems with budget constraints. In *Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on*, pp. 103–108.

Jennings, N. R., Moreau, L., Nicholson, D., Ramchurn, S., Roberts, S., Rodden, T., & Rogers, A. (2014). Human-agent collectives. *Commun. ACM*, *57*(12), 80–88.

Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika*, *32*(3), 241–254.

Jutzeler, A., Li, J. J., & Faltings, B. (2014). A region-based model for estimating urban air pollution. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, pp. 424–430.

Krause, A., Guestrin, C., Gupta, A., & Kleinberg, J. (2006). Near-optimal sensor placements: Maximizing information while minimizing communication cost. In *Proceedings of the 5th International Conference on Information Processing in Sensor Networks*, IPSN '06, pp. 2–10, New York, NY, USA. ACM.

Krause, A., Singh, A., & Guestrin, C. (2008). Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *J. Mach. Learn. Res.*, *9*, 235–284.

Low, K. H., Dolan, J. M., & Khosla, P. (2011a). Active markov information-theoretic path planning for robotic environmental sensing. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pp. 753–760. International Foundation for Autonomous Agents and Multiagent Systems.

Low, K. H., Dolan, J. M., & Khosla, P. (2011b). Active markov information-theoretic path planning for robotic environmental sensing. In *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '11, pp. 753–760, Richland, SC.

Mabahwi, N. A. B., Leh, O. L. H., & Omar, D. (2014). Human health and wellbeing: Human health effect of air pollution. *Proceedings of Social and Behavioral Sciences*, *153*, 221 – 229. International Conference on Quality of Life, The Pacific Sutera Hotel, Sutera Harbour, Kota Kinabalu, Sabah, Malaysia, 4-5 January 2014.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pp. 281–297, Berkeley, Calif. University of California Press.

Marchant, R., & Ramos, F. (2012). Bayesian optimisation for intelligent environmental monitoring. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2242–2249. IEEE.

McInerney, J., Rogers, A., & Jennings, N. R. (2013a). Learning periodic human behaviour models from sparse data for crowdsourcing aid delivery in developing countries. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 401–410.

McInerney, J., Stein, S., Rogers, A., & Jennings, N. R. (2013b). Breaking the habit: Measuring and predicting departures from routine in individual human mobility. *Pervasive Mob. Comput.*, *9*(6), 808–822.

McLachlan, G. J., & Peel, D. (2000). *Finite mixture models*. Wiley series in probability and statistics. J. Wiley & Sons, New York.

Minasny, B., & McBratney, A. B. (2005). The matern function as a general model for soil variograms. *Geoderma*, *128*(34), 192 – 207. Pedometrics 2003.

Ministry of Housing Communities, . L. G. (2014). Air quality. `http://planningguidance.communities.gov.uk/blog/guidance/air-quality/when-could-air-quality-be-relevant-to-a-planning-decision/`. [Online; accessed 19-January-2018].

Musthag, M., & Ganesan, D. (2013). Labor dynamics in a mobile micro-task market. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pp. 641–650, New York, NY, USA. ACM.

Nemhauser, G., Wolsey, L., & Fisher, M. (1978). An analysis of approximations for maximizing submodular set functionsi. *Mathematical Programming*, *14*(1), 265–294.

Ouyang, R., Low, K. H., Chen, J., & Jaillet, P. (2014). Multi-robot active sensing of non-stationary gaussian process-based environmental phenomena. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*, AAMAS '14, pp. 573–580, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Paciorek, C. J., & Schervish, M. J. (2004). Nonstationary covariance functions for gaussian process regression. *Advances in neural information processing systems*, *16*, 273–280.

Padhy, P., Dash, R. K., Martinez, K., & Jennings, N. R. (2010). A utility-based adaptive sensing and multihop communication protocol for wireless sensor networks. *ACM Trans. Sen. Netw.*, *6*(3), 27:1–27:39.

Paoletti, E., Bardelli, T., Giovannini, G., & Pecchioli, L. (2011). Air quality impact of an urban park over time. *Procedia Environmental Sciences*, *4*, 10 – 16.

Pineau, J., Gordon, G., & Thrun, S. (2006). Anytime point-based approximations for large pomdps. *Journal of Artificial Intelligence Research*, *27*, 2006.

Ramchurn, S. D., Mezzetti, C., Giovannucci, A., Rodriguez-Aguilar, J. A., Dash, R. K., & Jennings, N. R. (2009). Trust-based mechanisms for robust and efficient task allocation in the presence of execution uncertainty. *J. Artif. Int. Res.*, *35*(1), 119–159.

Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. The MIT Press.

Reddy, S., Estrin, D., Hansen, M., & Srivastava, M. (2010). Examining micro-payments for participatory sensing data collections. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, Ubicomp '10, pp. 33–36, New York, NY, USA. ACM.

Sahami Shirazi, A., Henze, N., Dingler, T., Pielot, M., Weber, D., & Schmidt, A. (2014). Large-scale assessment of mobile notifications. *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems - CHI '14*, 3055–3064.

Sánchez-González, J., Pérez-Romero, J., Agustí, R., & Sallent, O. (2016). *On Learning Mobility Patterns in Cellular Networks*, pp. 686–696. Springer International Publishing, Cham.

Schwager, M., Dames, P., Rus, D., & Kumar, V. (2017). *A Multi-robot Control Policy for Information Gathering in the Presence of Unknown Hazards*, pp. 455–472. Springer International Publishing, Cham.

Seaton, A., Godden, D., MacNee, W., & Donaldson, K. (1995). Particulate air pollution and acute health effects. *The Lancet*, *345*(8943), 176 – 178.

Seinfeld, J. H., & Pandis, S. N. (2012). *Atmospheric chemistry and physics: from air pollution to climate change*. John Wiley & Sons.

Singh, A., Krause, A., Guestrin, C., & Kaiser, W. J. (2009). Efficient informative sensing using multiple robots. *J. Artif. Int. Res.*, *34*(1), 707–755.

Song, C., Qu, Z., Blumm, N., & Barabási, A.-L. (2010). Limits of predictability in human mobility. *Science*, *327*(5968), 1018–1021.

Stevens, M. (2012). *Community memories for sustainable societies: The case of environmental noise*. Ph.D. thesis.

Stevens, M., & D'Hondt, E. (2010). Crowdsourcing of Pollution Data using Smartphones. In Vukovic, M., Kumara, S., & Greenshpan, O. (Eds.), *Workshop on Ubiquitous Crowdsourcing, held at Ubicomp '10 (September 26-29, 2010, Copenhagen, Denmark)*.

Stranders, R., Farinelli, A., Rogers, A., & Jennings, N. R. (2009). Decentralised coordination of continuously valued control parameters using the max-sum algorithm. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '09, pp. 601–608, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Stranders, R., Munoz De Cote, E., Rogers, A., & Jennings, N. R. (2013). Near-optimal continuous patrolling with teams of mobile information gathering agents. *Artif. Intell.*, *195*, 63–105.

Stranders, R., Fave, F. M. D., Rogers, A., & Jennings, N. (2010). A decentralised coordination algorithm for mobile sensors. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, pp. 874–880. Event Dates: 11 - 15 July, 2010.

Thepvilojanapong, N., Tsujimori, T., Wang, H., Ohta, Y., Zhao, Y., & Tobe, Y. (2013). Impact of incentive mechanism in participatory sensing environment. In *SMART 2013, The Second International Conference on Smart Systems, Devices and Technologies*, pp. 87–92.

Tiwari, K., Honoré, V., Jeong, S., Chong, N. Y., & Deisenroth, M. P. (2016). Resource-constrained decentralized active sensing for multi-robot systems using distributed gaussian processes. *The 16th Conference on Control, Automation and Systems*, 13–18.

WAQI (2015). UNEP Air Quality Monitoring Station. http://aqicn.org/faq/2015-10-28/unep-air-quality-monitoring-station/. [Online; accessed 19-January-2018].

Whitney, M., & Richter Lipford, H. (2011). Participatory sensing for community building. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '11, pp. 1321–1326, New York, NY, USA. ACM.

WHO (2014). 7 million premature deaths annually linked to air pollution. http://www.who.int/mediacentre/news/releases/2014/air-pollution/en/. [Online; accessed 19-January-2018].

World-Bank (2016). *The cost of air pollution : strengthening the economic case for action.* World Bank Group, Washington, D.C.

Zenonos, A., Stein, S., & Jennings, N. R. (2015a). An algorithm to coordinate measurements using stochastic human mobility patterns in large-scale participatory sensing settings. In *Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*, pp. 3936–3942.

Zenonos, A., Stein, S., & Jennings, N. R. (2015b). Coordinating measurements for air pollution monitoring in participatory sensing settings. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '15, pp. 493–501, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Zenonos, A., Stein, S., & Jennings, N. R. (2017). A trust-based coordination system for participatory sensing applications. In *Proceedings of the 5th AAAI Conference on Human Computation and Crowdsourcing.*

Zheng, Y., Li, Q., Chen, Y., Xie, X., & Ma, W.-Y. (2008). Understanding mobility based on gps data. In *Proceedings of the 10th International Conference on Ubiquitous Computing*, UbiComp '08, pp. 312–321, New York, NY, USA. ACM.

Zheng, Y., Liu, F., & Hsieh, H.-P. (2013). U-air: When urban air quality inference meets big data. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pp. 1436–1444, New York, NY, USA. ACM.

Zheng, Y., Xie, X., & Ma, W.-Y. (2010). Geolife: A collaborative social networking service among user, location and trajectory.. *IEEE Data Eng. Bull.*, *33*(2), 32–39.

Zheng, Y., Zhang, L., Xie, X., & Ma, W.-Y. (2009). Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pp. 791–800, New York, NY, USA. ACM.

Zilli, D., Parson, O., Merrett, G. V., & Rogers, A. (2013). A hidden markov model-based acoustic cicada detector for crowdsourced smartphone biodiversity monitoring. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI'13, pp. 2945–2951. AAAI Press.