Data Management in Participatory Sensing

THÈSE N° 6530 (2015)

PRÉSENTÉE LE 27 MARS 2015
À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS
LABORATOIRE DE SYSTÈMES D'INFORMATION RÉPARTIS
PROGRAMME DOCTORAL EN INFORMATIQUE ET COMMUNICATIONS

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Mehdi RIAHI

acceptée sur proposition du jury:

Prof. W. Zwaenepoel, président du jury
Prof. K. Aberer, directeur de thèse
Prof. B. Faltings, rapporteur
Prof. I. Podnar, rapporteuse
Prof. A. Zaslavsky, rapporteur

EPFL

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2015

Acknowledgements

First and foremost, I would like to express my gratitude to my thesis supervisor Prof. Karl Aberer. I am thankful to him for giving me the opportunity to do my PhD in LSIR and supporting me throughout my PhD.

I am very thankful to the members of my thesis committee: Prof. Boi Faltings, Prof. Arkady Zaslavsky, Prof. Ivana Podnar, and Prof. Willy Zwaenepoel for their important comments and discussions to improve my dissertation.

I would like to thank my colleagues with whom I collaborated during the work on this thesis, especially Thanasis Papaioannou, Rameez Rahman, Robert Gwadera, Jean-Paul Calbimonte, and Immanuel Trummer. I thank all my great colleagues from LSIR. A special thanks goes to Chantal who helped me sort out so many not only administrative issues.

I would like to thank all my friends from the doctoral school and beyond, for their support and for all the great moments we spent together: Ali, Timothée, Rammohan, Surrender, Saket, Hoyoung, Dipanjan, Alexandra, Tri, Michele, Jean-Eudes, Amitabha, Mia, Pamela, Berker, Alevtina, and many others.

Finally, I would like to thank my parents for their love and support, and for the many sacrifices that they have had to make.

Abstract

In recent years there has been a proliferation of privately owned sensing devices such as GPS devices, cameras, home weather stations and, more importantly, smart-phones. Most of these devices are either intrinsically mobile, e.g., smart-phones and GPS devices, or can be easily carried by people during their daily activities. Nowadays, it is possible to embed various sensors in small devices as the result of sensor technology advancement. For example, we can consider smart-phones as sensing devices because they are equipped with several sensors such as GPS, accelerometer, gyroscope, microphone, and proximity sensors.

This provides an unprecedented opportunity for a new application paradigm called *participatory sensing*, in which people collect and share sensing data about some phenomenon of interest in their environment. This unique opportunity is mainly due to (I) the ubiquity of smart-phones with various built-in sensors, (II) the availability of small, low-cost and pluggable sensors, and (III) the easy access to various connectivity media such as 3G, 4G, and WiFi. However, for using the full potential of participatory sensing, several challenges exist that must be addressed. These challenges include, but are not limited to, privacy protection of participants, quality assessment of collected data, efficient energy consumption of sensing devices, data unavailability due to uncontrolled mobility of the participants, and efficiently incentivizing people to participate. In this thesis we propose methods for addressing some of these issues. In particular, this thesis addresses the following topics:

Efficient Data Acquisition in Participatory Sensing. In participatory sensing systems participant often require to make effort for data collection and sharing, which includes the consumption of limited resources on their devices. Some people might altruistically participate in such systems. However, it is not realistic to assume that all participants offer this effort altruistically. Therefore, adequate incentives must be given to people to participate. One common approach is to provide the participants with monetary incentives. Additionally, data need not be constantly collected at all places. In many applications, data collection is necessary only when there is some *utility* for the data. The difference between the value of the collected data to the application and the data collection cost is defined as the utility of the data. We propose a data acquisition framework in Chapter 3 for participatory sensing systems. This framework takes into account the major factors pertinent

to this context and efficiently shares sensor data among queries of different types with the objective of maximizing the total utility. Queries for sensor data can come from multiple different applications with arbitrary utility considerations.

Truthful Data Elicitation in Participatory Sensing. In participatory sensing systems, some participants might have incentives to report wrong data. For example, a participant might report higher costs for her data or wrong location tags for the data with the objective of receiving higher payments. Therefore, it is critical to prevent dishonest behavior of participants by appropriately designing the participatory system. We follow a game-theoretic approach towards addressing this problem in Chapter 4 by designing incentive compatible and individually rational mechanisms for collecting cost information and measurements from the participants. Moreover, we propose mechanisms for truthful data elicitation for privacy conscious participants by allowing them to make trade-offs between their privacy and monetary compensation.

Quality Assessment for Sensor Data. In order to determine the utility of the sensor data collected in participatory sensing systems, it is essential to assess the quality of the data. Several outlier detection techniques in sensor networks exist that classify the data as being normal or outlier. Some of these approaches can be adapted to assess the quality of data in sensor networks. In Chapter 5, we propose a novel online pattern-based quality assessment for sensor streams. We use itemset mining to find a frequent correlated pattern, consisting of the given sensor value (the *tested value*) and the sensor values on other streams that occur at the same time as the tested value (the *context*) that maximizes the logistic regression function in the sensor stream seen so far. The quality score is computed by combining the following features of the frequent pattern: 1) the relative frequency of the pattern, 2) the conditional probability of the tested value given the context, and 3) the relative size of the pattern with respect to the number of streams.

Keywords: *participatory sensing, privacy, one-shot query, continuous query, quality assessment, utility, mechanism design, optimization, frequent itemset mining, incentive compatible*

Résumé

Ces dernières années ont vu la prolifération de capteurs personnels tels que les GPS, appareils photos, station météorologiques amateurs, et plus important encore, les smartphones. La plupart des appareils sont mobiles par natures e.g. smartphones, GPS, ou peuvent être portés par les utilisateurs dans leurs activités quotidiennes. Grace à l'avancée technologique dont ont bénéficié les capteurs, il est désormais possible de les intégrer dans de petits appareils. Par example, nous pouvons considerer les smartphones comme outil d'enregistrement car ils sont équipés de plusieurs capteurs tels que GPS, accéléromètre, gyroscope, microphone et capteur de proximité.

Cette avancée technologique offre une opportunité sans précédent de créer un nouveau paradigme appelé participatory sensing, dans lequel les utilisateurs captent et partagent des données en rapport avec un phénomène particuliers lié à leur environnement. Cette opportunité est principalement due à (I) l'omnipresence des smartphones avec leurs capteurs embarqués, (II) la possibilité d'avoir des capteurs de petite taille, à bas coup et adaptable aux smartphone et (III) la facilité d'accès aux différents types de connexion tels que 3G, 4G ou WiFi. Néanmoins, afin d'utiliser pleinement le potentiel du participatory sensing, plusieurs défis doivent être relevés. Ces défis incluent en autre, la protection de la vie privée des participants, la vérification de la qualités des données collectées, l'efficacité énergétique des capteurs, les problèmes liés à la disponibilités des données lors des déplacements de l'utilisateur et la motivation des utilisateurs afin d'augmenter leur adhérence. Dans cette thèse, nous proposons plusieurs méthodes répondants à ces défis.

Acquisition efficace de données dans le contexte du participatory sensing. Dans le contexte du participatory sensing, les participants doivent souvent fournir des efforts particuliers afin de collecter et partager des données, comme par exemple l'utilisation d'une partie des resources de leurs appareils. Certaines personnes altruistes peuvent adhérer à ce système spontanément, cependant nous ne pouvons pas nous attendre à ce que tout les participants fournissent cet effort sans rien en échange. Afin de pallier ce problème, les utilisateurs doivent être incités à participer. Une approche courante est de proposer une contrepartie financière aux participants. De plus, les données ne doivent pas nécessairement être collectées constamment et en tout lieux. Dans de nombreuses applications, la collection de données n'est nécessaire

que lors que ces données sont utiles. La différence entre la valeurs des données collectées et le cout d'acquisition de ces données est définit comme étant l'utilité de ces données. Nous proposons, dans le chapitre 3 de cette thèse, un framework d'acquisition de données dans le cadre du participatory sensing. Ce framework prend en compte les facteurs les plus importants dans ce contexte et met en place un système efficace de partage des données collectées provenant de différentes requêtes, avec pour but de maximiser l'utilité totale. Les demandes pour ces données peuvent provenir de plusieurs applications avec différentes definition de l'utilité.

Obtention des données authentiques dans le participatory sensing.
Dans les systèmes de participatory sensing, les participants peuvent avoir intérêt à fournir de fausses données. Par exemple, un participant pourrait indiquer un prix plus élevé que nécessaire pour ses données, ou alors fournir de fausse données de géolocalisation avec pour objectif de recevoir plus d'argent. C'est pour cela qu'il est primordial d'empêcher les comportement malicieux des participants en concevant judicieusement le système participatif. En nous basant sur la théorie des jeux, nous définissons dans le chapitre 4 des mécanismes compatibles en motivation et individuellement rationnels afin de collecter des informations sur les couts des données et les données en elles memes. De plus, nous proposons des mécanismes permettant d'obtenir uniquement les données authentiques et ce, dans le respect de la vie privée des participants en leur permettant de définir un compromis entre leur vie privée et la compensation financière.

Evaluation de qualité des données collectées. Afin d'évaluer l'utilité des données collectées par un système de participatory sensing, il est essentiel d'estimer la qualité et la confiance que l'on peut accorder à ces données. Plusieurs méthodes de detection de données aberrantes pour les réseaux de capteurs existent déjà et permettent de classifier les données comme étant normales ou anormales. Plusieurs de ces approaches peuvent être adaptées afin d'estimer la qualité des données de réseaux de capteurs. Dans le chapitre 5, nous proposons une nouvelle méthode basée sur la reconnaissance de motifs, permettant d'estimer à la volée la qualité que l'on peut attribuer à des données. Nous recherchons des ensembles d'items (itemset mining) afin de corréler des motifs, en nous basant sur la valeur retournée par le capteur (la valeur testée) et les données provenant d'autres streams (le contexte), qui arrivent dans la même fenêtre de temps que la valeur testée, et qui maximisent la fonction de regression logistique dans les streams qui ont été évalués jusqu'ici. L'indice de qualité est calculé en combinant les caractéristiques du motif suivantes: 1) la fréquence relative du motif, 2) la probabilité de la

valeur testée conditionné par le contexte, 3) la taille relative du motif, par rapport au nombre de streams.

Mots clés : *participatory sensing, vie privée, requête en une fois, requête continue, évaluation de qualité, utilité, conception des mécanismes d'incitation, optimisation, recherche d'ensemble d'items, compatibilité en motivation*

Contents

List of Figures

List of Tables

List of Algorithms

Introduction

1.1 Background

In the recent years we have been witnessing a proliferation of privately owned sensing devices such as GPS devices, cameras, home weather stations and, more importantly, smart-phones. In addition, most of these devices are intrinsically mobile, e.g., smart-phones and GPS devices, or can be easily carried by people. The advancement in sensor technology has made it possible to have various sensors in small hand-held devices. For example, smart-phones nowadays are equipped with several sensors such as GPS, accelerometer, gyroscope, lux meter, microphone, and proximity sensors. Therefore, we can consider mobile phones as sensing devices.

This provides an unprecedented opportunity for a new application paradigm called *participatory sensing*, where ordinary people collect and share sensing data about some phenomenon of interest [18]. The unique enablers of participatory sensing applications are (I) the ubiquity of smart-phones with built-in sensors, (II) small, low-cost and pluggable sensors, and (III) the availability of several connectivity media such as 3G, 4G, and WiFi. Participatory sensing applications span areas such as public health and well-being [24, 34, 107], urban planning [35, 83], environmental monitoring [6, 32, 62], and transportation and traffic monitoring [14, 43, 52, 86, 92, 128].

The term *participatory sensing* has been used in the literature to refer to the sensing applications in which participant's intervention is necessary in the sensing and sharing process [18]. On the other hand, the term *opportunistic sensing* has been used for sensing applications where the participant does not intervene (at least not actively) in the sensing and sharing process [19]. However, the level of participation or involvement in the process can vary from one application to another or even from one participant to another. Therefore, we can use the general term *crowdsensing* to refer to the wide spectrum of user participation. In this case, participatory sensing and opportunistic sensing reside at the extremes of the spectrum. The term *mobile crowdsensing* is used

in [42] to refer to this broad range of community sensing applications. Other names that have been used in the literature to refer to the same concept of crowdsensing include *community sensing* [68], *urban sensing* [19], *people-centric sensing* [19, 20], and *citizen sensing* [18].

In this thesis, however, we use the term **participatory sensing** regardless of the level of user involvement in the sensing process. The reason is twofold: (I) the described phenomenon is widely known in the community by the name *participatory sensing*; and (II) by using the name *participatory sensing* we emphasize that *people* participate in collecting and sharing data using their mobile devices in contrast to deploying privately-owned stationary or mobile sensor networks. In fact, in participatory sensing even these stationary or mobile sensor networks can be regarded as *participants*. Throughout this thesis, the terms *participant* and *user* are used interchangeably. In cases where clear distinction is required between users of the system and the data collectors, we use the term *end user* to refer to the ultimate user of the system.

Participatory sensing enables applications and scenarios which could not be achieved by privately-owned sensor network deployments because they were economically not viable or technically not feasible. In particular, contrary to the sensor network deployments, the participatory sensing paradigm takes advantage of the following facts.

- Participatory sensing takes advantage of the mobility of participants. People have different mobility patterns: each person has his/her specific itinerary, which can also change according to the time of the day or week. Therefore, a large enough number of participants can collectively cover a wide area of the desired sensing region at any time of the day.

- There is no need to install and maintain sensing devices since the sensing devices are either built in mobile phones or come as low-cost, pluggable, and packed extensions that can easily be purchased and attached to the mobile phones.

- The connectivity media such as 3G or WiFi access points are widely available and used by people. Thus, there is no need to separately install and enable connectivity in the deployment process.

- Contextual information from participants can be inferred and used to enrich the knowledge gained from the sensed data and also to provide the participants with personalized feedback.

Motivated by the cost-effectiveness, scalability and wide coverage of sensing offered by participatory sensing, a vast range of applications is envisioned. For example, Haze Watch [21] provides the participants with sensing devices that can be mounted on their vehicles. These devices measure carbon monoxide, ozone, sulphur dioxide, and nitrogen dioxide. The measurements are transferred to the participant's smartphone through Bluetooth interface. The mobile phone then sends the data along with GPS coordinates and time to the server. A pollution map is constructed based on the collected data.

The participants exposure to pollutants can be measured and displayed on their mobile phones. Similarly, Safecast [3] is a participatory sensing project aiming at collecting radioactive radiations in large scale so as to create high resolution radiation maps. People who volunteer to collect data receive sensing devices that can be mounted on their cars. Radiation level measurements are taken by these devices and sent to a server. The radiation level data collected in this manner is claimed to have higher geographical resolution and better consistency than the data reported by the government. Chapter 2 provides a review of the existing participatory sensing initiatives.

1.2 Research Challenges in Participatory Sensing

In order to use the potential of participatory sensing to its full extent, there are several challenges that must be addressed. Researchers have already identified these challenges, e.g., in [61, 63]. Consequently, the research community has started providing solutions for some of these challenges. The key research challenges in participatory sensing are the following.

Preserving participant privacy. Humans are at the heart of participatory sensing. By participating, people might directly or indirectly reveal sensitive information about themselves and their surroundings. For example, it is often essential to tag the measurements with the location where they are taken. This could threaten the participants' privacy. Consequently, people who are concerned about their privacy might lose their incentive for participation. Mechanisms are required to ensure preserving user's privacy to a desired level. At the same time, mechanisms should be put in place to ensure a certain quality of data knowing that most of the privacy preserving mechanisms disturb data quality.

Assuring and assessing integrity of data. In participatory sensing, several factors can impact the quality of data. Because of mobility of participants, in some regions the number of data collectors might not be enough for the purpose of the applications. Privacy protecting mechanisms might restrict the areas and times of taking measurements, which could lead to data unavailability. In addition, quality of data can be reduced mainly because of three sources of data distortion: (I) privacy protection mechanisms which add noise to data, (II) malfunctioning sensing devices for example due to calibration problems, and (III) malicious users who intentionally modify data. Effective mechanisms are needed to assess the quality of data collected by the participants and to ensure a certain quality level for each application that uses the data.

Inferring user context. In many applications, it is necessary to infer the context and activity of users to enrich the data collected or to give personalized service to the users. For example, if a user wants to get automatic updates about current safe jogging tracks, in addition to her current location, the system needs to know that the user is

indeed jogging. Embedded sensors in mobile phones such as GPS, microphone, and accelerometer can be used to enable this task. While addressing the task of inferring context and user activities, energy efficiency and user privacy must be considered as they are tightly coupled with this task.

Energy-efficient data collection. Participatory sensing largely relies on participants' mobile devices (e.g., smartphones) for taking measurements, process and transmit them. However, these mobile devices often have a limited source of energy. Users would like to prolong the discharging time of the batteries on their mobile phones so as to use their devices for their main activities, such as making phone calls, etc., without recharging them frequently. Therefore, it is essential for participatory sensing systems to be as energy-efficient as possible.

Incentivizing participation. Distributed systems, in which participants interact freely without any centralized authority require robust incentives to ensure contribution. Since participatory sensing systems are also not owned by anyone in particular, they too require the provision of social and economic incentives for participants. These include incentives for following the protocol; abstaining from malicious activities; and contributing their resources. The design of such incentive schemes could be guided by various approaches, including mechanism design, heterodox economics, and other socially inspired mechanisms.

1.3 Contributions

In this thesis work, we made the following main contributions towards addressing some of the research challenges in participatory sensing, individually or jointly.

1.3.1 Utility-driven data acquisition in participatory sensing

Participating in a participatory sensing system requires some level of effort from the participants. This includes lending their limited resources to data collection and sharing. We cannot assume that all participants offer this effort altruistically. Therefore, some strong incentives should be given to people to participate. One common approach is to provide the participants with some monetary incentives. Moreover, data need not be collected and shared all the times at all places. In many applications, data is required only when there is some *utility* for it. Utility is defined as the difference between the value of the collected data to the application and the data collection cost. For example, when there is at least one query asking for measurements about the state of a phenomenon at a specific place or region and the application can provide the cost of collection of certain measurements, the system can ask some participants to provide the required data. In this thesis, we propose a utility-driven framework for efficient data collection in participatory sensing. In particular we make the following contributions:

- We propose a data acquisition framework in the context of participatory sensing that takes into account the factors pertinent to this context and efficiently shares sensor data among queries of different types. Queries for sensor data come from multiple different applications or users that can have any arbitrary utility considerations.

- We formulate the optimal data acquisition problem as a multi-query optimization with the objective of maximizing the total utility (or *social welfare*) and propose efficient heuristic solutions for various query types and query mixes.

- Important query categories, including one-shot and continuous queries, in the context of participatory sensing are considered and efficient data acquisition algorithms are proposed for each query type as well as the combination of different query types.

1.3.2   Truthful data elicitation in participatory sensing

In participatory sensing systems, participants are not always truthful and sometimes have incentives to report falsified data. For example, one participant might think that by reporting a higher cost for her data or by tagging the data with a wrong location, she can receive a higher payment. As an another example, in an air pollution data collection scenario, the participant might be involved in generating the pollutants. In this case, she has strong incentives to report wrong measurements. Therefore, it is essential to devise mechanisms to detect and prevent untruthful behavior of participants. In this thesis, we propose a game-theoretic approach towards addressing this problem by designing incentive compatible and individually rational mechanisms for collecting cost and measurements from the participants. Specifically, we make the following contributions.

- We formulate the problem of optimal data acquisition for multiple point queries and we propose incentive compatible mechanisms for truthful cost and data elicitation in participatory sensory context.

- We also propose mechanisms for truthful data elicitation when participants are privacy conscious by allowing the agents to make trade-offs between their privacy and monetary compensation. Our mechanisms perform this trade-off in order to maximize the utility of the center.

1.3.3   Quality assessment of sensor data streams

Assessing quality of data collected in participatory sensing systems is an essential task for determining the utility of the data. There exists a large body of work in outlier detection in sensor networks. Some of the proposed approaches can be adapted to assess the quality of data in sensor networks. In this thesis, we propose a novel online pattern-based quality assessment for sensor data streams. Even though this approach was proposed for data streams which come from stationary sensors, we believe that similar principles can be

used for trust assessment in participatory sensing, where sensors are often mobile. In summary, we make the following contributions.

- We use itemset mining to find a frequent correlated pattern, consisting of the *tested value* and a *context* that maximizes the logistic regression in the sensor data stream seen so far. Tested value is the stream value for which the trust is computed and context refers to the sensor values on other streams having the same timestamps as the tested value.

- We compute a quality score for a tested value using the following features of the pattern: 1) the relative frequency of the pattern, 2) the conditional probability of the tested value given the context, and 3) the relative size of the pattern with respect to the number of streams.

1.4 Thesis Organization

We start the remainder of the thesis by surveying the state of the art relevant to this thesis in Chapter 2. In Chapter 3 we introduce our framework for utility-based data collection in participatory sensing and present our proposed algorithms for efficient data collection given different types of queries. Chapter 4 is devoted to our mechanism design approach for truthful data elicitation in participatory sensing. We present our incentive compatible and individually rational mechanisms and through simulations demonstrates different properties of these mechanisms. Our frequent pattern-based technique for assessing quality of sensor data is presented in Chapter 5. Lastly, we conclude the thesis in Chapter 6 and lay out detailed future work.

1.5 Selected Publications

Among the published research papers in the course of this thesis work, the following publications are the main constituents of this thesis:

- M. Riahi, R. Rahman and K. Aberer. Privacy, Trust, and Incentives in Participatory Sensing. In Participatory sensing, opinions and collective awareness *(Under publication)*. (Chapter 2)

- M. Riahi, T. G. Papaioannou, I. Trummer and K. Aberer. Utility-driven Data Acquisition in Participatory Sensing. 16th International Conference on Extending Database Technology (EDBT), Genoa, Italy, 2013. (Chapter 3)

- M. Riahi, R. Rahman, K. Aberer and K. Larson. Truthful Data Acquisition in Participatory Sensing. *Submitted to ACM MobiHoc 2015.* (Chapter 4)

- R. Gwadera, M. Riahi and K. Aberer. Pattern-wise trust assessment of sensor data. IEEE MDM 2014 - 15th IEEE International Conference on Mobile Data Management, Brisbane, Australia, 2014. (Chapter 5)

Chapter 2

Background

2.1 Introduction

In this chapter we review the literature related to this thesis. In particular, we review the research work regarding the concept of participatory sensing and its challenges, as well as participatory sensing campaigns and projects in this area. We also review the state of the art in data collection and query processing mechanisms in participatory sensing and sensor networks. Since one of the contributions of this thesis is the proposal of truthful mechanisms for eliciting data in participatory sensing, we provide a review of the related work in truthful data elicitation. We also survey the state of the art techniques in quality assessment of sensor data.

The rest of this chapter is organized as follows. In Section 2.2 we provide a detailed review of research work related to the concept of participatory sensing and its pertinent research challenges. In addition, we provide a list of participatory sensing projects and review their characteristics. In Section 2.3, the related work regarding efficient data collection mechanisms in participatory sensing systems and traditional wireless sensor networks is reviewed. Section 2.4 briefly reviews the relevant work in the area of query processing and query optimization in sensor networks. This section together with Section 2.3 serves as the background for our contribution in Chapter 3. Section 2.5 is devoted to reviewing the state of the art techniques in truthful data elicitation from participants, not necessarily in the context of participatory sensing. This section aims to provide the necessary background for our contribution in Chapter 4. Lastly, in Section 2.6, we review the related work regarding quality assessment of sensor data streams. This section provides the required background for our contribution in Chapter 5 regarding quality assessment of sensor data.

2.2 Participatory Sensing Applications

2.2.1 Background

The concept of *participatory sensing* was first introduced in [18]. It states that participatory sensing takes advantage of everyday mobile devices to create interactive, participatory sensor networks to empower ordinary people as well as professionals to collect, analyze, and share information about a local phenomenon. By placing the users at the center of the sensing process and increasing the quantity, quality, and credibility of collected data, participatory sensing is deemed to improve existing data collection and analysis efforts, such as small-scale research-oriented data collection campaigns or autonomous stationary and wireless sensor networks.

With the idea of moving from traditional small-scale, single-purpose, and application-specific wireless sensor networks to large-scale and general-purpose sensor networks that can directly benefit the general public, [19] introduced the concept of *people-centric sensing*. In people-centric sensing, humans are at the center of the sensing activities; people and their surroundings are being sensed by people.

MetroSense is an *opportunistic sensor network* architecture proposed in [19] to enable large-scale people-sensing applications. The sensor network is called opportunistic because it takes advantage of the sensing and communication opportunity provided by mobile sensors carried by people. MetroSense enables interactions between mobile sensors, stationary sensors, and edge wireless access points to achieve opportunistic tasking, sensing, and data collection.

2.2.2 Applications

In recent years a new class of applications has emerged which is based on the participatory sensing paradigm. These applications not only use the environmental data collected by the users, but also can infer and utilize the context and activities of the users. These applications are categorized into *people-centric* and *environment-centric* applications by [23]. People-centric applications use the sensors embedded into mobile phones to collect and analyze user activities. Environment-centric applications take advantage of external sensors or sensors integrated into mobile phones to measure some parameters of the environment. However, in many cases applications in these two groups overlap. For example, the user context and activity can be inferred and combined with environmental measurements to provide personalized recommendations for the users. Based on the phenomenon being sensed, [42] classifies the participatory sensing applications into three classes: *environmental* applications which measure a natural environmental factor such as air pollution; *infrastructure* applications which collect data about a public infrastructure such as traffic congestion and road conditions; and *social* applications which enable the users to share information and achieve a social benefit. In the following we present a finer-grained classification of participatory sensing applications and for each class we review some representative examples. A more comprehensive overview of the existing participatory sensing applications can be found in [65] and [23].

Air quality monitoring. Haze Watch [21] provides the participants with sensing devices that can be mounted on their vehicles. These devices measure carbon monoxide (CO), ozone (O_3), sulphur dioxide (SO_2), and nitrogen dioxide (NO_2). The measurements are transferred to the participant's smartphone through Bluetooth interface. The mobile phone tags the data with time and location and sends it to the server. The data collected is used to construct a pollution map. The exposure of the participants to pollutants can be measured and displayed on their mobile phones.

PollutionSpy [62] is a prototype application for measuring air pollution in traffic using mobile phones and generating pollution maps. The pollution map can be viewed on the mobile phone using the locally collected data. The mobile phone can also send the data to a remote server. A Web interface is provided for viewing the pollution maps created based on the reported measurements by the participants. External sensors such as CO, CO_2, NO, NO_2, SO_2, temperature, and windspeed are connected to the mobile phone using a Bluetooth interface.

Other examples of participatory sensing systems for air quality monitoring are Common Sense [32, 100] and OpenSense [6].

Noise and ambiance monitoring. NoiseTube [83] converts the mobile phones into noise monitoring devices by providing an application that runs on the mobile phones and continuously measures the loudness of the environmental sound captured by the phone's microphone. The application displays real-time pollution maps on the phone from the noise measurements combined with GPS location information. Individually collected data can be transferred to a server in order to create collective noise pollution maps. Participants can semantically tag the noise levels to specify the pollution sources or location information in places where GPS cannot be used.

MetroTrack [8] is a mobile-event tracking system that uses mobile phones carried by people. In particular MetroTrack uses the microphones on the mobile phones to detect the noise source and estimate its distance to the mobile phone. The future location of the noise source is predicted based on a distributed Kalman-Consensus filtering algorithm. Mobile phones collaborate with each other to track the source noise by forwarding the tracking tasks to the phones that are closer to the predicted future location of the target.

Further examples of noise and ambiance monitoring applications are NoiseSpy [62], SoundSense [78], EarPhone [106], MoVi [12] and community maps for London Thames gateway [35].

Other environmental hazard monitoring. Safecast [3] is a participatory sensing project aiming at collecting radiation measurements in large scale so as to create high resolution radiation maps. People who volunteer to collect data receive sensing devices that can be mounted on their cars. Radiation level measurements are taken by these devices and sent to a server. Primarily deployed in Japan, the radiation level data collected is claimed to have higher geographical resolution and better consistency than the data reported by the government.

Community-based sensors have also been used for detecting earthquakes [39]. In the prototype application, built-in accelerometer sensors in the mobile phones, stand-alone sensors, and accelerometer sensors that are connected through USB to host computers are used for rapidly detecting earthquakes. These heterogeneous sensors are managed by a cloud computing platform that runs data fusion algorithms and issues real-time early-warning of seismic hazards.

Traffic monitoring. A cooperative public transport tracking participatory system is proposed in [127] that uses mobile phones to track the public transport vehicles with the ultimate goal of improving the passengers experience. When a user is riding in a public transport vehicle, the mobile phone periodically sends location information to a central tracking server. An automatic approach using accelerometer data is used to determine whether a user is riding in a vehicle. A spatio-temporal trajectory matching mechanism is used to determine if a user is riding a public transport vehicle, and if so which one. For tracking underground public transport a different approach is proposed because in the underground environment GPS-based or WiFi-based localization cannot be employed.

CarTel system [52] deploys dedicated sensing and computing devices equipped with GPS sensors on cars to opportunistically obtain information about traffic delays observed as cars move and to use that information in traffic monitoring and route planning applications. CarTel relies on intermittent connectivity through WiFi or Bluetooth to the centralized server by creating a delay-tolerant network stack. By analyzing the time it takes a participant to commute to work, CarTel can determine traffic congestion, and visualize jammed roads on a map.

Other examples of participatory traffic monitoring applications include Nericell [92], Mobile Millennium [14], VTrack [128], and GreenGPS [43].

Public infrastructure monitoring. The Pothole Patrol [37] deploys dedicated sensing devices on cars for detecting and reporting road surface conditions. Potholes and other rough road surface anomalies are detected using accelerometer data combined with GPS localization data. The large-scale and continuous road surface condition monitoring is made possible thanks to the inherent mobility of the participants and opportunistic data collection.

ParkNet [86] is a participatory sensing system for monitoring road-side parking space occupancy. The sensing platform consists of a low-cost ultrasonic sensor that reports the distance to the nearest obstacle and a GPS receiver that specifies the corresponding location. The sensor devices are deployed in vehicles that opportunistically collect and report parking space availability to a server. The server provides a real-time parking occupancy of the city.

Further sample participatory sensing applications for monitoring civil infrastructures include participatory risk management [56], participatory waste management [97], and road bump monitor [25].

Personalized health monitoring. Jog Falls [95] is a system for monitoring patients energy expenditure and calorie intake with the goal of providing them with continuous awareness of their diet and activities. Energy expenditure, i.e., calories burned, is automatically calculated by combining heart rate and accelerometer data captured by wearable sensors. Jog Falls provides an interface for the patients to enter their calorie intake, and monitor their trends and goals. It also provides a backend interface for the physician to monitor the progress and compliance of the patients and give them necessary advice/coaching to better manage chronic disease conditions.

MobAsthma [62] is an asthma monitoring system that can provide the patients with real-time exposure to pollution. It can also monitor the asthma condition of the patients and alert medical staff in case of detecting asthma attacks. This application enables asthma specialists and allergists to study the relationship between the asthma and respiratory problems and the personal exposure to air pollution. The required data can be provided by medical devices such as asthma peak-flow combined with other air pollution sensors and GPS coordinates connected to the mobile phone through the Bluetooth interface.

Additional examples of participatory sensing applications for monitoring personal health conditions include UbiFit Garden [24], HyperFit [54], DietSense [107], HealthSense [125], HealthAware [44], BALANCE [28], SPA [114] and the work in [9], which is applicable for pediatric obesity applications.

Measuring exposure to environmental factors. In PEIR (Personal Environmental Impact Report,) air quality parameters are not measured by the participants. However, PEIR is a participatory sensing system which enables users to use their mobile phones to find out their exposure to CO_2 and PM 2.5 particulates [94]. The participant's mobile phone continuously sends GPS and cell tower location traces to a server. The server in turn determines the participant's transportation mode and her trajectory. This information, combined with the input from weather station reports, traffic conditions, and vehicle emission models, enables PEIR to calculate participant's exposure to CO_2 and particles.

ExposureSense [103] is a mobile participatory sensing infrastructure that combines activity recognition on the mobile phones with external air quality data from OpenSense [6] to determine participant's exposure to air pollutants. In addition, external pluggable sensors such as O_3 can directly provide air quality data to the application on the mobile phone.

Monitoring and recording sport activities. Biketastic [117] is a participatory sensing projects in which bikers use their mobile phones that are quipped with GPS localization and accelerometer sensors to collect and share data about their biking routes and the roughness of the roads they are commuting on. Sound samples from the phone's microphone can be used to document the noise level of the routes. In addition to personal use of the collected data, participants can share their gathered data among

each other. This information can be combined with existing data such as air quality, traffic conditions and traffic accident to enable the bikers to select their desirable biking routes. For example, one can choose a route with minimum exposure to noise or air pollutants, or a route with minimal probability of accident.

BikeNet [34] is another participatory sensing application that collects and shares data about biking performance and road and environmental conditions using several peripheral sensing devices. Sensing devices and the biker's mobile phone create a Bike Area Network. The BikeNet system can operate in delay tolerant sensing or real-time sensing modes depending on whether the mobile phone can transfer the data to the back end in real-time. When two bicycles meet each other, they can exchange data to facilitate data transfer to the back end. Cyclist performance and fitness data collected and stored by the system include: current speed, average speed, distance traveled, calories burned, path incline, heart rate, and galvanic skin response (as an indicator of emotional excitement or stress level). The cyclists are provided with real-time information about the healthiness of their routes in terms of air quality, allergen levels, noise levels, and roughness of the road. The environmental data can be shared with the larger community. Environmental data fused with cyclists performance measurements can provide a holistic picture of the cycling experience.

SkiScape [33] and UbiFit Garden [24] are other examples of participatory sensing applications for documenting sport activities.

Enhancing social media and public awareness. NutriSmart [142] is a participatory sensing application with the goal of eliminating "food deserts", the communities lacking enough healthy and fresh food choices. The users log their food purchase experience including the type of food they are consuming and the places where they are purchasing the food. By aggregating data from the participants, the system can detect problematic areas, i.e., areas with a lack of grocery stores or access to fresh food and areas with a high concentration of fast food restaurants. With this participatory collection of evidence, people and authorities can react to eliminate the detected "food deserts".

The objective of MicroBlog [45] is to enable a high resolution view of the world by building a "virtual information telescope" consisting of billions of mobile phones acting as its "virtual lenses". Participants record multimedia blogs, including pictures, video, audio, etc., using their mobile phones. Enriched with other physical sensor readings and geotagged, the blogs are uploaded to a remote server to create a global map that can be consulted by the public. Users browse the maps and look for their desired information. If the information is not found, the system can submit a query that is answered either by the participants or by the sensor readings automatically collected.

CenceMe [90, 91] and MoVi [12] are further examples of participatory sensing applications which aim at enhancing user experience with social networking.

Price monitoring. MobiShop [111] is a participatory sensing application for collecting, processing, and delivering product price information from street-side shops to potential buyers. Using her mobile phone, a participant takes pictures of the shop receipt, which lists the products bought by the user and their prices. This information is extracted by an Optical Character Recognition (OCR) application and transmitted, along with the GPS location of the user and the time of the purchase, to a central server. The server maintains the updated prices of products for each shop. Users can query the server for the price of particular products in the shops in their neighborhood. Each query contains the GPS location of the user. The server returns a list of shops in the vicinity of the user containing the requested product and their prices. LiveCompare [27] is another application for participatory product price collection.

PetrolWatch [31] takes advantage of mobile phones of the participants equipped with cameras to take pictures of fuel prices on the road-side price boards of gas stations. Tagged with time and location information, these pictures are transferred to a central server. The server runs computer vision algorithms to extract price information from the pictures. It is assumed that mobile phones are placed in positions where the cameras are facing the road-side so that the cameras can automatically take pictures of the price boards. With the help of a GIS application and GPS location of the vehicle, the camera can be automatically commanded to take pictures when the vehicle approaches a gas station.

2.2.3 Discussion

In Chapter 1 we identified the key research challenges in participatory sensing applications as *preserving participant privacy*, *data integrity assurance and assessment*, *inferring user context*, *resource-efficient data collection*, and *incentivizing participants*. Each of the applications outlined in this section addresses one or more of these challenges and ignores the rest. In this thesis we propose methods for addressing some of these key research challenges that can complement the existing approaches taken in the current participatory sensing applications. In particular, we assume that monetary compensation can be provided to participants to encourage their long-lasting participation. In order for this incentive mechanism to be efficient, given the existing requests for data, we propose a framework for data collection in participatory sensing by selecting the best set of participants for collecting the required data at each time period. In this regard, in Section 2.3 and Section 2.4, we review the related work in the areas of sensor selection and query processing in sensor networks and participatory sensing as our approach is built on the ideas from these research fields.

In this thesis, we also present mechanisms that incentivize the participants to report true information about their costs and measurements. Our contribution in this regards is based on the concepts and ideas from the research area of truthful data elicitation in information systems. We provide a review of the state of the art in this area in Section 2.5. Finally, towards addressing the important challenge of quality assessment of the collected data, we propose a novel method for quality assessment of sensor readings,

with primary target of stationary sensor networks. The related work regarding this contribution is presented in Section 2.6.

2.3 Sensor Selection

There is a large body of work in the field of sensor selection, placement, and scheduling in wireless sensor networks. In this section, we review some representative works in the area of sensor selection. Before reviewing the state of the art in sensor selection, we formally define the problem of sensor selection. We also provide an introduction to the concept of submodularity as it is widely used in sensor selection schemes.

2.3.1 Sensor Selection Problem

Efficient data collection for query processing in sensor networks, where the sensors are energy-constrained and resource-limited, boils down to selecting appropriate sensors so as to achieve a specific objective such as maximizing the coverage of the field and/or maximizing the lifetime of the network. Sensor selection schemes aim to select a subset of sensors such that the *total utility* is maximized while the total cost does not exceed the budget or the total cost is minimized. Maximizing utility and minimizing cost are two conflicting goals. Therefore, the goal of sensor selection schemes is to find the best utility-cost tradeoff. Formally, given a set $S = \{s_1, s_2, \ldots, s_n\}$ of sensors, a sensor selection scheme tries to find the best subset $S' \subseteq S$ in order to achieve its specific objective. In this thesis we define utility as *the difference between the value of the selected sensors to the application and the cost of selecting those sensors.* Let $v(S')$ denote a function that gives the value of the sensors in subset S' for the application and $c(S')$ a function that gives the cost of selecting sensors in S', then the utility of sensors in S' is given by

$$u(S') = v(S') - c(S'). \tag{2.1}$$

A sensor selection scheme with the goal of maximizing utility, finds the subset $S^* \subseteq S$ such that

$$S^* = \arg\max_{S' \subseteq S} u(S'). \tag{2.2}$$

The sensor selection problem with arbitrary $v(.)$ and $c(.)$ functions is a hard problem. For example, in a simple setting where $v(S') = \sum_{s \in S'} v(s)$, $c(S') = \sum_{s \in S'} c(s)$, and $c(S') \leq B$, where B is a constant, the sensor selection problem is NP-hard because it can be converted to the KNAPSACK Problem [53].

2.3.2 Submodularity

A *set function* is a function $f : 2^S \to \mathbb{R}$ that maps each subset $A \subseteq S$ to a value $f(A)$, where S is a finite set and 2^S denotes power set. Submodularity is a property of set functions that is defined as follows.

(a) Adding s_3 to set $\{s_1, s_2, s_4\}$ (b) Adding s_3 to set $\{s_1, s_4\}$
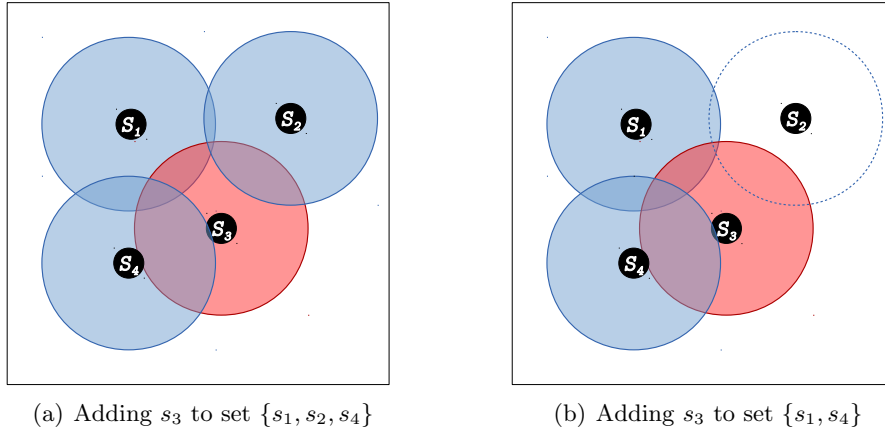
Figure 2.1: Graphical illustration of the diminishing return property of the sensor coverage function $\mathcal{G}$. When s_3 is added to the set $\{s_1, s_4\}$, the increase in coverage is more than when it is added to the set $\{s_1, s_2, s_4\}$; $\mathcal{G}(\{s_1, s_2, s_3, s_4\}) - \mathcal{G}(\{s_1, s_2, s_4\}) \leq \mathcal{G}(\{s_1, s_3, s_4\}) - \mathcal{G}(\{s_1, s_4\})$.

Definition 2.1. *(Submodularity)* A function $f : 2^S \to \mathbb{R}$ is *submodular* if for every $A, B \subseteq S$ it holds that

$$f(A \cup B) + f(A \cap B) \leq f(A) + f(B). \tag{2.3}$$

Submodular functions have a natural *diminishing return* property, which states that the marginal increase in the function decreases as the size of the input set increases. The following definition of submodularity, which is equivalent to the above definition, better demonstrates this property.

Definition 2.2. *(Submodularity)* A function $f : 2^S \to \mathbb{R}$ is *submodular* if for every $A \subseteq B \subseteq S$ and $x \in S \backslash B$ it holds that

$$f(B \cup \{x\}) - f(B) \leq f(A \cup \{x\}) - f(A). \tag{2.4}$$

A function which does not decrease as the size of the input set increases is called a *monotone* function. Monotone submodular functions constitute an important subclass of submodular functions.

Definition 2.3. *(Monotonicity)* A function $f : 2^S \to \mathbb{R}$ is *monotone* if for every $A \subseteq B \subseteq S$, $f(A) \leq f(B)$.

Example 2.1. Assume that each sensor s is capable of measuring a phenomenon in a disk with radius r_s centered at s. This area is called *sensing range* or *coverage* of s. The sensors are spread in a rectangular area R. Let $\mathcal{G}(A)$ be a function that gives the aggregate coverage of the sensors in set A. $\mathcal{G}$ is a monotone submodular function. Figure 2.1 graphically illustrates the diminishing return property of the submodular function $\mathcal{G}$.

2.3.3 Overview of Existing Work in Sensor Selection

A survey study of sensor selection schemes in wireless sensor networks is provided in [109], which based on the purpose of selection, classifies sensor selection schemes into four categories: (1) *coverage schemes* in which sensors are selected so that the complete coverage of the field is achieved; (2) *target tracking and localization schemes* where sensors are selected to track or localize target objects; (3) *single mission assignment schemes* in which sensors are selected in a way that a specific mission, that is repeatedly performed over time, is accomplished most efficiently; and (4) *multiple mission assignment schemes* in which multiple missions are accomplished in the most efficient way by selecting the appropriate set of sensors. In the following we present an overview of sensor selection approaches in three categories: *centralized sensor selection*, *distributed sensor selection*, and *sensor selection in participatory sensing*.

2.3.3.1 Centralized Sensor Selection

A utility-based sensor selection framework is proposed in [16] in which applications can specify the utility of each set of sensors in a wireless sensor network. The goal is to select a sequence of sets to maximize the total utility while not exceeding the available energy. Two utility function classes, namely submodular and supermodular, are studied and the algorithmic performance characteristics are identified for each class. Supermodular functions have the opposite property of submodular functions. In addition, geometric penalty functions in which the utility of measurements are inversely related to distance of the measurements from targets are studied. A *tiered* sensor network architecture is considered in this work, which consists of a large number of small battery-powered sensors that send data to smaller number of more powerful nodes in the upper tier.

A heuristic approach based on convex optimization is proposed in [58] for the sensor selection problem with the objective of minimizing the estimation error. In this paper, the sensor selection problem is defined as selecting k out of m measurements such that the vector $x \in R^n$ is best estimated. The ith measurement is given by $y_i = a_i^T x + n_i$, where a_i denotes the ith measurement vector and n_i is a Gaussian noise with distribution $\mathcal{N}(0, \sigma^2)$. The performance appears to be near optimal in numerical experiments, but there is no guarantee on the gap between the optimal solution and the approximate solution. In [115], the problem of sensor selection, where a set of sensors is selected according to the maximum a posteriori or the maximum likelihood rules, is formulated as optimizations of submodular functions over uniform matroids. It is shown that a greedy algorithm in this case performs within $(1 - \frac{1}{e})$ of the optimal solution. The sensor selection problem in this work is identical to that of [58].

Submodularity is also used in [51] to approach the problem of scheduling sensors for model-based reconstruction of a spatio-temporal continuous physical phenomenon in a distributed manner. Sensor scheduling is the process of activating sensors at different times so as to best estimate the state of a physical phenomenon and maximize the lifetime of the sensor network. Individual sensors capture the state of the phenomenon at discrete

points. In order to quantify the phenomenon in a continuous manner using the discrete samples taken from the sensors, background information of the phenomenon in the from of physical models can be used to fully reconstruct the state of the phenomenon. This approach is called *model-based reconstruction*. Assuming that the phenomenon is given as a linear stochastic system, a covariance reduction reward function, which is submodular, is used to rate each sensor schedule. A hierarchically structured communication scheme is used to enable distributed sensor scheduling with the assumption that each sensor knows its own measurement parameters and can calculate its own potential covariance reduction.

Simultaneous placement and scheduling of sensors is considered in [70], where an algorithm is proposed to efficiently and simultaneously decide where to place sensors and when to selectively activate them. It uses the submodularity of the utility function in order to guarantee a constant factor approximation to the optimal solution. The algorithm is also shown capable of trading off power consumption and accuracy.

2.3.3.2 Distributed Sensor Selection

A distributed active sensor selection (DASS) scheme is introduced in [116] for selecting a set of sensors to activate in order to completely cover a sensing field while the lifetime of the network is maximized. It uses the properties of Voronoi diagram to activate as few sensors as possible. The main assumption of this work is that the field is fully covered by sensors and the objective is to avoid selecting redundant sensors. DASS is an initiator-driven sensor selection scheme which is distributed in the sense that there is no central control over which sensors are selected. Each sensor only needs location and remaining energy information of the sensors in its one-hop distance. If the Voronoi cell of each sensor is covered by its sensing range, then the plane is guaranteed to be at least 1-cover and vice versa. A sensing field is said to be k-cover if any point in the field is covered by at least k sensors [50].

A distributed sensor selection for dense sensor networks is proposed in [79], which provides k-coverage of the sensing field. Each sensor is initially inactive in sensing but periodically checks whether it is necessary to activate its sensing unit based on its *contribution* or *coverage merit* so that the filed is k-covered. Each sensor waits for a back-off period before deciding to activate its sensing unit. The back-off period is shorter for sensors with larger contribution. Therefore, sensors turn on their sensing unit in decreasing order of their contribution, which results in fewer number if active sensors for providing k-coverage. The contribution of a sensor is calculated based on its probability of detecting events and the number of additional sensors needed to fulfill the required coverage.

When the utility function is not known a priori, it must be gradually learned from the data. The utility function can even sometimes change over the course of time. A distributed online sensor selection scheme is proposed in [46] to address this issue. When the unknown utility function is submodular, strong theoretical bounds can be proven for the algorithm proposed for this purpose. The algorithms are analyzed for two

network communication models: *broadcast model*, where sensors can broadcast messages to others, and *star model*, where sensors only communicate with the base station. At each time step t a set of sensors S_t are selected that send their data to the base station. The base station calculates the utility of the measurements $f_t(S_t)$. The goal is to maximize the utility obtained by the base station over T rounds, $\sum_{t=1}^{T} f_t(S_t)$.

2.3.3.3 Sensor Selection in Participatory Sensing

One of the main works in utility-based sensor selection in participatory sensing is [69]. The physical phenomenon is modeled as a *stochastic process* using the background information about the phenomenon. The utilitarian approach states that sensor readings that are more demanded by applications should be preferred over readings from other sensors. The importance of sensor readings is modeled by a *demand model*. A formal approach is also pursued to take into account the uncertainty about the location and availability of sensors. Lastly, the user preferences regarding privacy and resource consumption are included into the overall problem formulation.

In [69], a spatiotemporal phenomenon is modeled by a stochastic process, with a random variable $\mathcal{X}_s$ for each location $s \in \mathcal{V}$ (e.g., $\mathcal{X}_s$ can represent average car speed over road segment s or the radiation at location s). After observing values at some locations $\mathcal{X}_{\mathcal{A}} = X_{\mathcal{A}}$, we can predict the phenomenon values at the unobserved locations $\mathcal{V} \backslash \mathcal{A}$ by means of conditional expectations $\mathbb{E}[\mathcal{V} \backslash \mathcal{A} | \mathcal{X}_{\mathcal{A}} = X_{\mathcal{A}}]$. Since the predictions are not certain, the model is used to predict the variance at each location $s \in \mathcal{V} \backslash \mathcal{A}$ and the *reduction* in the predicted variance is used to quantify the value of the sensor locations after observing $\mathcal{X}_{\mathcal{A}} = X_{\mathcal{A}}$. The reduction in variance is given by the following:

$$Var(\mathcal{X}_s) - Var(\mathcal{X}_s | \mathcal{X}_{\mathcal{A}} = X_{\mathcal{A}}). \tag{2.5}$$

Having taken a utilitarian approach, the aim is to achieve the highest reduction in variances at locations s which are most demanded. To this end, a spatial process $\mathcal{D}_s$, called the *demand process*, is defined over all locations $s \in \mathcal{V}$ and then the expected *demand-weighted* variance reduction is considered:

$$R(\mathcal{A}) = \sum_{s \in \mathcal{V}} \mathbb{E}[\mathcal{D}_s(Var(\mathcal{X}_s) - Var(\mathcal{X}_s | \mathcal{X}_{\mathcal{A}} = X_{\mathcal{A}}))]. \tag{2.6}$$

It might not be possible to sample at locations $\mathcal{A}$ directly as there is uncertainty in the current sensor locations. In addition, for privacy protection of users, we need to incorporate some *noise* in *selection*. Thus, we assume that we can choose among a set $\mathcal{W}$ of *observations*, e.g., cars, sensor owners, etc. Each observation $w \in \mathcal{W}$ corresponds to a *distribution* over possible sensor locations, and any subset $\mathcal{B} \subseteq \mathcal{W}$ leads to a distribution $P(\mathcal{A}|\mathcal{B})$ over subsets $\mathcal{A}$. The final informational objective to maximize is then the following

$$F(\mathcal{B}) = \mathbb{E}_{\mathcal{A}|\mathcal{B}}[R(\mathcal{A})] = \sum_{\mathcal{A}} P(\mathcal{A}|\mathcal{B})R(\mathcal{A}). \tag{2.7}$$

In order to couple the utility of information with the sensor owner constraints on sharing preferences and resource usage, a cost function C is defined to associate each set $\mathcal{B}$ of observations with a non-negative cost $C(\mathcal{B})$. Based on the introduced model, given the budget L that can be spent on observations, the goal is to select a set of observations $\mathcal{B}^*$ such that

$$\mathcal{B}^* = \arg\max_{\mathcal{B}} F(\mathcal{B}) \text{ subject to } C(\mathcal{B}) \leq L. \tag{2.8}$$

This problem requires solving an NP-hard discrete optimization problem for which finding the exact solution is typically intractable. However, when $F(\mathcal{B})$ is submodular, a greedy algorithm exists that performs within $(1 - \frac{1}{e})$ of the optimal solution.

2.4 Query Processing and Optimization

Data collection for query processing in participatory sensing differs from query processing in traditional database management systems (DBMS). Contrary to query processing in databases where data is always available, in participatory sensing the data at required locations might not always be available because of the uncontrolled mobility of the participants. Moreover, the cost of obtaining data is different in participatory sensing compared to traditional databases. The types of queries that are formulated by end-users in participatory sensing is also different than the types of queries in DBMSs. Regarding data collection and query processing, participatory sensing is closer to traditional wireless sensor networks than to DBMSs.

2.4.1 Model-Based Data Acquisition

Statistical models of physical phenomena can be incorporated in data collection and query processing in sensor networks to provide more robust interpretation of sensor readings and help optimize sensor data acquisition [29]. Models can be used to identify outliers or to estimate values of the phenomenon in locations where no sensor reading is available. Models are also used to determine whether queries can be answered solely by the model or new sensor readings are required to be collected. In BBQ [29], a model is denoted as a probability density function (pdf), $p(X_1, X_2, \ldots, X_n)$, that assigns a probability to any possible assignment of attributes $X_1, \ldots, X_n$. Each X_i represents an attribute of a particular sensor. In particular, BBQ employs a time-varying multivariate Gaussian as the model. This model is initially constructed from historical data.

BBQ supports probabilistic queries, which are normal SQL queries that include error tolerances and confidence levels that specify the tolerable uncertainty in the answer. Queries are translated into probabilistic computations over models. If a query cannot be answered by the model in the specified confidence threshold, more data will be acquired from the sensors to update the model and to provide the required confidence. Based on the query, the model, and the network topology, BBQ generates an *observation plan* and sends the plan to the network to acquire the necessary readings. The cost model for generating the optimal observation plan takes into account the energy cost, which

consists of communication and acquisition costs. In addition, model-based querying can incorporate correlations between readings from different sensors as well as correlations between different sensor attributes, e.g., temperature and voltage.

2.4.2 Multi-query Optimization in Sensor Networks

The problem of multi-query processing has been systematically defined in [112] in the context of relational database systems. The basic idea of multi-query processing is that since some queries might have some data in common, instead of processing each query individually, we process the queries together. For example, consider the relation *Emp(name, salary, experience)* and the following concurrent queries:

```
Q₁: SELECT * FROM Emp WHERE salary > 20000
Q₂: SELECT * FROM Emp WHERE salary > 30000 AND experience < 2
```

Then, it might be more efficient to process Q_1 and Q_2 together because the result of Q_2 is a subset of the result of Q_1.

The idea of multiple query optimization and processing has also been used in query processing in wireless sensor networks. For example, a two-tier multiple query optimization scheme is proposed in [136] with the purpose of sharing data acquisition, computation, and communication cost of multiple concurrent queries in a resource limited wireless sensor network. The first optimization phase is performed at the base station. This optimization involves cost-based query rewriting, to convert user queries to synthetic queries such that duplicate data requests are minimized and the query results are correct. Radio transmission cost is considered as the performance metric. The generated queries are then injected into the sensor network, where the second optimization phase is performed. The main idea of the in-network optimization is to simultaneously acquire data for all the synthetic queries during certain time interval. The number of messages in the network is reduced by dynamically routing message dissemination such that data aggregation is done as soon as possible with the involvement of fewer nodes. In addition, by taking advantage of the broadcast nature of radio transmission, acquired data is transmitted to satisfy all the queries that need the data. Mapping and calculation is used to obtain corresponding results for user queries after the sensor network returns results for the synthetic queries.

In [93], Multiple user queries (UQs) are merged into one network query (NQ) and then user data streams are extracted from network data streams, in order to efficiently enable multiple applications on top of a single sensor network. The basic idea is that as new user queries arrive, they can be merged into the current network query to create a new network query with lower selectivity that can answer all the existing user queries. The resulting network query then is sent to the sensor network, which produces data for the query. User query answers are then extracted from the network query data stream by carefully doing a down-sampling and attribute projection. As an illustrative example, consider the following user queries:

```
UQ1: SELECT nodeid, light FROM sensors SAMPLE PERIOD 5s
UQ2: SELECT nodeid, light FROM sensors SAMPLE PERIOD 15s
UQ3: SELECT light FROM sensors SAMPLE PERIOD 50s
UQ4: SELECT nodeid, light, temp FROM sensors WHERE nodeid=1 AND temp
     >25 SAMPLE PERIOD 20s
```

These queries can be merged into the following network query:

```
NQ1: SELECT nodeid, light, temp FROM sensors SAMPLE PERIOD 5s
```

The stream of data produced for NQ1 is then used to create data required for each user query. For example, to answer user query UQ1, the query processor selects one data record out of each three records and drops the *temp* attribute. Similarly, for answering UQ4, the query processor selects one data record out of each four records and outputs the record only when $nodeid = 1$ and $temp > 25$.

Differently to merging queries into a single query that is selective enough to be used to answer all the user queries, [73] proposes to rewrite a new query based on the existing queries and evaluate it in the base station instead of injecting it to the sensor network. Testing query rewritability given the existing set of candidate queries is NP-complete. Therefore, a heuristic approach is used for this purpose. As an example, consider the following existing queries Q_1 and Q_2, and the new query Q_{new}.

```
Q₁: SELECT nodeid, temp FROM sensors WHERE temp > 15 SAMPLE PERIOD 4s
Q₂: SELECT nodeid, light FROM sensors WHERE light > 200 SAMPLE PERIOD
     2s
Qnew: SELECT nodeid, temp, light FROM sensors WHERE temp > 25 AND
     light > 250 SAMPLE PERIOD 8s
```

Q_{new} can be rewritten to Q'_{new} based on Q_1 and Q_2 as follows.

```
Q'new: SELECT nodeid, temp, light FROM Q₁, Q₂ WHERE Q₁.nodeid = Q₂.
     nodeid AND temp > 25 AND light > 250 SAMPLE PERIOD 8s
```

Optimizing multiple aggregate queries in sensor networks with the objective of minimizing the communication cost while taking into account the processing limitations of the sensor nodes is studied in [130, 131]. Users pose their queries to the base station (or gateway) node, which instead of immediately sending the queries to the sensor network for in-network aggregation, it collects the queries in batches. The aggregate queries with the same aggregation operator are grouped and optimized. The batches are dispatched to the network once every *epoch*. Each epoch consist of two phases: *query preparation* and *result propagation*. *sum* queries and its relatives (e.g., average, count) and *min* queries are the two aggregate query classes that are considered. The aggregation is performed over sensors that are located in the queried rectangle. The sensor network is assumed to be given by a dissemination tree having the gateway node as its root.

Multi-query optimization in [130, 131] is based on the notion of *equivalent classes* (ECs). An equivalent class is the union of all regions that are covered by the same set of sensors. Thus, each query can be represented by a bit vector having 1's for the

ECs that contain the query, and 0's for other ECs. Each node expresses the queries in terms of ECs that intersect its subtree. For determining these ECs, the *bounding box* of the subtree is calculated, which is the smallest rectangle that contains all nodes in the subtree. Each node can calculate the bounding box of its subtree and the ECs that intersect its bounding box. Therefore, the query dissemination and multi-query optimization can be achieved in a distributed manner, hence, reducing communication and computation and memory usage per node. For *sum* queries, an optimal algorithm in terms of communication cost is proposed in this work. However, the problem of minimizing communication cost for *min* queries are proved to be NP-hard.

A utility-driven architecture for supporting geographically distributed multi-user radar sensor network is proposed in [74]. In such a system, diverse users, having different priorities, pose various queries that sometimes have contradicting requirements. The goal of the proposed architecture is to maximize data sharing among users and avoid duplicate data requests, compress and prioritize data transmission based on the importance of the data to the end users, and to gracefully degrade user utility when bandwidth is limited. The first step of the operation of the system is multi-query aggregation in which multiple queries are aggregated into a single query. The aggregation is performed at each epoch. Query aggregation minimizes the number of radar scans, a time and energy-intensive operation, and allows one-time data transmission to the base station. The base station sends the batched query at each epoch to the relevant radars. Each query is specified by its *query type*, e.g., tornado detection or 3D wind direction estimation, *region of interest*, *priority*, and *deadline*. Multiple user queries are aggregated into a single aggregated query plan, a pixelated spatial map for the areas covered by radars. A list of tripples $< qt_i, w_i, d_i >$ is assigned to each pixel, where qt_i, w_i, and d_i are, respectively, the type, weight, and deadline of the ith query. The weight of a pixel for a query is the importance of transmitting data from that pixel to the query. The weight is calculated as the multiplication of the priority of the query and the inverse of its deadline value. The unified query plan is used by the radars for progressive compression and scheduling. It is also used by the base station for global data transmission control.

2.4.3 Multi-query Optimization in Stream Processing Systems

Multi-query optimization techniques are also considered in stream processing systems, which sit one level above the sensor networks. For example, In the AdaptiveCQ framework [129], for efficient processing of multiple continuous queries, the intermediate results of queries are shared at a fine level without materializing them on disk. In AdaptiveCQ, multiple join operators can be shared amongst several queries, thus improving query performance.

AdaptiveCQ is based on an extension of the *eddy* query processing technique introduced in [10], which is originally designed for one-time queries. An eddy is a query processing operator that continuously reorders the operators in a query plan on a tuple-by-tuple basis in order to adapt to fluctuations in input data streams. Data flows into

the eddy from input streams. The eddy routes tuples to operators, which run as independent threads. Operators return tuples to the eddy after performing their operations. The eddy sends a tuple to the output only when it has been handled by all the operators. The eddy adaptively chooses an order to route each tuple through the operators.

Based on the eddy query processing technique, [82] proposes an adaptive continuous query processing architecture called *CACQ*. The *CACQ* uses the eddy operator to provide adaptivity to the changing query workload, data delivery rates, and overall system performance. It explicitly attaches the work that has been performed on a tuple to the tuple. This encoded work, which is refered to as the *lineage* of the tuple, allows operators from several queries to be applied to a single tuple. Hence multiple overlapping queries can share the work and state information. A *predicate index* is used for applying various selections to each single tuple. Roughly speaking, a predicate index takes multiple predicates and a tuple, and returns the set of predicates that accept the tuple in an efficient way. Joins are split into unary operators called *state modules* that allow pipelined join computation and sharing of state between joins in different queries.

SQPR [60] proposes a query planner for distributed stream processing systems, which exploits overlaps between queries and sharing partial results with the objective of efficient resource allocation. The optimal query plan is formulated as a single constrained optimization problem that provides answer to queries while resource utilization is minimized and the allocation objective of the system, e.g., load balancing among hosts, is respected. When new queries arrive, past query plans are revisited in order to keep the efficiency of resource allocation. It also revisits running query plans when their resource requirements changes. The solution to the optimal query plan problem, governs query admission, operator placement, and query reuse.

The interested user can refer to further work in multi-query optimization in sensor networks and stream processing systems such as [13, 64, 81, 113, 121, 135, 141, 144].

2.5   Truthful Elicitation of Sensor Measurements

In this section we review the state of the art in eliciting truthful data from participants in participatory sensing. Most of these works employ *game theoretic* or *mechanism design* approaches. The approach we take in Chapter 4 for truthfully eliciting data from participants is also a mechanism design approach. Therefore, we start this section by a brief introduction to the concept of mechanism design. We assume that the reader is familiar with the basic concepts of game theory.

2.5.1   Mechanism Design

Mechanism Design (MD) is a sub-field of micro-economics and game theory that deals with implementing desirable system-wide solutions to problems that involve self-interested agents, who have private information about their preferences for different outcomes. Each agent i has a *type* $\theta_i \in \Theta_i$, that specifies its preferences over different outcomes. We denote by $u_i(o, \theta_i)$ the utility of agent i with type θ_i for outcome $o \in \mathcal{O}$, which states

the "goodness" of the outcome for the agent. For example, in a multi-item auction, agents are bidders and the type of each agent is its valuation of the items being auctioned. The set of outcomes of the auction, $\mathcal{O}$, is the set of all possible ways that items can be assigned to bidders. The utility of agent i for outcome o, given its type θ_i is its valuation for the outcome o, provided that participating in the auction does not incur any cost to the agent.

The system-wide goal of a mechanism is defined by a *social choice function*, which selects the optimal outcome given agent types.

Definition 2.4. *(Social choice function)* Social choice function $f : \Theta_1 \times \cdots \times \Theta_I \to \mathcal{O}$ chooses an outcome $f(\theta) \in \mathcal{O}$, given types $\theta = (\theta_1, \ldots, \theta_I)$, where I is the number of agents.

The goal of mechanism design is to design a game and its rules in such a way that despite agent's self-interest, the social choice function picks the desired outcome for the designer. A mechanism defines the strategies available (e.g., agents can bid only once) and the method that chooses an outcome based on agent strategies (e.g., the item is sold to the agent with highest bid for the price of the second highest bid). The central authority that enforces the rules and coordinates the game is often called the *center*.

Definition 2.5. *(Mechanism)* A mechanism $\mathcal{M} = (\Sigma_1, \ldots, \Sigma_I, g(.))$ defines the set of strategies Σ_i available to each agent, and an outcome function $g : \Sigma_1 \times \cdots \times \Sigma_I \to \mathcal{O}$, such that $g(s)$ is the outcome implemented by the mechanism for strategy profile $s = (s_1, \ldots, s_I)$.

Agents are said to be *rational* if their goal is to maximize their utility. Therefore, agent i is rational if it chooses its best strategy s_i^* such that

$$s_i^* = \arg\max_{s_i \in \Sigma_i} E[u_i(g(s_1, \ldots, s_I)], \tag{2.9}$$

where $E[.]$ is the expectation operator.

We say that mechanism $\mathcal{M}$ with outcome $g(.)$, implements social function $f(\theta)$ if the outcome computed with agent strategies in *equilibrium* is a solution to the social choice function for all possible agent preferences.

Definition 2.6. *(Mechanism implementation)* A mechanism $\mathcal{M} = (\Sigma_1, \ldots, \Sigma_I, g(.))$ *implements* social choice function $f(\theta)$ if $g(s_1^*(\theta_1), \ldots, s_I^*(\theta_I)) = f(\theta)$, for all $(\theta_1, \ldots, \theta_I) \in \Theta_1 \times \cdots \times \Theta_I$, where strategy profile $(s_1^*, \ldots, s_I^*)$ is an *equilibrium* solution to the game induced by $\mathcal{M}$.

In the following we provide some other important definitions.

Definition 2.7. *(Efficient mechanism)* A mechanism is *efficient* if it selects the outcome that maximizes total utility (social welfare).

Definition 2.8. *(Direct mechanism)* A direct mechanism is a mechanism in which the strategy space available to each agent is its type, i.e., $\Sigma_i = \Theta_i$.

Definition 2.9. *(Incentive compatible mechanism)* A direct mechanism is *incentive compatible* if it has an equilibrium s^* where $s_i^*(\theta_i) = \theta_i$ for all i and $\theta_i \in \Theta_i$. That is, telling the truth by all agents is an equilibrium.

Incentive compatibility is a desirable property of mechanisms as it shows that agents have incentives to truthfully reveal their types. Another desirable property of mechanisms is *individual rationality* as it incentivizes the agents to participate in the mechanisms. Individual rationality states that each agent receives (in expectation) more or zero utility by participating in the mechanisms as compared to not participating.

2.5.2 Overview of Existing Work in Truthful Data Elicitation

One of the most well-known incentive compatible and individually rational mechanisms that can work in many settings is Vickrey-Clarke-Groves (VCG) mechanism [84]. VCG is an efficient mechanism, which selects an outcome that maximizes the total utility and pays every agent its marginal contribution to the final outcome. Utility functions of agents are *quasi-linear*. That is, the utility is the sum of the payoff of the agent from the outcome of the mechanism and a monetary transfer from/to the agent. The payoff of an agent from the outcome of the mechanism only depends on its own type. However, when the agent's payoff depends on other agents types, i.e., there exists *interdependency* among valuations, it is proved that achieving both efficiency and incentive compatibility is impossible [55, 85]. Mezzetti has shown that by using a two-stage mechanism, where in the first stage the final outcome is found and in the second stage the payments are calculated, we can overcome the impossibility of an efficient mechanism when valuations are interdependent [88].

2.5.2.1 Mechanisms for Bandwidth Allocation

Authors in [66] propose a mechanism, based on [88], for efficient bandwidth allocation in tactical data networks in which agents have interdependent valuations. The network has a limited bandwidth and agents are tracking objects. The data from an agent about objects can be valuable to other agents. However, agents are self-interested and prefer receiving data than sharing data, so that other agents allocate bandwidth for transferring data. The information about the quality of an agent's data is not revealed to the other agents. The value of an agent for receiving data about its objects of interest not only depends on its private data but also on the data of other agents. This implies interdependent valuation. The proposed mechanism incentivizes the agents to truthfully reveal their private data about the tracked objects in a way that the bandwidth is optimally allocated. In [26] a similar problem is considered. In this work, valuation functions of agents are based on the Kalman Filter and the mechanism is similar to the classical VCG mechanism. The valuation function of each agent is used to valuate the information gain of data that the agent receives from other agents. The agents transfer their valuation functions and their observations (about the objects being tracked) to

the center. The center then calculates the optimal allocation (i.e., the allocation that maximizes the social welfare) and the payments of each agent.

2.5.2.2 Mechanisms for Task Allocation

Porter et al. [102] consider a task allocation setting in which, in addition to task execution costs, the probability of success of each task is private information for the agents. Several settings are considered, which include single task, multiple tasks with combinatorial properties, and multiple tasks with dependencies. For some settings mechanisms are proposed that satisfy the goals in terms of incentive compatibility, efficiency, individual rationality, and budget balance. In other settings, where achieving desired goals are proved impossible, mechanisms are provided for a weaker set of goals. In Chapter 4, we propose mechanisms for eliciting truthful data from participants that are inline with the work in [102].

In [102], the mechanisms rely on the reports from the agents about their probability of success. However, even when agents are truthful, the actual success probability of an agent might not be the reported one. Ramchurn et al. [105] extend the work in [102] by introducing the notion of *trust* as the combination of other agents' perceptions of the probability of success of a particular agent. In this manner, interdependent valuation is introduced since each agent uses other agents reports to build trust values. The trust-based mechanism proposed, which is evidently a case of the general result of Mezzetti [88], is efficient, incentive compatible and individually rational.

In a service-oriented computing environment, service consumers procure self-interested distributed service providers to complete computational tasks that have deadlines. In this kind of task allocation settings, the execution time of the tasks cannot be determined with certainty. Moreover, service consumers might procure redundant providers to perform a task in order to increase the probability of success and mitigate the uncertainty in service execution time. [102] and [105] do not consider redundant task allocation and uncertain task execution times. Stein et al. [123] propose incentive compatible mechanisms in different pricing schemes for truthfully eliciting private information from service providers about their capabilities.

2.5.2.3 Mechanisms Based on Scoring Rules

Scoring rules [110] are used for evaluating and rewarding probabilistic predictions about an event. Scoring rules incentivize the predictor to truthfully reveal its forecast and maximize its reward. The score is computed based on the reported probability distribution and the actual event that is finally observed. A scoring rule is said to be *proper* if by reporting its true probability, the forecaster maximizes its expected score, and by reporting any other probability it receives a strictly lower score.

Payment schemes based on *proper scoring rules* are proposed in [145] for truthfully eliciting information represented by discrete random variables. The proposed payment scheme results in truth-telling when all the agents and the center have the same belief

(i.e., underlying probability distribution) about the state of the world. In the case of probability distributions that vary very slightly among agents, an approach is proposed for designing mechanisms that are truthful and robust to a certain degree of variations in beliefs. Finally, when agents have access to extra unsold information that impacts the information requested by the center, it is shown that designing optimal mechanisms is computationally hard.

Peer-prediction method, proposed in [89], is a mechanism based on proper scoring rules for eliciting honest feedback, such as ratings of items, from the agents when independent and objective outcomes are not available. This method uses the report of one agent to update a probability distribution for the report of the *reference* agent. The score assigned to an agent is calculated not by comparing its rating to the ratings of other agents, but by comparing the likelihood of a reference agent's ratings and the actual rating of the reference agent. By appropriately *scaling* scoring rules, not only honest reporting is guaranteed, but also the agents are incentivized to exert costly effort for acquiring information and reporting the ratings. The scaling is performed in a way that reporting honestly is better for the agents by at least a margin Δ.

The method proposed in [89] can lead to arbitrary high payments, which are provided by the mechanism. Jurca and Falting in [59] propose a method for computing minimal budget required for payments in incentive compatible mechanisms for achieving a certain margin Δ. The optimal payments are represented by linear optimization problems. Experiments show that these linear programs can be computed using linear program solvers. It is shown that using several reference raters for scoring one feedback, can further reduce the budget required for the mechanism. Another approach for reducing the optimal expected budget is to use probabilistic filtering to filter out false reports. This is based on the fact that lying agents, for achieving considerable benefit, must provide reports that are significantly different from the average reports given by honest agents. [134] extends the work of [89] and [59] for a dynamic setting in which the quality of the products changes over time. A Markov process can model the quality changes over time.

The authors in [99] propose two-stage mechanisms based on *strictly proper scoring rules* for truthful elicitation of probabilistic estimates. In the base case, the center first announces a desired estimation precision. Then it selects an agent with the minimum reported cost for the required precision. In the second stage, the center announces its scoring rule. The selected agent generates and reports its prediction and the precision of the prediction. After observing the actual outcome, the center issues a payment to the agent based on the scoring rule. This mechanism is proved to be incentive compatible and individually rational. Two extensions of the base mechanism are considered. In the first case, the required precision cannot be satisfied by only one agent. In this case the base mechanism is extended to select multiple agents and combine their predictions to achieve the desired precision. In the second case, the assumption of access to the actual outcome is removed. In this case, the base mechanism is extended in a way that the reported estimates of the selected agents are combined to serve as ground truth.

These mechanisms are shown to have lower variance in payment compared to the peer prediction approach [89].

2.5.2.4 Mechanisms in Mobile Sensing

An incentive compatible mechanism for eliciting truthful measurements in community sensing, called *Peer Truth Serum*, is proposed in [38]. Given the reported measurement, a payment is calculated based on a reference estimate using a publicly available prior probability distribution for the variable being measured. The reference value is taken from the model, which is updated using other reports received in the same time interval. When the agents adopt a publicly known distribution as their prior distribution, the Peer Truth Serum incentivizes truthful reporting. In the case that agents are more informed about the event, their prior distributions might differ from the public distribution in the sense that their distribution is closer to the true distribution of the event. In this case truth-telling is not guaranteed, but it is shown that Peer Truth Serum incentivizes *helpful* reports in a way that the reports help the public distribution to converge faster to the true distribution.

Truthfully eliciting data in participatory sensing when participants are privacy conscious is considered in [122]. The privacy tradeoffs in participatory sensing is modeled as an adaptive submodular optimization problem. For this problem modeling, the utility functions are required to be submodular. The aim is to design mechanisms for selecting participants to provide data for the application in order to achieve a near-optimal utility for the applications while the budget limit is respected. For protecting privacy, the self-interested agents obfuscate the data that they report (e.g., their locations). *Sensing profile* of an agent is the set of locations that are covered by that agent. As obfuscation, agents share with the system a set of sensing profiles instead of their current sensing profile. Only after being selected by the center, the agents reveal their actual sensing profile. Each agent declares a cost for reporting its private data. After receiving cost information from the agents, the center iteratively selects an agent, computes a payment to be issued to that agent, and receives the private information of the agent. This mechanism, called *SeqTGreedy*, is shown to be truthful regarding reporting costs.

Last but not least, a mobile commerce scenario is considered in [22], where users get benefits from service providers (companies) by reporting their location with a certain level of granularity. The privacy mechanisms proposed in this work help companies motivate users by means of monetary incentives to obtain more accurate location information. An information theoretic approach is used to quantify the anonymity level of each mobile user. The privacy game is designed in a way that the equilibrium point can be moved towards a desirable location granularity. The unknown parameters of users, such as their risk factors, are learned by the companies using an iterative distributed algorithm and regression learning.

2.6 Quality Assessment for Sensor Date Streams

One of the essential tasks in sensor networks is to assess the quality of the sensor data in order to guarantee reliable and meaningful data for the applications of the sensor network. The reason is that sensor readings might be affected by faults. Faults encompass errors due to temporary malfunctioning of sensors, loss of connectivity, resource limitation, interference, etc. Quality assessment methods normally assign a *quality score* to each sensor reading, which indicates how close the reading is to the true value. When the quality score of a sensor reading is lower than a threshold, the reading is considered as an *outlier*. There exist several work in the literature that try to detect outliers in sensor data streams. These approaches might also, directly or indirectly, be used as quality assessment methods. It is worthwhile to note that the terms *quality*, *trust*, and *trustworthiness* have been used interchangeably in the literature. In this thesis we use the term *quality* to determine how close a sensor measurement is to the true value of the phenomenon at the location that the measurement has been taken.

2.6.1 Review of Sensor Data Quality Assessment Methods

In this section we review state of the art mechanisms in data quality assessment in sensor networks. We present the approaches based on *value similarity in a fixed neighborhood* and the techniques based on *Bayesian and probability theory*. We also present relevant related research work in *computing sensor error bounds*. Finally, we briefly review the existing *outlier detection techniques*.

2.6.1.1 Value Similarity in Fixed Neighborhood

There exist several techniques for quality assessment of sensor data that use multiple measurements of the same phenomenon collected from different sensors that are located in a fixed neighborhood. A cyclic trustworthiness assessment framework is proposed in [76]. In this framework, the trust scores of sensor data streams affect the trust scores of the network nodes. Inversely, the trust scores of network nodes affect the trust scores of sensor data streams. Therefore, there is an inter-dependency between trust scores of nodes and streams. The trust scores of data items are computed based on the *value similarity* and *provenance similarity* between them. Value similarity states that the more similar the values of data items are, the more trustworthy they are. Provenance similarity indicates that the data items with similar values are more trustworthy if their provenance is more different. Provenance of a data item is the path it traverses from its source node to the aggregator node. Value similarity can be calculated using any reasonable similarity model. The similarity model introduced in [76] is presented in the following.

In the cyclic trust computation framework three types of trust scores are maintained for nodes and data items: *current*, *intermediate*, and *next* trust scores. In each cycle of the framework the following steps are performed: (1) The current trust scores of data items are calculated based on the current trust scores of the nodes according to
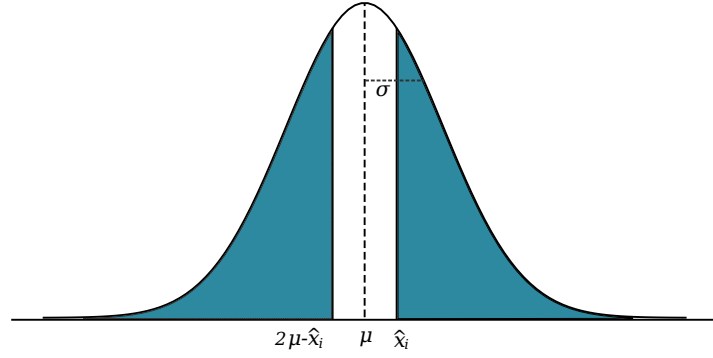
Figure 2.2: Trust (or quality) score computation based on value similarity. The distance between the sensor reading $\hat{x}_i$ and the mean of the distribution μ, determines the trust score.

the provenance of the data items. (2) The intermediate trust scores of data items are computed first based on value similarity among data items. This score is then adjusted based on their provenance similarity. (3) The next trust score of each data item is calculated by linearly combining its current and intermediate scores. (4) The intermediate trust scores of nodes are computed as the average of the next trust scores of the data items that are generated by or passed trough them. (5) The next trust score of each node is calculated as a linear combination of its current and intermediate scores.

The data items in the same event are assumed in [76] to follow a normal distribution since it represents well natural phenomena. Intuitively, the values that are closer to the mean of the distribution can be regarded as more trustworthy as compared to the values that are far from the mean. Based on this observation, the intermediate trust score of a data item based on its similarity with other data items can be computed using the cumulative probability of the normal distribution as shown in Figure 2.2. In this figure, $\hat{x}_i$ is the value of the data item (i.e., a reading from sensor s_i), μ and σ^2 are the mean and variance of the normal distribution of the values in the same event, respectively. The intermediate trust score of $\hat{x}_i$, when $\hat{x}_i > \mu$, can be computed as the following.

$$score(\hat{x}_i) = 2 \int_{\hat{x}_i}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(v-\mu)^2}{2\sigma^2}} \, \mathrm{d}v. \tag{2.10}$$

A basic approach for determining faulty sensors is to take advantage of the spatial correlation among sensor readings by comparing the values with neighboring sensors. Based on *majority voting* scheme, the sensor reading is sent to each sensor node in the neighborhood. The sensor node compares its own value with the received value. If the difference is more than a pre-defined threshold, the sensor node gives a negative vote. If the number of negative votes is greater than the positive votes, the reading is considered as faulty. However, if the number of faulty sensors in the neighborhood is high, the simple majority voting scheme fails. Based on the assumption that closer sensors are more correlated, the *weighted voting* scheme assigns more weight to the votes of closer sensors. However, this assumption does not always hold. Moreover, when the closer sensors are faulty the scheme fails. In [137] a distributed approach for detection

of faulty values in the presence of events is presented to overcome the shortcomings of these two voting schemes. The approach proceeds as follows: (I) it builds a correlation network between sensors that is based on value similarity (II) it views the correlation network as a Markov chain and it computes a rank for every sensor in terms of transition probabilities from neighbors to a sensor (III) it uses a voting mechanism to detect faulty values that consists of two phases: (i) self-diagnosis phase, where every sensor verifies whether its current value is unusual (ii) neighbor-diagnosis phase is executed if the self diagnosis indicated unusual value, where the sensor node consults its neighbors to further validate if the value is faulty.

The correlation between the readings of two sensors s_i and s_j is calculated based on the Extended Jaccard similarity measure [124] as follows.

$$corr_{i,j} = \frac{b_i(t) \cdot b_j(t)}{||b_i(t)||_2^2 + ||b_j(t)||_2^2 - b_i(t) \cdot b_j(t)}, \tag{2.11}$$

where $b_i(t)$ and $b_j(t)$ are the reading vectors of s_i and s_j at time t in a sliding window of size Δt. In the Markov chain, the transition probability from sensor s_i to sensor s_j is given by the following:

$$p_{i,j} = \frac{corr_{i,j}}{\sum_{k \in \mathcal{N}_i} corr_{i,k}}, \tag{2.12}$$

where $\mathcal{N}_i$ represents the neighborhood of sensor s_i. The votes of the sensors are weighted by their *SensorRank* which is given by the following formula:

$$rank_i = \sum_{j \in \mathcal{N}_i} rank_j \cdot p_{j,i}. \tag{2.13}$$

In [30] an algorithm for faulty sensor identification is presented, where the algorithm uses a fixed neighborhood $\mathcal{N}_i$, of radius r centered at each sensor s_i. It decides if a sensor reading $\hat{x}_i$ is faulty as follows: (I) it computes the difference between the sensor reading and the median of sensor readings in the neighborhood $d_i = \hat{x}_i - med^{(i)}$; (II) it computes d_j for each sensor s_j in $\mathcal{N}_i$; (III) it computes the average and variance of the differences; (IV) it computes the Z-score for d_i, where s_i is considered faulty if the Z-score exceed a user defined threshold. This work is based on the assumption that faulty sensors are spatially independent, while events are spatially correlated.

2.6.1.2 Bayesian and Probabilistic Approaches

As a different approach towards distinguishing events and errors in sensor networks, [71] presents a distributed Bayesian algorithm for fault-tolerant event detection. It estimates the probability of a binary variable that a sensor reading is part of an event region given values of such a binary variable from each neighbor in a fixed neighborhood of radius r. The model gives the same weight to the evidence from each neighbor in the fixed neighborhood. It assumes that sensors are densely deployed such that nearby sensors are likely to have similar values in response to the same event unless they are at the boundary of the event. It assumes that sensor faults are likely to be stochastically uncorrelated, while event measurements are likely to be spatially correlated.

In [36] a framework for cleaning and querying noisy sensor readings is proposed. It consists of a cleaning module and a query processing module. In an online fashion, the cleaning module computes the uncertainty models of the unknown true values by following a Bayesian approach. Using the uncertainty models produced by the cleaning module, the query processing module provides answers to queries posed on the noisy data. The cleaning module has three inputs: noisy sensor readings, sensor error models, and prior distribution of sensor values. The noise is assumed to be normally distributed with zero mean and a known standard deviation of σ. The true value x hence, follows a normal distribution with mean $\mu = x$ and variance σ^2. Using Bayes' theorem the posterior probability of the true value is computed as:

$$p(x|\hat{x}) = \frac{p(\hat{x}|x)p(x)}{p(\hat{x})}, \tag{2.14}$$

where $\hat{x}$ is the sensor value. When the readings of a sensor s follow the Gaussian distribution $\mathcal{N}(\mu_s, \sigma_s^2)$, the posterior probability $p(x|\hat{x})$ also follows a Gaussian distribution $\mathcal{N}(\mu_x, \sigma_x^2)$, where μ_x and σ_x^2 can be computed using the Bayes' theorem and the properties of the Gaussian distribution.

In [96] a Bayesian faulty sensor detection approach based on a sort of majority voting is proposed. The essential assumption is that the non-faulty sensors generate very similar values. Moreover, it is assumed that data from non-faulty sensors is locally modeled by a linear model. First, it selects a subset of sensors that can best represent the data by casting the problem as a maximum a posteriori probability (MAP) problem. In this way, with a Bayesian detection approach a subset of sensors is selected that maximizes the posterior probability of the subset, i.e. the conditional probability that the subset is non-faulty given the data. Then, the selected subset is used to determine if a given sensor is faulty or not.

A fixed structure Bayesian Network model is applied in [133] for anomaly (event) detection in gas monitoring sensor networks for underground coal mines in order to capture spatio-temporal correlations. The structure of the network is constructed by embedding the time series in a d-dimensional phase space and creating a dependency between each node and the nodes that precedes it in time. The same network structure is used as "subnets" of the Bayesian Network model for the combination of multiple sensors. Each node is modeled as a one dimensional Gaussian. Using the maximum likelihood (ML) algorithm the single unknown parameter (i.e., the conditional probabilities that quantify each node) is learned from the training data. Then, the learned network is used to detect anomalies by measuring how well observations fit the Bayesian Network model, by computing their likelihood values. An observation is identified as anomaly if its likelihood value is low.

In [126] a method, called *True-Alarm* is presented for finding trustworthy alarms and meaningful objects in sensor networks. The authors argue that the methods based on the *neighbor similarity hypothesis*, such as [137] and [71], can cause false alarms if the sensors in the neighborhood, even when reliable, cannot detect the object's activity because of

the distance. The trustworthiness of a value (alarm) is defined as the probability that the value is correct. The trustworthiness of an object is defined as the probability that the object really exists. The trustworthiness is defined in terms of a conditional alarm trustworthiness given a specific object causing the alarm. The trustworthiness of a value is given by the maximum conditional alarm trustworthiness given the set of sensors detecting the alarm (the monitoring sensor set of the object). The trustworthiness of an object is given by the average of all conditional alarm trustworthiness values over the monitoring sensor set of the object.

2.6.1.3 Sensor Accuracy Bound Computation

The phenomenon model is used in [49] to calculate maximum feasible phenomenon change over a given temporal and spatial distance. Sensor drift from a time to another and sensor noise (both can be obtained from sensor specification) represent the bounds on sensor inaccuracy. Computing sensing accuracy is an iterative process in which the accuracy bound is updated based on the previous accuracy and the drift from last time and next time. Then the signal range is computed which represents the feasible phenomenon range at each time given the calibrated sensor reading and sensor accuracy at that time. This is also an iterative process in which the bounds based on the phenomenon model and signal range of other readings are used. Measurements that fall outside of the predicted bounds, computed based on the phenomenon and sensor models, are considered as erroneous.

2.6.1.4 Outlier Detection Techniques

A comprehensive review of methods for outlier detection in wireless sensor networks is presented in [143], where many of them are relevant to the task of assessing quality of sensor values. The state of the art techniques are classified in the following main categories:

- *Statistical-based approaches:* use a statistical model and calculate how well a sensor reading fits the model. In these approaches, which are also called *model-based* approaches, it is assumed that the distribution of the data is represented by a statistical model. The model can sometimes be learned from the data itself. Given the model, if the probability of a sensor reading being generated by that model is lower than a threshold, the sensor reading is considered as outlier.

  For example, in the approach proposed in [15] each sensor node learns the statistical distribution of the difference between its measurements and the measurements of its neighbors. The distribution of the difference between the measurements of each node at different times is also learned. Given these distributions, every new measurement can be tested for errors. The probability of observing a difference d which is more extreme than d_i is given by its p-value. Given the p-values of sensor nodes for their temporal differences and the differences with their neighbors, and a significant level, each measurement can be classified as standard measurement,

point failure, or common event. The essential assumption of this approach is that the sensor readings in a neighborhood are strongly correlated. Moreover readings of each sensor over time must be correlated.

- *Nearest neighbor-based approaches:* use a distance measure to determine how close a sensor reading is to the readings of its nearest neighbors. If the distance is more than a pre-specified threshold, the reading is considered as outlier. For example, the approach proposed in [30] falls into this category.

- *Clustering-based approaches:* group sensor readings into clusters based on certain similarity metric. A sensor reading which does not belong to any cluster is considered as outlier. Sensor readings that form a very small cluster compared to the size of the other clusters are also considered as outliers. The distributed anomaly detection approach based on clustering proposed in [104] belongs to this category. The average inter-cluster distance of K nearest neighbors clusters are used to determine which clusters are anomalous.

- *Classification-based approaches:* use a classifier that is trained on a set of training data to classify new senor readings as outlier or normal. Bayesian detection and Bayesian Network-based approaches, such as [36, 71, 96, 133], fall into this category.

Chapter 3

Utility-driven Data Acquisition in Participatory Sensing

3.1 Introduction

Participatory sensing is becoming a popular paradigm for collecting and sharing data about phenomena of social interest, such as air quality, well-being, traffic, etc. Even though some people might altruistically participate in such data collection systems, we believe that adequate incentives must be provided to people to encourage more participation. The burden that participation imposes on the participants, e.g., battery and network consumption and privacy leakage, should be compensated to guarantee long-term *sustainability* of the system. Moreover, in a popular participatory sensing environment, there can be many users/applications that are interested in the data being collected and pose different types of queries, instant or continuous ones. At the same time, some of the users may participate in the sensor data collection. Such participatory sensing system can be envisioned by introducing some sort of incentives, e.g., payments from the querying user, to the users from whom the data for the query is collected. It is critical for the sustainability of the system to provide to the users as much utility as possible. In this context, utility is defined as the difference between the value of the query results to the users and the price they pay for obtaining the results.

There exists a great number of works in the area of sensor data acquisition, which either have a single application-specific objective, e.g., achieving complete coverage of the sensing field [116], or assume certain structures for the utility functions, e.g., submodularity as in [16, 69, 70, 115]. Similarly, there is a large body of work in the context of multi-query optimization in sensor networks and in stream processing systems, e.g., [73, 129, 130]. However, the existing approaches cannot be directly applied to the context of participatory sensing for the following reasons: 1) because of the uncontrolled

mobility of the participants, the query processor needs to deal with data unavailability; and 2) there is a lack of sophisticated utility considerations in the existing work.

In summary, our main contributions in this chapter are the following:

1. We propose a data acquisition framework in the context of participatory sensing that takes into account the factors pertinent to this context and efficiently shares sensor data among queries of different types, so as to enable sustainability. Queries for sensor data come from multiple different applications or users that can have any arbitrary utility considerations.

2. We formulate the optimal data acquisition problem as a multi-query optimization with the objective of maximizing the total utility for the queries and propose efficient heuristic solutions for various query types and query mixes.

3. Important query categories, including one-shot and continuous queries, in the context of participatory sensing are considered and efficient data acquisition algorithms are proposed for each query type as well as the combination of different query types.

4. We verify the effectiveness of our approach through extensive simulations on real and synthetic data traces.

The remainder of this chapter is organized as follows. In Section 3.2, we introduce our context and formally define the problem. We present heuristic algorithms for sensor scheduling in Section 3.3 and evaluate those algorithms experimentally in Section 3.4. We review the related work in Section 3.5 and finally we conclude this chapter in Section 3.6.

3.2 The Context

In a participatory sensing system several participants carrying heterogeneous sensing devices move in a certain region. The sensing devices communicate with a server, which is called the *aggregator*. Sensing devices take a measurement only when they are selected by the aggregator to do so. Participants ask for a payment for each measurement they provide. Each sensor has a specific sensing range. Each measurement includes a sensor-specific inherent inaccuracy. In this chapter, we use the term sensor to refer to the actual sensor on the sensing device, the sensing device, or even the combination of the participant and the sensing device she carries.

End users (or applications) submit queries to the aggregator. The aggregator periodically collects the queries and tries to optimally answer them. Our optimization objective is to maximize the overall utility (or *social welfare*), since this objective matches our requirement for sustainable operation of the system, as opposed to data value maximization or cost minimization. Alternatively, an egalitarian approach could be followed, where the number of users with positive utility is maximized. Utility maximization can be achieved by selecting appropriate sensors for providing measurements, considering
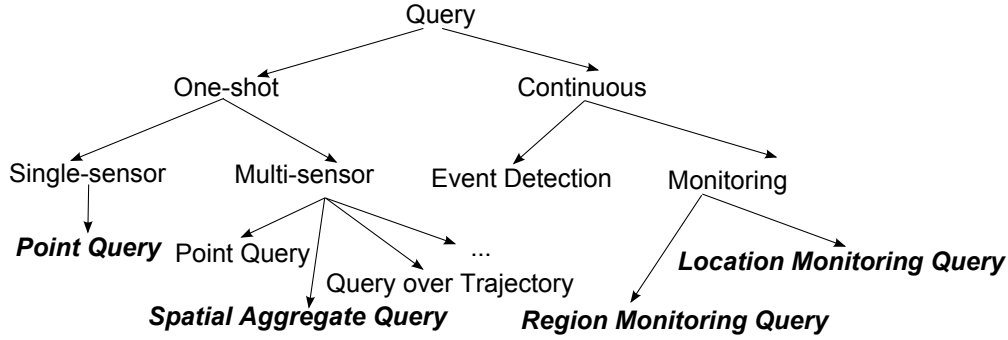
Figure 3.1: Query categories in the participatory sensing context. The query types in boldface are explicitly handled in this chapter.

the value of the measurements to the queries, the cost of obtaining such measurements, and exploiting possible common data requirements among queries. In a participatory sensing context with diverse set of end users who have different criteria for evaluating the quality of query results, the aggregator relies on the end users to provide a valuation function, $v_q(.)$, with each query q. This function returns the value, in real or virtual currency, of a set of measurements that can provide the answer to the query based on the quality of the measurements. Users have a limited budget to spend for obtaining query answers.

Queries issued by end users can fall into two major categories, namely *one-shot queries* and *continuous queries*. One-shot queries are executed only once, while continuous queries are continuously evaluated. Major one-shot queries in the participatory sensing context are *point queries*, *spatial aggregate queries over a region*, and *queries over trajectories*. Continuous queries can be split into two sub-categories of *monitoring queries* and *event detection queries*. *Single-sensor* queries only need one sensor reading while *multi-sensor* queries need multiple sensor readings. Figure 3.1 shows these categories and the query types that we handle explicitly in this chapter. Each query category is explained in more details later in this section. Table 3.1 summarizes our most frequently used notation in this chapter.

3.2.1 Problem Formulation

We assume, without loss of generality, that the system runs for a period of T, e.g., from 6 a.m. to 9 p.m. in a day. This period is discretized into several time slots of fixed length, e.g., 5 minutes. All the sensors communicate with a unique aggregator and if necessary, at the beginning of each time slot announce their location and price of providing a measurement at that location.

The objective is to acquire data for the queries from the available sensors in order to maximize the utility over T. Formally, we let $\mathcal{Q}$ denote the set of all queries issued from time 1 to T, $\mathcal{S}^t$ denote the set of available sensors at time slot t, and $K : \mathcal{Q} \to \times_{t=1}^{T} 2^{\mathcal{S}^t}$ define an allocation scheme that assigns sensors to each query. $Y(K, t)$ is a function that returns the set of sensors that are assigned to all queries at time t. We denote by
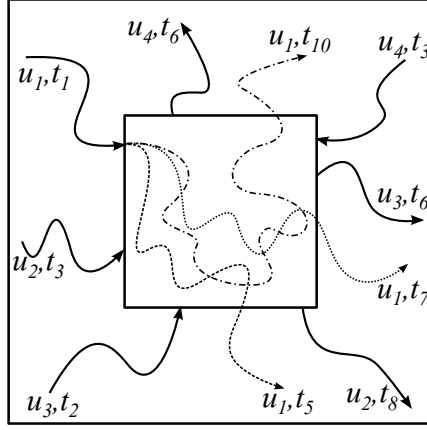
Figure 3.2: Random arrival and departure / unpredictable mobility patterns: user u_1 enters at time t_1 and can take three possible trajectories and exit at times t_5, t_7, or t_{10}.

$c_s(K, t)$, the cost of sensor s at time t given the allocation K. Let $\mathcal{K}$ denote the set of all possible allocation schemes. The goal is to find allocation $K^* \in \mathcal{K}$ that maximizes the *social welfare*:

$$K^* = \arg\max_{K \in \mathcal{K}} \Big(\sum_{q \in \mathcal{Q}} v_q(K(q)) - \sum_{t=1}^{T} \sum_{s \in Y(K,t)} c_s(K, t) \Big). \tag{3.1}$$

For solving the above problem we need to know in advance all the queries that will be issued over T, and the location and cost of all the sensors at each time slot. However, in a participatory sensing system, users must be able to submit new queries whenever they desire. Therefore, it is not realistic to ask the users to pose all their queries in the beginning of T. Due to the uncontrolled mobility of the sensors, their exact locations at a specific time slot cannot be determined a priori. Moreover, the cost of a sensor might vary from one time slot to another based on the preferences of the sensor owner. Due to the lack of access to all the required information to solve the long-term optimization problem (3.1), we resort to a *myopic* approach, in which we try to maximize the utility at the current time slot without considering the future state of the system. This approach would be further motivated in a "hotspot" monitoring setting (cf. Figure 3.2) described as follows: Consider a hotspot area, e.g., the downtown, of a city where users carrying smart phones continuously enter and exit, and roam around while they are inside it. In this case, the mix of available sensors in the hotspot area dynamically changes and short-term optimization towards monitoring sustainability becomes more important.

Let Q denote the set of all queries available at the current time slot t. Q can include one-shot queries issued for time t and continuous queries that started before or at t. Let S be the set of available sensors at t and c_s denote the reported cost of each sensor s. Let $M : Q \to 2^S$ define an allocation scheme that assigns sensors to each query. $Y(M)$ is a function that returns the set of sensors assigned to queries. Let $\mathcal{M}$ denote the set of

all possible allocation schemes. The goal is to find allocation $M^* \in \mathcal{M}$ that maximizes the total utility in the current time slot:

$$M^* = \underset{M \in \mathcal{M}}{\arg\max} \Big(\sum_{q \in Q} v_q(M(q)) - \sum_{s \in Y(M)} c_s \Big). \tag{3.2}$$

After finding the best allocation scheme, the cost of each selected sensor s is shared among queries that are answered using the measurement from s. We denote by $\pi_{s,q}$ the amount that query q pays for using data from sensor s. We must ensure that for each selected sensor s, the total payment from the queries using that sensor is equal to c_s. Moreover, for each query q, which is answered using sensors S_q, its utility must be positive, i.e., $v_q(S_q) - \sum_{s \in S_q} \pi_{q,s} > 0$.

3.2.2 One-shot Queries

We can distinguish between the queries that only need data from one sensor and queries that ask for several sensor readings. More specifically, spatial aggregate queries and queries over trajectories always require several sensor readings, whereas there exist some point queries that ask for only one sensor reading and some point queries that ask for more than one sensor reading. The former type of point queries is referred to as *single-sensor point queries* and the latter is referred to as *multiple-sensor point queries*. The reason for this distinction is that single-sensor queries can be treated more efficiently due to their special characteristics.

3.2.2.1 Point Queries

A user who is interested in knowing the value of a phenomenon at a certain location, submits a point query at that location to the system. The queries are required to come with a quality valuation function to valuate the quality of the sensor readings. Generally, the value of a sensor reading for an application is a function of the quality of that sensor reading and the quality of the sensor readings obtained so far. The number of samples required for finding the value of a phenomenon depends on the phenomenon itself and the trustworthiness of the sensors. For example, it might be necessary to take redundant measurements to assess the trustworthiness of a particular sensor that can be used for providing the measurements. For instance, a single-sensor point query q might have the following valuation function:

$$v_q(s) = \begin{cases} B_q \theta_{q,s} & \theta^q_{min} \leq \theta_{q,s} \leq 1, \\ 0 & \theta_{q,s} < \theta^q_{min}, \end{cases} \tag{3.3}$$

where $0 \leq \theta_{q,s} \leq 1$ is the quality of the sensor reading for q, θ^q_{min} is the minimum acceptable quality by the query, and B_q is the query budget. This implies that the user is willing to pay B_q for a sensor reading with the highest possible quality.

The quality of a sensor reading depends on the distance of the sensor from the queried location (more accurately, it depends on the correlation between the phenomenon value

at the queried location and the location of the sensor,) the inherent sensing inaccuracy, and the trustworthiness of the sensor. We assume that this dependency is given by a user-defined function $\theta_q(s, l_q)$, where l_q is the queried location. The following is an example of such a function:

$$\theta_q(s, l_q) = \begin{cases} (1 - \gamma_s)(1 - \frac{|l_s - l_q|}{d_{max}})\tau_s & \text{if } |l_s - l_q| \leq d_{max} \\ 0 & \text{otherwise,} \end{cases} \tag{3.4}$$

where γ_s is the inaccuracy of s measured in percentage of the value range of the sensor, $0 \leq \tau_s \leq 1$ is the trustworthiness of s, l_s is the current location of s, and d_{max} is the maximum distance in which the sensors can be considered to provide data. Hereafter, we assume the same function for all queries and we only use θ_s when l_q is implied by the context.

In the case of multiple-sensor point queries, the querying application is requested to provide a more general valuation function $v_q(S)$, that takes as input a set of sensors and determines their value to the query.

3.2.2.2 Spatial Aggregate Queries

When issuing spatial aggregate queries, applications are interested in an aggregate value of the measurements (e.g., average, min, and max) over a region. Users assign a budget B_q to each query q and spend it based on their valuation of the quality of the result. The quality of an aggregate query answer depends on the qualities of the sensor readings used for providing the answer as well as the coverage of these readings. The application provides, along with the query q, a function $v_q(S_q)$ that evaluates the quality of the result. S_q denotes the set of selected sensors for answering query q. The following is an example of such a function:

$$v_q(S_q) = B_q \mathcal{G}_q(S_q) \frac{\sum_{s \in S_q} \theta_s}{|S_q|}, \tag{3.5}$$

where $\mathcal{G}_q$ is a function that calculates the coverage of the selected sensors. A simple coverage function can calculate the fraction of the area covered by the sensors, while a more general function might also take into account the dispersion or the importance of the locations that are covered by the selected sensors.

3.2.2.3 Queries over Trajectories

When a user issues a query over a trajectory, she would like to know the (aggregate) value of a phenomenon over that trajectory. For instance, a user might be interested in knowing the current maximum value of CO_2 on the way from her house to her work. This type of query can be treated as a special case of spatial aggregate query in which instead of providing a region of interest, a trajectory is specified.

3.2.3 Continuous Queries

Continuous queries are queries that are continuously executed for a certain time period or until they are removed by the users. In general, two categories of continuous queries can be distinguished: 1) *monitoring queries* that ask for continuously monitoring a phenomenon at a certain location or area, and 2) *event detection queries* that ask for monitoring a location or region for detecting the occurrence of an event. In the following example queries, Q1 and Q2 are monitoring queries and Q3 and Q4 are event detection queries.

Q1: Monitor CO_2 level at location l in the period $[t_1, t_2]$.

Q2: Monitor CO_2 level in region r in the period $[t_1, t_2]$.

Q3: Notify me when $CO_2 > x$ with $confidence > \alpha$ at location l in the period $[t_1, t_2]$.

Q4: Notify me when $avg(CO_2) > x$ with $confidence > \alpha$ in region R in the period $[t_1, t_2]$.

For queries similar to Q1, which are referred to as *location monitoring queries*, applications are requested to provide the desired sampling times $\mathcal{T}$, as well as a valuation function $v_q(\mathcal{T}')$, which returns the value of sampled times $\mathcal{T}'$. Since the locations of sensors, rather unpredictably, change over time, satisfying all the desired sampling times cannot be guaranteed. On the other hand, it is likely that a sensor moves close to a queried location at time $t' \notin \mathcal{T}$. Taking a measurement at these time instances, especially when the sensor can be shared with other queries, can increase the utility of the query at hand. In Section 3.3.3 we propose an approach to answering location monitoring queries with the objective of increasing the utility of the queries.

In the case of queries similar to Q2, which we refer to them as *region monitoring queries*, we rely on the querying applications to provide their desired sampling points (i.e., sampling locations and times), as well as a valuation function $v_q(.)$, which calculates the value of the measurements (taken at any sampling points). As for location monitoring queries, it might not be possible to satisfy all the desired sampling points. Also, there are opportunities to use other sampling points considering the sharing possibilities with other queries. In Section 3.3.3 we introduce an approach for answering region monitoring queries considering the opportunistic nature of participatory sensing.

In this work, we do not specifically deal with event detection queries. However, we believe that data acquisition for this type of continuous queries is very similar to data acquisition for monitoring queries. The main difference is that redundant sampling might be needed to ensure the confidence requested by the queries.

3.2.3.1 Example Valuation Function for Region Monitoring Queries

One common approach for finding the valuation of a set of sensors for an application is to use the notion of *expected reduction in variance* [29, 69]. In this approach the phenomenon is modeled as a Gaussian process. Let $\mathcal{V}$ be the set of locations at which a measurement can be taken, i.e., there exists at least one sensor at each of these locations. The state of the phenomenon can be modeled using a set of random variables $\mathcal{X}_{\mathcal{V}}$.

Assume, for the moment, that the goal is to select a subset $\mathcal{A} \subseteq \mathcal{V}$ of the locations to maximize the sensing quality $F(\mathcal{A})$ while the budget constraint is satisfied. The value of the phenomenon at the unobserved locations are then predicted based on the process model given the observed locations. The expected reduction in variance at the unobserved locations can be used to measure the quality of sensing if the set $\mathcal{A}$ of locations are selected to take measurements from. This quantity is given by:

$$F(\mathcal{A}) = Var(\mathcal{X}_{\mathcal{V}}) - \int P(\mathbf{x}_{\mathcal{A}}) Var(\mathcal{X}_{\mathcal{V}} | \mathcal{X}_{\mathcal{A}} = \mathbf{x}_{\mathcal{A}}) d\mathbf{x}_{\mathcal{A}}, \qquad (3.6)$$

where $\mathbf{x}_{\mathcal{A}}$ is the measurements observed at locations $\mathcal{A}$. Consequently, the following valuation function can be used for region monitoring queries:

$$v_q(S) = B_q \cdot F(S) \cdot \frac{\sum_{s \in S} \theta_s}{|S|}, \qquad (3.7)$$

where S is the set of sensors (and their locations). Notice that in the above modeling, the assumption is that the phenomenon is a spatial process. In order to expand the approach for spatio-temporal phenomena, one needs to add a time dimension to the random variables.

3.2.4 Costs

Sensor owners participate in the system as long as the resource consumption on their devices as well as their location privacy loss are compensated. In this regard, each sensor asks for a certain price in return for providing a measurement to the aggregator. Therefore, the cost of obtaining a measurement from sensor s which is located at l_s, consists of two components as demonstrated in the following equation:

$$c_s(\mathcal{E}_s, H_s, l_s) = c_s^e(\mathcal{E}_s) + c_s^p(p_s(H_s, l_s)), \qquad (3.8)$$

where $\mathcal{E}_s$ is the remaining energy, and H_s is the history of revealed locations of s. c_s^e is a function that gives the energy cost of taking a measurement and transmitting it to the aggregator, and c_s^p is a function that calculates the cost of the sensor's privacy loss due to revealing its location. The privacy loss is computed by the function p_s. We do not impose any restrictions on the form of these two functions.

3.3 Our Data Acquisition Approach

In this section we describe our approach to the problem of utility-driven data acquisition for a mixture of queries of different types. We first introduce data acquisition for each query type. Data acquisition for the query mix, which is based on the data acquisition algorithms for individual query types, is explained in the end of this section.

3.3.1 Single-Sensor Point Queries

We present two algorithms for answering single-sensor point queries. The first one finds the optimal solution but does not scale to large problem instances. The second one is an efficient heuristic approximation.

Symbol	Semantic
$\theta_{q,s}$	quality of readings from sensor s for query q (in $[0,1]$)
τ_s	trustworthiness of sensor s (in $[0,1]$)
γ_s	inaccuracy of sensor s (in $[0,1]$)
B_q	budget for query q
v_q	utility function for query q
$\mathcal{E}_s$	remaining energy for sensor s
H_s	history of revealed locations for sensor s
l_s	location of sensor s
l_q	location queried by query q
$c_s/c_s^e/c_s^p$	total/energy/privacy cost for sensor s
T/t	total considered time period/specific time slot
$\mathcal{Q}/Q/Q^l$	all queries in $T/$ in the current time slot/ at location l
$\mathcal{K}/K$	set of all possible allocation schemes/specific allocation scheme
$\mathcal{M}/M$	set of all possible allocation schemes/specific allocation scheme in one time slot
$\mathcal{T}/\mathcal{T}'$	set of desired sampling times/set of sampled times for a location monitoring query
$\pi_{q,s}$	the payment of query q to sensor s

Table 3.1: Summary of introduced symbols

3.3.1.1 Optimal Scheduling

When there exist only point queries in the current time slot, we can express the optimized sensor allocation problem as a Binary Integer Linear Program (BILP). Assume n sensors are available and L locations are queried. For each queried location l, by m_l queries, we define a binary variable $Y_i^l \in \{0,1\}$ for each $i = 1, \dots, n$, which states whether or not sensor i is assigned to location l. For each sensor i, let $X_i \in \{0,1\}$ denote whether or not sensor i is assigned to any location. We denote by c_i the cost of sensor i. The following BILP solves the problem of optimally assigning sensors to answer single-sensor point queries:

$$\max \sum_{l=1}^{L} \sum_{i=1}^{n} v_l'(s_i) Y_i^l - \sum_{i=1}^{n} c_i X_i,$$
$$\text{s.t.} \tag{3.9}$$
$$Y_i^l \leq X_i \quad \forall i, l, \quad \text{and} \quad \sum_{i=1}^{n} Y_i^l \leq 1 \quad \forall l.$$

In the above formula $v_l'(s_i)$ is defined as:

$$v_l'(s_i) = \begin{cases} v_l(s_i) & \text{if } v_l(s_i) > 0 \\ -1 & \text{otherwise,} \end{cases} \tag{3.10}$$

where $v_l(s_i) = \sum_{q \in Q^l} v_q(s_i)$ in which Q^l is the set of queries at location l.

We split the sensor cost among queries proportionally to the value it yields to each query (*proportionate cost allocation.*) In other words, if $Y_s^l = 1$, then the user who has

issued the query $q^* \in Q^l$ has to pay according to the following:

$$\pi_{q^*,s} = \frac{v_{q^*}(s) \cdot c_s}{\sum_{a=1}^{L} v'_a(s) Y_s^a}. \tag{3.11}$$

This cost sharing scheme ensures that each query receives a positive net benefit because a sensor is selected only if the total valuation it yields is greater than its cost. It follows that $\pi_{q,s} < v_q(s)$ for each q that is answered by sensor s.

3.3.1.2 Heuristic Scheduling

Instances of the optimization problem (3.9) can be solved optimally by an ILP solver as long as the input size is not very large. When the input size is very large, we can resort to an approximation algorithm. We define the utility function $u : 2^S \to \mathbb{R}$ as the following:

$$u(S') = \sum_{l \in L} \max_{s \in S'} v_l(s) - \sum_{s \in S'} c_s, \tag{3.12}$$

where $S' \subseteq S$. Then the optimal sensor allocation problem reduces to finding $S^* \subseteq S$ such that $S^* = \arg\max_{S' \subseteq S} u(S')$. Once the optimal set of sensors is determined, each sensor is assigned to a query location for which it yields the best valuation compared to other sensors. It can be shown that $u(.)$ is a (non-monotone) submodular function.

A $\frac{1}{3}$-approximation algorithm for non-monotone submodular functions, referred to as *Local Search* algorithm, is proposed in [40]. This algorithm, presented in Algorithm 3.1, works as follows. It starts by adding the sensor which maximizes the utility function to the set of selected sensors W. Then it iteratively adds to W those sensors that increase the utility more than a certain threshold. In the next step, it removes from W the sensors that have become obsolete and then goes to the previous step. These steps are repeated until no obsolete sensors are found. If $u(W) \geq u(S \backslash W)$, then the set W is returned, otherwise $S \backslash W$ is returned as the set of selected sensors. This algorithm requires at most $O(n^3 \log n)$ calls to the utility function, where n is the number of available sensors. It is worthwhile to mention that a randomized local search algorithm is also proposed in [40], which achieves a $\frac{2}{5}$-approximation of the optimal solution. More recently, a deterministic $\frac{1}{3}$-approximation algorithm (see Section 4.3.1) and a randomized $\frac{1}{2}$-approximation algorithm have been proposed in [17]. However, in our experiments we only use the Local Search algorithm. The bounds for these approximation algorithms can be guaranteed only when the function is not negative. In order to make the above utility function non-negative we add the sum of all sensor costs to the value of the function.

3.3.2 Multiple-Sensor One-shot Queries

There exist queries which ask for measurements from more than one sensor. These queries include, but not limited to, *spatial aggregate queries*, *queries over trajectories*, and *multiple-sensor point queries*. At each time slot several of these queries can arrive to the aggregator. Many of them require data about the same phenomenon over different

---

Algorithm 3.1: Local Search

Data: Set S of available sensors, and utility function $u(.)$

Result: The set of selected sensors

1 $n \leftarrow |S|$

2 $s^* \leftarrow \arg\max_{s \in S} u(\{s\})$

3 $W \leftarrow \{s^*\}$

4 $done \leftarrow 0$

5 **while** $done = 0$ **do**

6   **while** $\exists s \in S \backslash W$ $s.t.$ $u(W \cup \{s\}) > (1 + \frac{\epsilon}{n^2})u(W)$ **do**

7 $W \leftarrow W \cup \{s\}$

8   **if** $\exists s \in W$ $s.t.$ $u(W \backslash \{s\}) > (1 + \frac{\epsilon}{n^2})u(W)$ **then**

9 $W \leftarrow W \backslash \{s\}$

10   **else**

11 $done \leftarrow 1$

12 **if** $u(W) > u(S \backslash W)$ **then**

13   **return** W

14 **else**

15   **return** $S \backslash W$

(potentially overlapping) regions. In order to maximize the overall utility, the aggregator must exploit as much as possible the common data requirements among these queries and select the best set of sensors that provide the required data. The problem of finding the optimal set of sensors is a combinatorial problem, since we have to enumerate all possible sensor assignments to queries and select the one that maximizes the overall net benefit. Therefore, we propose a greedy approach, presented in Algorithm 3.2, that iteratively selects sensors that maximize the partial overall utility.

The objective is to maximize the following utility function:

$$u(S') = \sum_{q \in Q} v_q(S') - \sum_{s \in S'} c_s, \tag{3.13}$$

where S' is the set of selected sensors, and Q is the set of queries. When all v_q's are submodular, it can be shown that $u(.)$ is also a submodular (non-monotone) function. While the algorithms proposed in [40] have proven performance guarantees for submodular functions, it is shown that the greedy algorithm can perform arbitrarily badly compared to the optimal solution. However, because the valuation functions are taken as black boxes, we use Algorithm 3.2 unless we have knowledge about submodularity of the valuation functions. In this case, we can adapt the aforementioned algorithms for non-monotone submodular function. The reason behind using Algorithm 3.2 instead of always utilizing the adapted approximate algorithms in [40] is that when the utility functions are not submodular, experimentally the former performs better in terms of total utility and it's also faster. It is worth mentioning that, for example, function (3.5) is not submodular, even though it is known that the coverage function is submodular. In-

volving sensor quality in evaluation of a set of sensors destroys the submodularity of the function. Only when all the sensors have the same quality, function (3.5) is submodular.

Algorithm 3.2: Greedy Sensor Selection

Data: Set Q of queries, S of available sensors, and quality valuation function v_q of each query q.

Result: $S \backslash \tilde{S}$ is the set of selected sensors.

1   $\tilde{S} \leftarrow S$

2   $\forall q \in Q, S_q \leftarrow \emptyset$

3   **while** $\tilde{S} \neq \emptyset$ **do**

4     $\forall s \in \tilde{S}, Q_s \leftarrow \emptyset$

5   **foreach** $s \in \tilde{S}$ *and* $q \in Q_{l_s}$ **do**

6       $\Delta v_{q,s} \leftarrow v_q(S_q \cup \{s\}) - v_q(S_q)$

7   **if** $\Delta v_{q,s} > 0$ **then** $Q_s \leftarrow Q_s \cup \{q\}$

8     $s^* \leftarrow \arg\max_{s \in \tilde{S}} \sum_{q \in Q_s} \Delta v_{q,s} - c_s$

9   **if** $\sum_{q \in Q_{s^*}} \Delta v_{q,s^*} - c_{s^*} > 0$ **then**

10       $\forall q \in Q_{s^*}, \ \ S_q \leftarrow S_q \cup \{s^*\}; \ \ \pi_{q,s^*} \leftarrow \frac{\Delta v_{q,s^*} \cdot c_{s^*}}{\sum_{q' \in Q_{s^*}} \Delta v_{q',s^*}}$

11       $\tilde{S} \leftarrow \tilde{S} \backslash \{s^*\}$

12   **else** Leave the while loop

Theorem 3.3.1. Let $S' = S \backslash \tilde{S}$ denote the set of selected sensors after Algorithm 3.2 terminates. Let $S_q = S_q^{(m)} = \{s_1, s_2, ..., s_m\}$ be the set of selected sensors for query q, where m is the number of these sensors. We have the following properties:

1. $\sum_{s \in S_q} \Delta v_{q,s} = v_q(S_q), \ \ \forall q \in Q.$

2. If $S' \neq \emptyset$, then $\sum_{q \in Q} v_q(S_q) > \sum_{s \in S'} c_s$, that is, the total utility is positive.

3. $v_q(S_q) > \sum_{s \in S_q} \pi_{q,s}, \ \ \forall q \in Q$, that is, the individual utility is not negative.

4. The algorithm requires $O(|Q||S|^2)$ calls to the valuation functions.

Proof. The first property is proved using the definition of $\Delta v_{q,s}$, the partial valuation of a sensor s for a query q:

$$\sum_{s \in S_q} \Delta v_{q,s} = \sum_{i=1}^{m} \Delta v_{q,s_i} = \sum_{i=1}^{m} \left(v_q(S_q^{(i-1)} \cup \{s_i\}) - v_q(S_q^{(i-1)}) \right)$$
$$= \sum_{i=1}^{m} \left(v_q(S_q^{(i)}) - v_q(S_q^{(i-1)}) \right) = v_q(S_q^{(m)}) - v_q(S_q^{(0)})$$
$$= v_q(S_q^{(m)}) = v_q(S_q).$$

The second property can be easily proved by using property 1 and the fact that the algorithm ensures $\sum_{q \in Q} \Delta v_{q,s} - c_s > 0$ for each selected sensor s. The proof of the third property is straightforward in the same way as for property 2 and by using the definition of proportionate cost allocation. The algorithm goes through the sensors in $\tilde{S}$ in every

iteration (at most $|S|$ iterations) and this continues until $\tilde{S}$ becomes empty. In each iteration all queries are considered. Therefore, the time complexity of Algorithm 3.2 is $O(|Q||S|^2)$. $\qquad\square$

3.3.3 Continuous Queries

We propose Algorithm 3.3 for providing the required data for a set of location monitoring queries. Each query q continuously needs the value of a phenomenon at location $q.l$ in the time period $q.t_1$ to $q.t_2$. The desired sampling times of query q is denoted by $q.\mathcal{T}$. The main objective of the algorithm is to obtain a measurement for each $t \in q.\mathcal{T}$. However, because of the uncertainty in succeeding to satisfy all the desired sampling times, we also follow an opportunistic approach to obtain measurements at all $t' \notin q.\mathcal{T}$.

At each time slot t, for each available location monitoring query, $CreatePointQuery()$ is called to create a point query at the queried location. After execution of the created point queries, procedure $ApplyResults()$ is invoked to apply the results for each query. Consider one location monitoring query q. If $t \in q.\mathcal{T}$, or if sampling at the last sampling time has been failed, or if t is greater than the final requested sampling time, a point query is created. The maximum value for the valuation function of the point query is denoted by Δv, which is the valuation of taking a sample at time t. When none of these conditions hold, the current extra budget is calculated and a fraction, denoted by parameter α, of this extra budget is used for a point query. The reason behind using only a fraction of the extra budget is to be able to keep some extra budget for uncertain future samples. A natural way for specifying α is to start with a small value and increase it (or possibly decrease it) as we learn the difference between the utility obtained compared to the expected utility and how much utility is expected to achieve in future. In this algorithm, $\mathcal{T}.first$ returns the first sampling time in $\mathcal{T}$, and $\mathcal{T}.next(t)$ returns the first sampling time which is greater than t. Note that although omitted in the algorithm, v_q considers the quality of the collected sensor readings or the expected quality of a sensor reading before the actual sensor selection at the current time.

Algorithm 3.4 is used for answering a set of region monitoring queries. The algorithm uses two main functions: $CreatePointQueries()$ and $ApplyResults()$. The first is called

---

Function CreatePointQuery(t, q)

Data: t is the current time and q is the query
Result: A point query for query q at time t

1   **if** $t = q.t_1$ **then**
2      $q.\mathcal{T}' \leftarrow \emptyset$; $q.\widehat{C} \leftarrow 0$
3      $q.lst \leftarrow -\infty$; $q.nst \leftarrow q.\mathcal{T}.first$
4   $\Delta v_t \leftarrow v_q(q.\mathcal{T}' \cup \{t\}) - v_q(q.\mathcal{T}')$
5   **if** $t \in q.\mathcal{T}$ *OR* $q.nst = \infty$ *OR* $q.lst < q.\mathcal{T}'.last$ **then** $\Delta v \leftarrow \Delta v_t$
6   **else** $\Delta v \leftarrow min\{\alpha(v_q(q.\mathcal{T}') - q.\widehat{C}), \Delta v_t\}$
7   **return** *A point query q_l with the valuation function with the maximum value of* Δv.

Procedure ApplyResults(t, q, π)

Data: t is the current time, q is the query, and π is the amount that q must pay.

1 **if** $\pi \geq 0$ **then**
2 $\quad$ $q.\mathcal{T}' \leftarrow q.\mathcal{T}' \cup \{t\}$
3 $\quad$ $q.\widehat{C} \leftarrow q.\widehat{C} + \pi$
4 $\quad$ **if** $t = q.nst$ **then** $\quad q.lst \leftarrow t; q.nst \leftarrow q.\mathcal{T}.next(t)$
5 **else if** $t \in \mathcal{T}$ **then** $\quad q.lst \leftarrow t; q.nst \leftarrow q.\mathcal{T}.next(t)$

for generating the required point queries, and the second is called for applying the results after execution of point queries. Consider a single region monitoring query q with region r_q. At each time t, a query-specific function f_q is consulted for obtaining the desired sampling locations based on the current locations and costs of sensors in r_q and the remaining budget. For each sampling location, a point query is created with the valuation function equal to the valuation improvement by the sensor at that location. The generated point queries, Q_t, are then executed along with all other point queries, e.g., using one of the algorithms introduced in Section 3.3.1.

After execution of point queries we can make use of the sensors that are selected for other queries if they fall into r_q. The maximum total cost contribution from query q for these sensors is $\alpha(C_t - \widehat{C}_t)$, where C_t is the expected cost to be spent, and $\widehat{C}_t$ is the actual cost spent in time t. The control parameter α is used for determining how much extra budget to keep for the next time slots. The actual cost contribution depends on the sensors' costs and their valuation improvement for the query.

Sensor data sharing is possible when the query regions overlap. This potential data sharing can be incorporated in Algorithm 3.4 by providing the input set $SC_{r,t}$ to the function f_q as a set containing weighted costs of sensors. For example, when some sensors are already selected for other queries, a weight of zero can be assigned to their costs in $SC_{r,t}$. Also, a heuristic approach for increasing the selection chance of a sensor which can be shared by k region monitoring queries, is to reduce its cost by a factor of $w(k)$, where w is a function that returns a real value between 0 and 1.

Because of the query budget constraints, a mechanism is needed to decide which sensors to take measurements from and when. In the context of sensor networks, this

Algorithm 3.3: Sensor Selection for Location Monitoring Queries at time t

Data: Set Q of location monitoring queries, and quality valuation function v_q of each query q.

1 $Q_p \leftarrow \emptyset$
2 **foreach** $q \in Q$ **do**
3 $\quad$ $Q_p \leftarrow Q_p \cup$ CreatePointQuery(t, q);
4 Select sensors for point queries in Q_p and for each point query calculate its payment $\pi_{q,t}$. If the point query is not satisfied, set $\pi_{q,t} \leftarrow -\infty$.
5 **foreach** $q \in Q$ **do**
6 $\quad$ ApplyResults($t, q, \pi_{q,t}$)

Function CreatePointQueries($t, q, S_{r,t}, SC_{r,t}$)

> **Data**: t is the current time, q is the query, $S_{r,t}$ is the set of sensors in region $q.r$ at time t, and $SC_{r,t}$ is their corresponding locations and costs
>
> **Result**: A set of point queries, the expected budget for the queries, and the set of sensors which are supposed to answer the point queries

1 **if** $t = q.t_1$ **then**
2 $\quad \lfloor \ q.S \leftarrow \emptyset, \ q.\widehat{C} \leftarrow 0$
3 $C_t \leftarrow 0, \ Q_t \leftarrow \emptyset$
4 $S_t \leftarrow f_q(S_{r,t}, SC_{r,t}, q.B - q.\widehat{C})$
5 **foreach** $s \in S_t$ **do**
6 $\quad$ Create a point query q_s with the valuation function $v_{pq} = v_q(S_t) - v_q(S_t \setminus \{s\})$.
7 $\quad \lfloor \ Q_t \leftarrow Q_t \cup q_s; \ C_t \leftarrow C_t + c_s$
8 **return** $\{Q_t, \ C_t, \ S_t\}$

Procedure ApplyResults($q, Q_t, C_t, S_t, \pi, A_{r,t}$)

> **Data**: q, Q_t, C_t, S_t are as for `CreatePointQueries`, π is the amount that q pays for the satisfied point queries, and $A_{r,t}$ is the set of sensors in region $q.r$ selected for other queries

1 **foreach** $q_s \in Q_t$, *if q_s is not satisfied* **do**
2 $\quad \lfloor \ S_t \leftarrow S_t \setminus \{s\}$
3 $\widehat{C}_t \leftarrow \pi$
4 Contribute to the costs of sensors in $A_{r,t} \setminus S_t$ by the maximum amount of $\alpha(C_t - \widehat{C}_t)$ and update $\widehat{C}_t$ accordingly.
5 $q.S \leftarrow q.S \cup (S_t \cup A_{r,t}); \ q.\widehat{C} \leftarrow q.\widehat{C} + \widehat{C}_t$

---

problem is referred to as *sensor selection problem*. To be able to support a wide range of applications, the queries are requested to provide a method for specifying the desired set of sampling points at each time slot (f_q in Algorithm 3.4.) In participatory sensing with uncontrolled mobility, applications are faced with an obstacle for finding out all their desired sampling points in advance: only at the current time we know which sensors are located in the queried region. As a workaround, instead of finding upfront all the desired sampling points, at each time slot we can select the best sampling locations based on the available sensors in the queried region.

We propose Algorithm 3.5 as an example approach for finding the set of best sensors to query for the current time t_c. The sensors in the queried region along with their costs and locations are provided as input to the algorithm. We assume that the current location of sensors will not change in the future. Even with this simplifying assumption, the problem is NP-complete. The proposed solution is hence a greedy approach. Notice that even though the algorithm selects (sensor) locations for different time instances, we are only interested in the locations for t_c. The multiplication of the sensing quality improvement by the fraction of the remaining time over the duration of the query is an attempt to increase the chance of selecting sensors for the current time.

Algorithm 3.4: Sensor Selection for Region Monitoring Queries at Time t

---
Data: Set Q of region monitoring queries, and quality valuation function v_q of each query q.

1 $Q_p \leftarrow \emptyset$
2 **forall the** $q \in Q$ **do**
3 $\quad$ Compute $S_{r,t}$ and $SC_{r,t}$
4 $\quad$ $X[q] \leftarrow \texttt{CreatePointQueries}(t, q, S_{r,t}, SC_{r,t})$
5 $\quad$ $Q_p \leftarrow Q_p \cup X[q].Q_t$
6 Select sensors for point queries in Q_p.
7 **foreach** $q \in Q$ **do**
8 $\quad$ $\pi \leftarrow$ the payment of q for the satisfied point queries in $X[q].Q_t$
9 $\quad$ $A_{r,t} \leftarrow$ selected sensors in region $q.r$ at time t for other queries
10 $\quad$ $\texttt{ApplyResults}(q, X[q].Q_t, X[q].C_t, X[q].S_t, \pi_q, A_{r,t})$

Algorithm 3.5: Sampling point selection for a region monitoring query at time t_c.

Data: Set S of available sensors in queried region r_q of query q, the budget B, and function F that quantifies the value of a set of sensors.

Result: S_{t_c} is the set of locations to query at current time t_c.

1 $C \leftarrow 0$
2 $S_t \leftarrow \emptyset$ for all $t = t_c, \ldots, q.t_2$
3 **while** $C < B$ **do**
4 $\quad$ **foreach** $s \in S$ **do**
5 $\quad\quad$ **foreach** $t = t_c$ *to* $q.t_2$ **do**
6 $\quad\quad\quad$ **if** $s \notin S_t$ **then**
8 $\quad\quad\quad\quad$ $\delta_{s,t} \leftarrow (F(S_t \cup \{s\}) - F(S_t)) \, \theta_s \frac{q.t_2 - t}{q.t_2 - q.t_1}$
9 $\quad$ $(s^*, t^*) \leftarrow \arg\max_{s,t} \delta_{s,t}$
10 $\quad$ $S_{t^*} \leftarrow S_{t^*} \cup \{s^*\}$
11 $\quad$ $C \leftarrow C + c_{s^*}$

3.3.4 Query Mix

When the aggregator receives queries of different types, it has the possibility of sharing the sensors among them and hence increasing the total utility. Indeed, since individually finding an optimal set of sensors for multiple point or aggregate queries is NP-Complete, finding the optimal set of sensors for the combination of queries is also NP-Complete. We therefore propose Algorithm 3.6 for selecting sensors considering the commonalities between the queries at hand.

This algorithm consists of four stages. In the first stage, the required point queries are generated for available location monitoring and region monitoring queries. For doing so, the functions *CreatePointQuery* used in Algorithm 3.3 and *CreatePointQueries* used in Algorithm 3.4 are called. In the second step, all the queries are jointly provided to Algorithm 3.2 as the input. This greedy algorithm selects the sensors with the objective of increasing the total utility and computes the amount that each query will be charged

for using the data from the assigned sensors. In the next stage, the results of the point queries generated for continuous queries are applied using the procedures *ApplyResults* used in Algorithms 3.3 and 3.4. The cost contribution from region monitoring queries for the extra sensors that they can use necessitates the adjustment of the payments for other queries sharing the same sensors. In the last stage, selected sensors are asked to send their measurements, which are then sent to the higher level query processor. Finally, the users are charged the amount that is calculated in the previous stage and each selected sensor is paid its announced price.

Algorithm 3.6: Data Acquisition for Query Mix

Data: Set $Q_{agg}, Q_p, Q_{lm}, Q_{rm}$ of aggregate, point, location monitoring, and region monitoring queries, set S of available sensors, and quality valuation function v_q of each query q.

   ▷ **Point query creation for continuous queries**

1 [Function `CreatePointQuery`] Create required point queries for location monitoring queries Q_{lm}. Let Q_p^{lm} denote the generated point queries.

2 [Function `CreatePointQueries`] Create required point queries for region monitoring queries Q_{rm}. Let Q_p^{rm} denote the generated point queries.

   ▷ **Sensor selection**

3 [Algorithm 3.2] Input all the queries $Q_{agg} \cup Q_p \cup Q_p^{lm} \cup Q_p^{rm}$ to Algorithm 3.2 for sensor selection.

4 [Algorithm 3.3 and 3.4] Run Algorithms 3.3 and 3.4 for applying the results of the corresponding point queries.

   ▷ **Payment adjustment**

5 Adjust the payments to be asked from the queries based on the potential cost contribution resulting from Step 4.

   ▷ **Data acquisition and accounting**

6 Ask the selected sensors to provide their measurements.

7 Provide the data to the query processor. Charge the users whose queries have been satisfied and pay the cost of selected sensors.

3.4 Experimental Evaluation

In order to prove the effectiveness of our utility-driven data acquisition framework, we have conducted a thorough simulation study using real and synthetic mobility datasets. In the following we first introduce these datasets and then for each query type presents the experiments and their results.

3.4.1 Setup

We consider a simulation period of 50 time slots in all the experiments. At each time slot new queries are generated and then executed jointly with the existing continuous queries, if any. The inaccuracy of each sensor is chosen randomly from the interval $[0, 0.2]$. We refer to the maximum number of readings that a sensor can provide as the *lifetime* of the sensor. When the number of measurement taken by a sensor reaches

its lifetime, it cannot be used anymore in the subsequent time slots. Unless otherwise stated, the lifetime is equal to the simulation period. We use two simple energy cost models: A *fixed cost model* defined by $c_s^e(\mathcal{E}_s) = C_s$, and a *linear cost model* defined by $c_s^e(\mathcal{E}_s) = C_s(1 + \beta(1 - \mathcal{E}_s))$, where C_s is a fix price, and β is the cost increment factor.

We assume the aggregator is a trusted entity and therefore, the sensors always report their true locations to the aggregator. However, the other consumers of the data are not trusted. The privacy computation model employed in the simulations works as follows: each sensor keeps a history of the times when it has reported a measurement to the aggregator. The size of the history is called the *privacy window* and is denoted by w. The privacy loss is the weighted average of the time distances between the times when a data is reported and the current time t:

$$p_s(H_s, l_s) = \frac{w + \sum_{t' \in H_s} (w - (t - t'))}{\frac{w(w+1)}{2}}. \tag{3.14}$$

Function (3.14) puts more weight on the recent data reporting times. Therefore, by applying this function, the sensor device tries to avoid reporting measurements in consecutive time instances, hence hiding its trajectory. We consider 5 different privacy sensitivity levels (PSL) for the sensor devices, namely *Zero, Low, Moderate, High,* and *Very High*, which are, respectively, mapped to values $0, 0.25, 0.5, 0.75, 1$. The privacy cost function is defined as:

$$c_s^p(p_s(H_s, l_s)) = PSL_s * p_s(H_s, l_s) * C_s. \tag{3.15}$$

In all the experiments we set $C_s = 10$ and unless stated otherwise, we use the fixed cost model for energy and we set the privacy sensitivity level to *Zero*.

A trust value in the interval $[0, 1]$ is assigned to each sensor. A trust value of zero indicates that the sensor readings cannot be trusted at all, while the trust value of one implies that the sensor readings are fully trusted. Even though the trustworthiness of the sensors can change over the course of time, for simplicity, we assume that this parameter remains unchanged over the whole simulation period. Since the trust or reputation assessment of sensors is not the focus of this work, we assume that there is a trust assessment mechanism in place which assigns trustworthiness values to the sensors upon initialization. In the simulations, unless specified otherwise, the sensors are assumed to be fully trusted.

3.4.2 Datasets

We use two mobility datasets: RWM generated based on the *random waypoint model* [57], and RNC which is a real mobility dataset from Nokia campaign [1]. In RWM each sensor moves from its current location with a speed randomly selected between zero and a sensor-specific maximum speed. The direction of the movement is either up, down, left, or right, and is randomly selected. The movements are limited to a region of 80×80

---

[1]http://opensense.epfl.ch

grids. Upon initialization the maximum speed of each sensor is set randomly to 4 or 5, which are spread randomly in the region. Only a central subregion of 50×50 is considered by the aggregator as the working region (or the "hotspot"). That is, only the queries and sensors that are bounded by this subregion are considered, but sensors can enter and leave this subregion. The default number of sensors for the experiments using RWM is 200.

RNC is derived from a data collection campaign in Lausanne, Switzerland consisting of location information of 180 participants. The whole region of movement is griditized into grids of length 100 meters. Only a region of 237×300 grids is considered and the working subregion is set to be a subregion of size 100×100. Because of the high sparsity of this mobility data, we have shifted the movement times to have more users in the same day. We also added some dummy users with the mobility patterns of the existing users but with randomly selected starting location and time of the movement from the real trajectories. This resulted in having in total 635 sensors in the whole region and on average 120 sensors in the working subregion in each time slot.

In the simulations involving region monitoring queries, we use Intel Lab dataset [2]. The simulations are performed over a 20×15 region. The reason for using this data set is that in the experiments for region monitoring query we need to have real sensor readings in addition to mobility data. Since the sensors in the Intel Lab deployment are stationary, we assign the sensor readings to the grids in which they are located. Then we use a random waypoint model for generating mobility data for 30 imaginary sensors. The sensor reading which is assigned to a grid is reported as the data for the imaginary sensor that is located in that grid.

3.4.3 Single-Sensor Point Queries

We have implemented a baseline algorithm which in each time slot takes queries one by one and for each query selects the sensor with maximum utility. A sensor that is selected to answer a query at a certain location is also assigned to all other queries at that location. The cost of the selected sensors is set to zero for the remaining queries. This algorithm resembles execution on query arrival and data buffering for the duration of a time slot.

In each time slot 300 users submit point queries each with the location randomly picked over the working region. The valuation function (4.12) with $\theta_{min}^q = 0.2$ is used for all point queries. For finding the quality of each sensor reading, function (4.11) is used with $d_{max} = 5$ for the experiments on RWM and with $d_{max} = 10$ for the experiments on RNC.

Figure 3.3(a) shows the average utility achieved by different algorithms per time slot w.r.t. the query budget when RWM is used. It can be seen that the Local Search algorithm finds solutions close to the optimal ones. In this experiment, the query budget is the same for all the queries. Figure 3.3(b) shows the fraction of point queries that

---

[2]http://db.csail.mit.edu/labdata

are answered (satisfaction ratio) by different algorithms. Since the baseline algorithm does not efficiently benefit from sensor sharing among queries, it cannot answer any queries when the query budget is small (i.e., 7, 10). On the contrary, the optimal and Local Search algorithms can always answer more than 60% of the queries. When the query budget is big enough, the average utility and the satisfaction ratio achieved by the algorithms become very close since the queries can afford the cost of any sensor. As the budget increases, the satisfaction ratio converges to around 73%. This shows that regardless of the amount of budget, about 27% of the queries can never be answered because of the lack of sensors with acceptable quality in their vicinities. We recall that our goal is to maximize utility, not to maximize the satisfaction ratio nor the quality of results. This means that the optimal algorithm might not always achieve the best satisfaction ratio compared to the heuristic algorithms. In other words, achieving higher utility sometimes requires refusing answering queries for which a lower total utility can be achieved.

Figures 3.4(a) and 3.4(b) show the results when RNC is used. Similar patterns as for the experiment with RWM are observed. However, the average utilities and satisfaction ratios are smaller than their counterparts in Figures 3.3(a) and 3.3(b). Besides the difference in the mobility patterns, the reason is that the simulation area in the experiments with RNC is larger, hence the sensors are more sparsely distributed. Hereafter, we only present the results on RNC dataset.

In practice we cannot assume that all the queries have the same budget. Therefore, in the next experiment we chose the query budget uniformly at random in *budget mean* ±10. Figures 3.5(a) and 3.5(b) show that the results are very similar to when the fixed budget scheme is used. Therefore, in order to highlight more easily the efficiency of the algorithms, in all the next experiments we use the fixed query budget scheme.

Figures 3.6(a) and 3.5(b) illustrate that as the number of queries increases, the possibility of sharing sensors among more queries increases, which results in more utility and slightly higher satisfaction ratio. In the next experiment, we randomly pick the privacy sensitivity level of each sensor and we set the sensors use the linear energy cost function with β randomly chosen in [0, 4]. The results are depicted in Figures 3.7(a) and 3.7(b) for lifetime 50 and in Figures 3.7(c) and 3.7(d) for lifetime 25. The figures demonstrate that in general the utility and satisfaction ratio drop when the participants become privacy sensitive and use non-constant energy cost (compare to Figures 3.4(a) and 3.4(b).) The difference in the utilities when the lifetime is 50 and when it is 25 is very small, which implies that only a few sensors are worn out during the simulation. Due to their mobility, sensors might enter and leave the working region at any time, which prevents sensors to be exhaustively used.

3.4.4   Spatial Aggregate Queries

Since other types of multiple-sensor one-shot queries introduced in this chapter can be treated similarly to the spatial aggregate queries, we only consider this query type. We have implemented a baseline algorithm for answering multiple-sensor one-shot queries
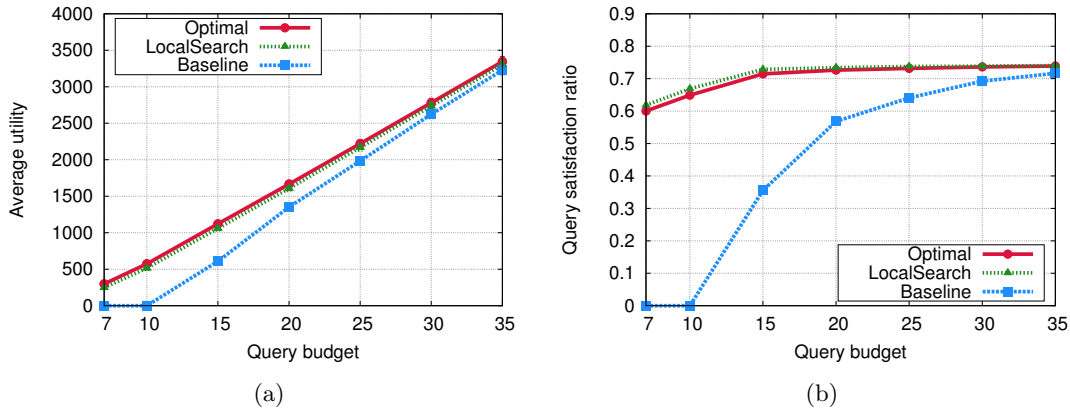
Figure 3.3: Single-sensor point queries, RWM dataset, a) average utility per time slot, b) satisfaction ratio.
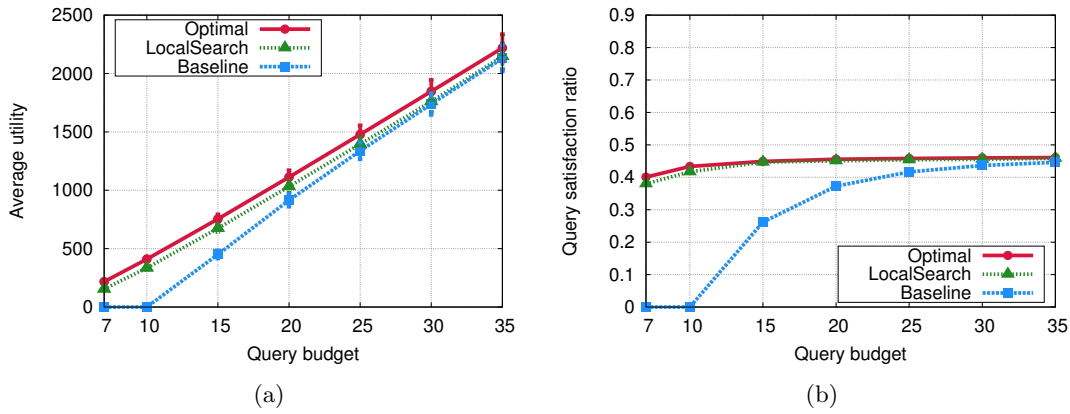
Figure 3.4: Single-sensor point queries, a) average utility per time slot, b) satisfaction ratio.

which resembles sequential execution of queries with data buffering. It takes the queries one by one and for each query selects the sensors that result in best utility. The cost of the selected sensors is set to zero for the subsequent queries in the time slot. The valuation function (3.5) is used for all queries. The sensing range of sensors is set to 10 units. In each time slot the number of aggregate queries is selected uniformly at random with the mean of 30 queries. The queried regions are generated randomly in the working region. The query budget for each query q is set to $\frac{A(r_q)}{1.5\pi r_s^2}b$, where $A(r_q)$ is the the size of the query area, r_s is the average coverage of the sensors (which is set to d_{max}), and b is the budget factor.

Figure 3.8(a) shows the average utility per time slot w.r.t. the budget factor. Algorithm 3.2 not only always significantly outperforms the baseline, but also can answer queries even when the budget is small. Figure 3.8(b) shows the average quality of results for the answered queries. The average quality of results for a query is the valuation of the set of selected sensors for that query divided by the maximum value of its valuation function.
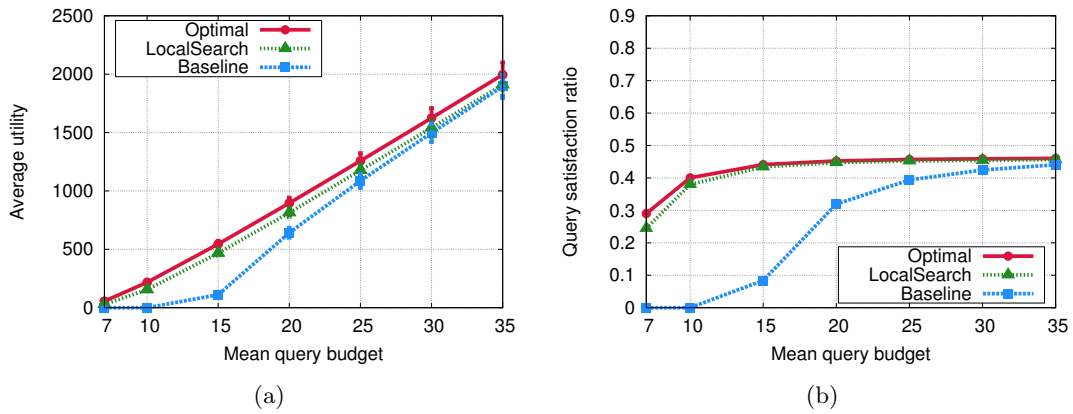
Figure 3.5: Uniformly distributed budget, a) average utility per time slot, b) satisfaction ratio of point queries.
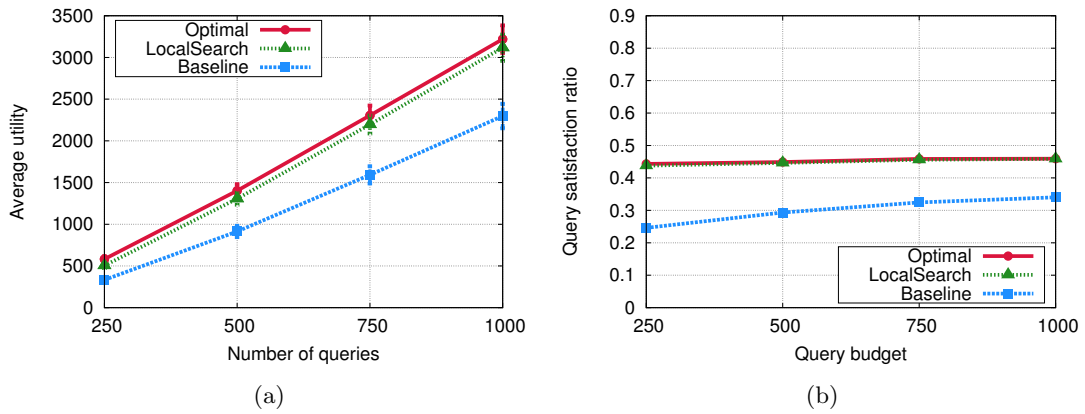
Figure 3.6: Varying the number of queries, with query budget fixed to 15. a) Average utility per time slot, b) satisfaction ratio of point queries.

3.4.5 Location Monitoring Queries

We use the technique proposed in [139] for determining the sampling times for a location monitoring query. This algorithm works on the historical data and selects the sampling times such that the residuals of the model based on the values at the sampling times and the model given all the historical data is minimized. The number of sampling times is assumed to be fixed and is given to the algorithm. This approach assumes that the data values for the current time interval are almost the same as the data values in the same time interval in the past. Even though this is a weak assumption, it shall not affect the effectiveness of our data acquisition approach, which is designed to work with any sampling method and any valuation function. We use a dataset containing a trace of ozone measurements from a deployments in Zurich, Switzerland [3]. A linear regression

---

[3]http://www.opensense.ethz.ch
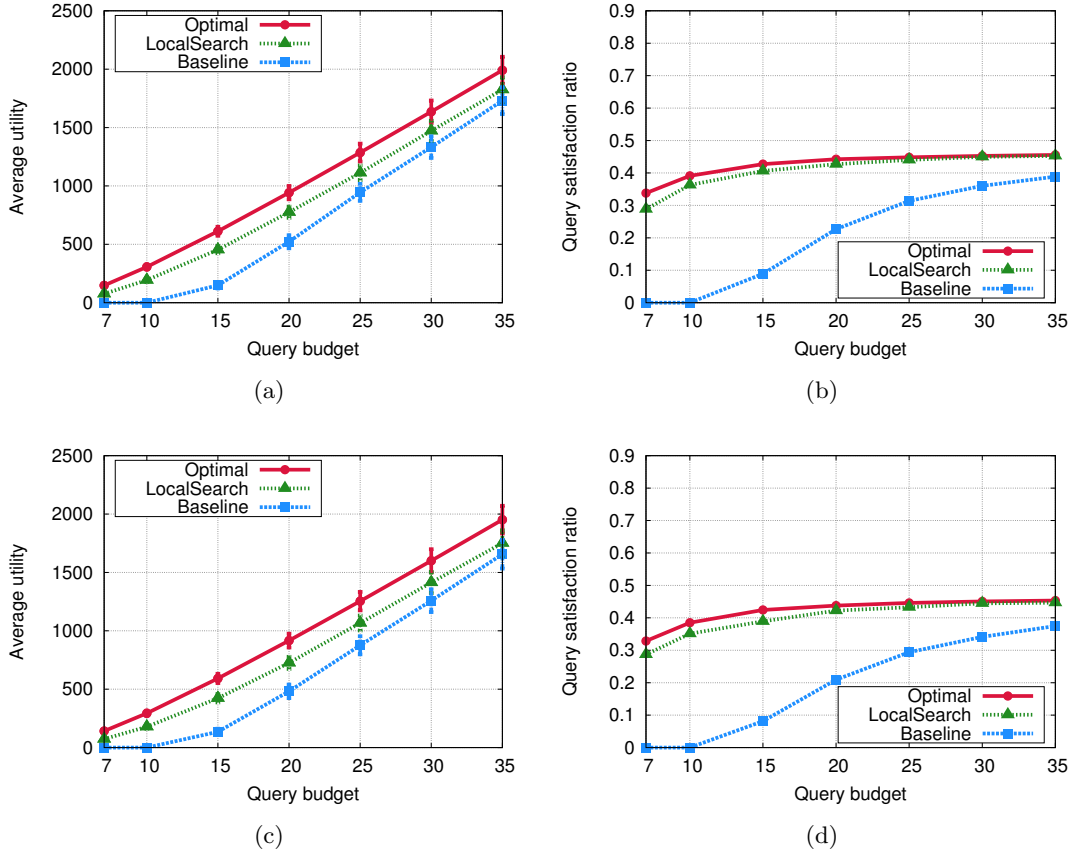
Figure 3.7: Random privacy sensitivity level and linear energy cost function, a) average utility per time slot - lifetime 50, b) satisfaction ratio of point queries - lifetime 50, c) average utility per time slot - lifetime 25, d) satisfaction ratio of point queries - lifetime 25.

model is used to model the data. We use the following valuation function:

$$v_q(\mathcal{T}', \Theta) = B_q G(\mathcal{T}') \frac{\sum_{\theta \in \Theta} \theta}{|\Theta|}, \tag{3.16}$$

with

$$G(\mathcal{T}') = \frac{\sum_{i=1}^{N} r_i^2 |\mathcal{T}}{\sum_{i=1}^{N} r_i^2 |\mathcal{T}'}, \tag{3.17}$$

where $\mathcal{T}$ is the desired sampling times, $\mathcal{T}'$ and Θ are the set of timestamps and qualities of the samples taken so far, B_q is the query budget, N is the number of historical data items, and $r_i|T$ is the difference between the actual value of the ith data item and the modeled value from the model generated using only data items with timestamps in T.

The setting is that at each time slot the number of existing queries and new queries is always less than 100. The location for each new query is randomly selected in the working subregion. The duration of each query is randomly chosen from $[5, 20]$ and the number of desired sampling times is set to one third of the query duration. The budget assigned to each query is equal to its duration times the budget factor. The parameter α is set to the constant value 0.5. Figure 3.9(a) shows the average utility per time slot w.r.t.
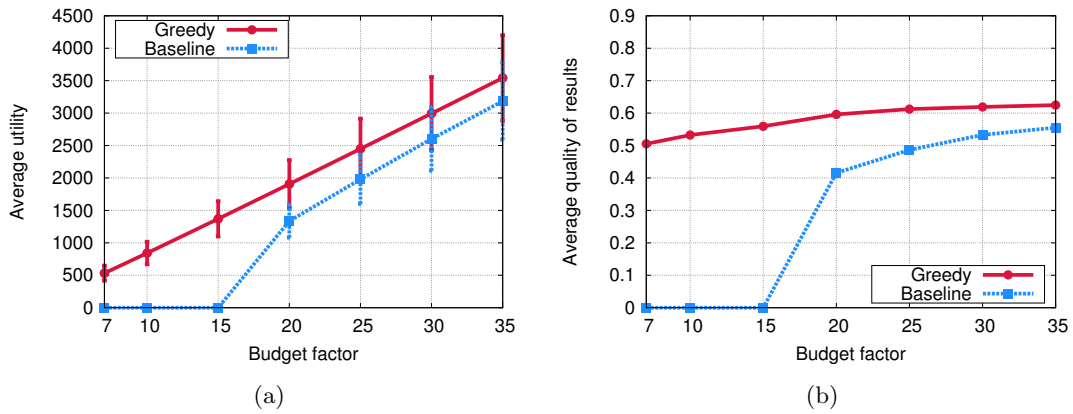
Figure 3.8: Aggregate queries, a) average utility per time slot, b) average quality of results.

the budget factor using Algorithm 3.3 compared to a baseline approach. *Alg3.3-O* and *Alg3.3-LS* state that, respectively, the optimal solution and the Local Search algorithm are used for answering point queries. In the baseline approach point queries are generated only at the desired sampling times and then the baseline approach introduced in Section 3.4.3 is used for answering the point queries. The average quality of results is shown in Figure 3.9(b). The relatively small values for the average utility and average quality of results stem from the lack of enough sensors close to the queried locations and the weak assumption in the technique used in determining the best sampling times, which assumes similar periodic patterns in the data. Nevertheless, our approach still outperforms the baseline.
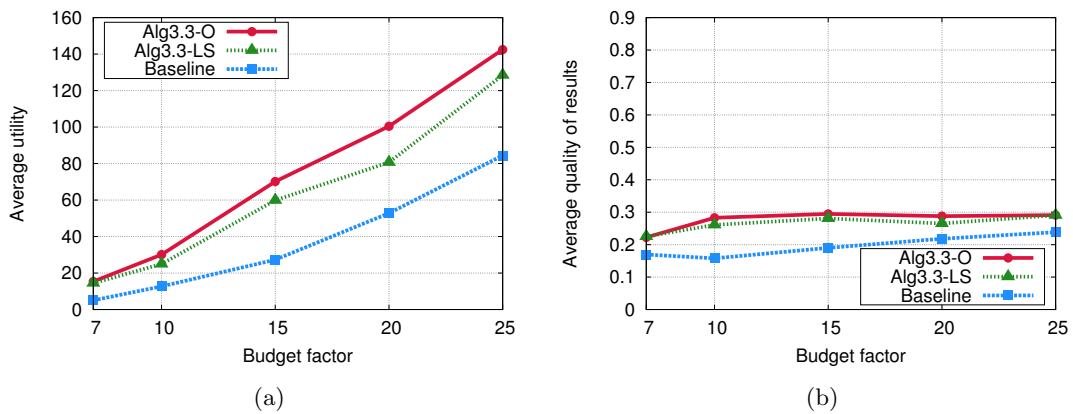
Figure 3.9: Location monitoring queries, a) average utility per time slot, b) average quality of results.

3.4.6 Region Monitoring Queries

In this experiment we assign the valuation function (3.7) to all region monitoring queries. The parameters of the Gaussian model are learned from a fraction of sensor readings in
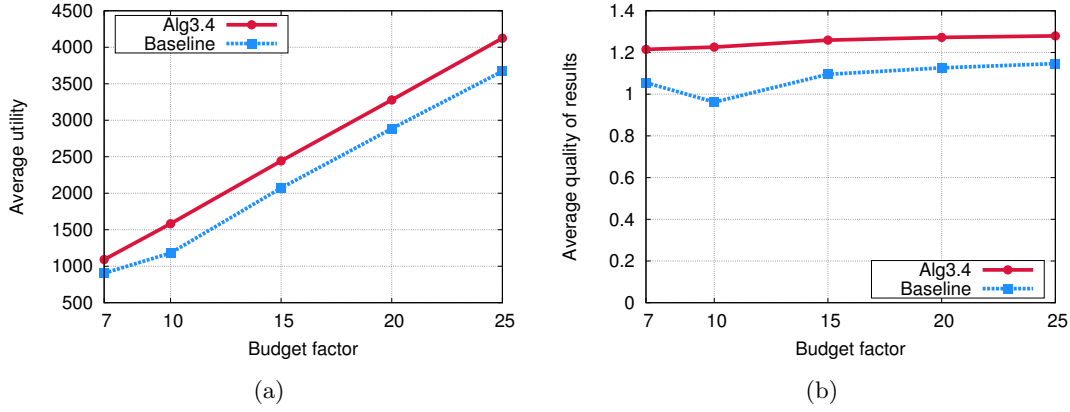
Figure 3.10: Region monitoring queries, a) average utility per time slot, b) average quality of results.

Intel Lab dataset. Function f_q in Algorithm 3.4 is implemented based on Algorithm 3.5. At each time slot one query is created with the query region randomly generated in the simulation area. The duration of the query is randomly chosen in $[5, 20]$. The budget assigned to each query is calculated as $\frac{A(r_q)}{3\pi r_s^2}b$, where $A(r_q)$ is the size of the queried region, r_s is the average coverage distance of the sensors (2 in this case), and b is the budget factor. The parameter α is set to the constant value 0.5. The following weight function is used to modify the cost of a sensor which falls into the region of $k > 0$ region monitoring queries:

$$w(k) = \begin{cases} \frac{11-k}{10} & k < 10 \\ 0.1 & otherwise. \end{cases} \tag{3.18}$$

Figure 3.10(a) shows the average utility per time slot w.r.t. the budget factor using Algorithm 3.4 compared to a baseline approach. In Algorithm 3.4 we use the optimal solution for answering point queries. In the baseline approach we do not use cost weighting and we omit sharing sensors that are selected for other queries and are not at the locations requested by the query. In addition, the baseline approach introduced in Section 3.4.3 is used for answering the point queries. Figure 3.10(b) shows that, most of the times, the average quality of results is more than 1, which means that the valuation of sensors selected for each query is more than what was requested by the queries. Note that this is possible since $F(\mathcal{A})$ is not bounded by 1.

3.4.7 Query Mix

In addition to Algorithm 3.6, we have implemented a baseline algorithm for answering a mixture of queries of different types. In this algorithm, first the aggregate queries are executed using the baseline algorithm for aggregate queries. The cost of selected sensors is set to zero for subsequent queries in the current time slot. In the next step, the required point queries are generated for continuous queries and then they are executed along with the point queries issued by end users using the baseline algorithm for answering single-sensor point queries. This baseline resembles sequential execution of queries in one time
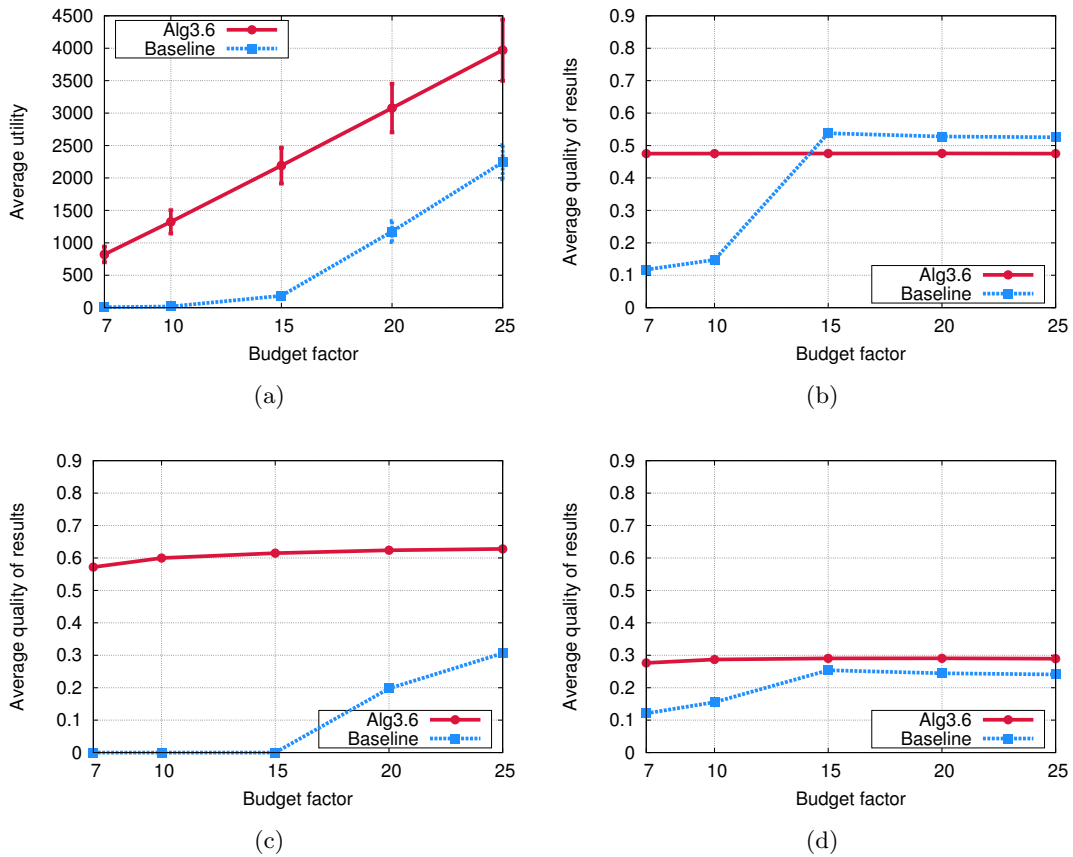
Figure 3.11: Mix of point, aggregate and location monitoring queries. a) average utility per time slot for query mix, b) average quality of results for point queries, c) average quality of results for aggregate queries, d) average quality of results for location monitoring queries.

slot with buffering data for the period of that time slot. The number of point, aggregate, and location monitoring queries is the same as in the experiments for each individual query type. Due to the lack of complete measurement data in RNC, we exclude region monitoring queries in this experiment. Sensor lifetime is set to 25 and a random privacy sensitivity level is assigned to each sensor. The linear energy cost function is used by each sensor with parameter β randomly chosen in $[0, 4]$.

Figure 3.11(a) shows the average utility per time slot w.r.t. the budget factor. It can be seen that Algorithm 3.6 significantly outperforms the baseline approach. As Figures 3.11(b), 3.11(c), and 3.11(d) show, the quality of results produced by the baseline approach for each query type is either zero or very small when the budget is small. In contrast, our approach can satisfy many queries even when the budget is small thanks to more efficient sensor sharing.

In order to observe the impact of the trust value distribution, three trust assignment schemes are considered in the next experiment. In the first scheme the trust values are assigned uniformly at random from the interval $[0, 1]$. In the second scheme the trust value is selected uniformly at random from the interval $[0.5, 1]$. In the last scheme all the
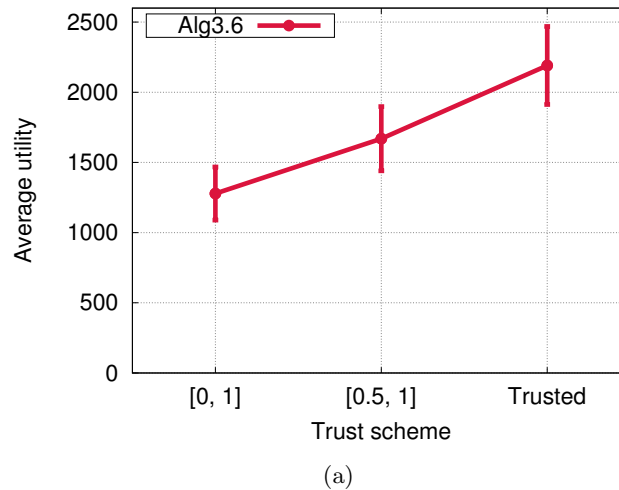
Figure 3.12: Average utility per time slot for three different trust assignment schemes for a mix of point, aggregate and location monitoring queries. The budget factor is 15. Privacy sensitivity levels and linear energy cost factor are randomly chosen.

sensors are assumed to be fully trusted. Figure 3.12(a) shows that the average utility per time slot increases as the trust value mean increases. This trend is indeed expected because the utility has a direct relation with the trustworthiness of the sensors.

3.5 Related Work

In Chapter 2 we have presented the state of the art in the areas of sensor selection and query processing and optimization in participatory sensing and sensor networks. In this section, we provide a brief overview of some of these works that are more relevant and compare them to our approach.

A utility-based sensor selection framework is proposed in [16] in which the applications can specify the utility of each set of sensors in a wireless sensor network. Submodular and supermodular utility function classes are considered. The goal is to select a sequence of sets to maximize the total utility while not exceeding the available energy. In [115], the problem of sensor selection, where a set of sensors is selected according to the maximum a posteriori or the maximum likelihood rules, is formulated as optimizations of submodular functions over uniform matroids. A heuristic approach based on convex optimization is proposed in [58] for the sensor selection problem with the objective of minimizing the estimation error. In our scenario, the network model is different and the objective is to maximize the net benefit. As we allow multiple applications, which potentially have different valuation functions, we cannot identify upfront in which function category our utility function falls.

Simultaneous placement and scheduling of sensors is considered in [70], where an algorithm is proposed to efficiently and simultaneously decide where to place sensors and when to activate them using the submodularity of the utility function. Two distributed sensor scheduling approaches are proposed in [46, 51]. These works are based

on the assumption that the utility function is submodular. In our approach we pursue a centralized solution, which is not restricted to submodular utility functions in order to be able to handle applications with diverse requirements.

The work of [69] is perhaps the closest to our approach. We distinguish our work in two main ways: 1) [69] tries to maximize the utility of data collection for the queried locations assuming that the sensors are fully trusted and the budget is fixed. In contrast, we aim for maximizing the utility of several concurrent queries, potentially of different types, assuming that the sensors are not fully trusted; 2) [69] assumes that the phenomenon follows a known distribution and utilize this for the near-optimal sensor selection, whereas we do not have any explicit assumption on the phenomenon and we obtain the utility functions from the applications.

The problem of multi-query processing has been systematically defined in [112] in the context of relational database systems. Lifetime-based and event-based queries are introduces along with normal queries in sensor networks in [81]. Optimization techniques such as reordering of predicates and event query batching has been used to preserve power. Merging multiple user queries into one network query and then extracting user data streams from network data streams is proposed in [93]. Optimizing multiple aggregate queries in sensor networks is studied in [130] with the objective of minimizing the communication cost while taking into account the processing limitations of the sensor nodes. In order to reduce energy consumption in a wireless sensor network, rewriting a new monitoring query based on the existing ones and evaluating it in the base station rather than injecting it into the network is proposed in [73]. In the AdaptiveCQ framework [129], for efficient processing of multiple continuous queries, the intermediate results of queries are shared at a fine level without materializing them on disk. [60] proposes a query planner for distributed stream processing systems which exploits overlaps among queries and sharing partial results with the objective of efficient resource allocation. In our approach, data sharing is implied without using techniques such as query rewriting. However, after data acquisition, if necessary, more sophisticated query optimization techniques can be performed in the higher level query processor.

3.6   Conclusion

In this chapter we proposed a holistic data acquisition framework for participatory sensing environments, where multiple applications may pose multiple queries of different types. We formulated the problem of optimal multi-query data acquisition with the objective of maximizing the total utility. We proposed heuristic algorithms for maximizing the utility in a myopic way for the most important query types and their mixes in this context. As a particular example, we considered efficient data acquisition for continuous queries in a participatory sensing environment with no guarantees on the data availability neither spatially nor temporally.

Truthful Data Acquisition in Participatory Sensing

4.1 Introduction

The promising features of participatory sensing, come with several challenges that need to be addressed. One of these challenges is that many people participate in participatory sensing systems as long as they are compensated for their resource consumption and privacy leakage. Many participants are sensitive to their privacy and might use privacy protection mechanisms for reducing their privacy leakage. Privacy protection mechanisms usually introduce some noise in the reported data or refrain to report the existence of the participant in certain locations. Another issue is that we cannot always assume that all participants are trustworthy as some of them might have incentives to falsify the data. This can make data collection less efficient as it reduces the quality of the data and possibly the coverage.

For overcoming these challenges, we need to design participatory sensing systems in a way that (I) enough incentives are provided to people to participate; (II) enough incentives are provided to the participants to truthfully report their measurements; and (III) the participants are motivated to trade their privacy for more payoffs. In Chapter 3 we assumed that the participants are compensated for the measurements that they provide according to the price that they announce. However, the participants can overstate their price to gain more payment. Therefore, mechanisms are needed to incentivize people to participate and to truthfully report their cost information and their measurements. In the rest of this chapter we use the term *agent* to refer to the participants.

The aim of the work in this chapter is to design mechanisms for truthfully eliciting information from the agents in two main settings: when the agents are not concerned about their location privacy and they are willing to reveal their exact location, and when they are location privacy-aware and refrain from reporting their actual location. In all

the settings we assume that the ground truth is observable after the aggregator (*center* in mechanism design terms) obtains measurements from the agents. This can be the case, for example, in a traffic monitoring setting, where the state of the traffic can be observed sometime after reporting the measurements. Even though the assumption of observing the ground truth holds in several cases, it is possible to estimate the ground truth when it is not observable. For example, the ground truth can be estimated using a spatio-temporal model of the phenomenon combined with multiple readings from different sensors [38, 89, 99, 134]. Finally, in this work, we focus on data acquisition in a single time slot and do not consider long-term query processing and sensor scheduling.

Chapter 2 reviews the important literature in truthful data elicitation. The participatory sensing scenario that we consider in this chapter is different from the common scenario assumed in most of the existing work. We assume that the center cannot determine the value of the reports from the agents on its own. For this purpose, it uses the valuation functions that are provided by the applications. No restrictions are imposed on the form of valuation functions. In addition, in many of the existing works, the privacy-reward trade-off is not addressed. For example, in the approaches based on *proper scoring rules* such as [89, 99, 134, 145], the center cannot generate scoring rules assuming that a single distribution of the event exists because each application might use a different distribution. None of these works, and works such as [38, 67, 102, 105], consider privacy conscious agents.

The main contributions of this chapter are the following:

- We formulate the problem of optimal data acquisition for multiple point queries and we propose incentive compatible mechanisms for truthful cost and data elicitation in participatory sensory context.

- We also propose mechanisms for truthful data elicitation when participants are privacy conscious by allowing them to make trade-offs between their privacy and monetary compensation. This trade-off is performed in a way that the center's utility is maximized.

- Through extensive simulations we demonstrate the effectiveness of our mechanisms.

The remainder of the chapter is organized as follows. In Section 4.2, we formulate the problem of optimized sensor selection. We present our mechanisms for truthful data elicitation in Section 4.3 and evaluate properties of these mechanisms in Section 4.4. Finally, we conclude this chapter in Section 4.5.

4.2 Optimized Sensor Allocation

In this section, we formulate the problem of optimal data acquisition for multiple point queries. In a participatory sensing system, different types of queries can be posed by the end users. For example, they can issue *point queries* asking for the current value of a phenomenon (e.g., CO_2 level) at a specific location, or they can issue *spatial aggregate*

queries asking for the aggregate value of the phenomenon over a region. In Chapter 3, we introduced different query types and data acquisition algorithms. In this chapter, we limit our focus only to point queries. In addition, we assume that for answering point queries at a single location, only one measurement is required. If multiple queries ask for a measurement at a specific location, the data obtained from a sensor at or near that location is used for all of them.

Assume n agents are available. The sensor corresponding to agent i is denoted by s_i. $S = \{s_1, s_2, \ldots, s_n\}$ denotes the ordered set of sensors. Each query q has a limited budget to spend for obtaining the query answer. Also, every query q comes with a valuation function $\bar{v}_q(s_i)$ that evaluates the quality of a measurement from sensor s_i, if a measurement from s_i is to be used to answer the query. Since $\bar{v}_q(s_i)$ is used before observing the actual measurement of s_i, it gives an expectation of the valuation of the measurement. Another function $v_q(x, \hat{x}_i, s_i)$ is assumed to be provided by each query q, that gives the valuation of a measurement $\hat{x}_i$ reported by sensor s_i, given the true value of the phenomenon x at location of s_i. When the agents are privacy conscious, valuation functions are expected to consider the uncertainty in the location of sensors (Section 4.3.2). The values given by the valuation functions are of unit of the currency that is used for sensor costs and payments. Therefore, valuation functions must incorporate the limited budget of the queries.

While valuation functions are provided by the queries and in our mechanisms they are taken as black boxes, we can assume that the evaluation is based on the *quality* of sensor readings for the queries. Sensor reading quality depends on several factors such as the distance of the sensor from the queried location (more accurately, it depends on the correlation between the phenomenon values at the queried location and the location of the sensor), and the inherent sensing inaccuracy of the sensor.

In a given time slot the center collects several point queries asking for the phenomenon value at various locations. The primary objective of the center is to answer the queries in a way that the total utility provided to the queries are maximized. For doing so, it has to solve an allocation problem to find the best sensors for providing measurements.

We can express the optimized sensor allocation problem as an (Binary) Integer Linear Program (ILP) exactly as it is described in Section 3.3.1. Assume n sensors are available and L locations are queried. For each queried location l, by m_l queries, we define a binary variable $Y_i^l \in \{0, 1\}$ for each $i = 1, \ldots, n$, which states if sensor s_i is assigned to location l. For each sensor s_i, let $X_i \in \{0, 1\}$ denote if s_i is assigned to any location. We denote by c_i the cost of sensor s_i. The following integer linear program solves the problem of optimally assigning sensors to answer queries such that the center's utility is maximized. It is also assumed that only one sensor is enough for answering all the queries at each

Symbol	Semantic
l_i	location of agent i (or sensor s_i)
l_q	location queried by query q
Q/Q_l	all queries/queries for location l
$c_i/\hat{c}_i$	actual/reported cost of agent i
$\mathcal{K}/K$	set of all possible allocation schemes/specific allocation scheme
π_i	payment to agent i
$\bar{u}_i$	expected utility of agent i
γ_i	number of cells in the obfuscation region (obfuscation level) of agent i
R_i	obfuscated region reported by agent i
$\hat{x}_i/x_i$	measurement reported by/true value at location of agent i

Table 4.1: Summary of introduced symbols

location.

$$
max \sum_{l=1}^{L} \sum_{i=1}^{n} v'_l(s_i) Y_i^l - \sum_{i=1}^{n} c_i X_i,
$$
$$
s.t.
$$
$$
Y_i^l \leq X_i \ \forall i, l \tag{4.1}
$$
$$
\sum_{i=1}^{n} Y_i^l \leq 1 \ \text{ for } l = 1, \ldots, L
$$

In the above formula, $v'_l(s_i)$ is given by:

$$
v'_l(s_i) = \begin{cases} v_l(s_i) & \text{if } v_l(s_i) > 0 \\ -1 & \text{otherwise}, \end{cases}
$$

where $v_l(s_i) = \sum_{j=1}^{m_l} \bar{v}_{q_j}(s_i)$. When $v_l(s_i) \leq 0$, then the above definition ensures $Y_i^l = 0$. Table 4.1 summarizes the notation we frequently use throughout this chapter.

4.3 Mechanisms for Truthful Data Elicitation

Having formulated optimal data acquisition for multiple point queries, in this section we use the principals of Vickrey-Clarke-Groves (VCG) mechanisms for truthful data elicitation. However, our mechanisms are two-stage mechanisms because a single VCG mechanism does not work in our scenario. In a standard (single-step) VCG mechanism, the utility of the center is calculated based on the reported types of the agents. However, in our case, the utility of the center not only depends on the reported costs of the agents, but also on the measurements that the selected agents provide to fulfill the sensing tasks. Moreover, reporting costs and measurements cannot be merged in single stage because the agents report their measurements only when they are selected by the center to do so. In principle, our mechanisms consist of two stages. In the first stage a set of agents is selected to fulfill the sensing task, based on their reported costs, such that the expected center's utility is maximized. In the second stage each selected agent reports its measurement. The reported measurements are used to answer queries. Considering

---

Mechanism 1: **SQ**

---

1. First Stage

    (a) Center asks agents to report locations and costs

    (b) Center solves sensor assignment problem based on reported costs and locations, and reputation of agents. Formally, it chooses agent $i = \arg\max_{j \in \{1,2,\dots,n\}} \bar{v}_Q(s_j) - \hat{c}_j$, where $\hat{c}_j$ is reported cost of agent j

2. Second Stage

    (a) Agent i reports its measurement $\hat{x}_i$

    (b) After observing actual outcome x, center makes payment $\pi_i = v_Q(x, \hat{x}_i, s_i) - \bar{v}_Q(s_j) + \hat{c}_j$ to agent i, where $j = \arg\max_{k \in \{1,2,\dots,n\} \setminus \{i\}} \bar{v}_Q(s_k) - \hat{c}_k$

---

the accuracy of this measurement, after observing the true value, the payment to the agent is calculated.

4.3.1 Privacy Oblivious Agents

We start with the simplest case where it is assumed that the agents truthfully reveal their locations. This is a reasonable assumption because if an agent gets selected to report its data, its utility depends on the accuracy of the report. We assume that the ground truth can be observed after the agents report their measurements. Having multiple rounds and a reputation mechanism that assigns reputation scores to each agent incentivise the agents to not lie if they are interested in increasing their long-term utility. However, in this work we don't deal with this issue.

4.3.1.1 Single Query Location

We consider the case of existing queries at a single location and we seek to select a sensor that can provide data for this location which yields the greatest utility for the queries. Let Q denote the set comprising these queries. The expected valuation of a sensor s_i is denoted by $\bar{v}_Q(s_i) = \sum_{q \in Q} \bar{v}_q(s_i)$, which is calculated based on the expected quality of a measurement from s_i.

After observing the true value x, we use $v_Q(x, \hat{x}_i, s_i) = \sum_{q \in Q} v_q(x, \hat{x}_i, s_i)$, for calculating the actual valuation. We propose **SQ**, a two-stage mechanism for incentivising the agents to truthfully report their costs and measurements. This mechanism is inspired by [102], which addresses assigning tasks to agents whose private information is not only their costs of performing tasks but also their failure probabilities. However, in our scenario in addition to the costs, we deal with the quality of reported measurements.

SQ is incentive compatible in the first stage regarding the costs. It is also incentive compatible in the second stage with regard to reporting the measurement. Finally, it is individually rational in expectation. With the payment defined in the second stage of

SQ, the expected utility of the selected agent is:

$$\bar{u}_i = \bar{v}_Q(s_i) - \bar{v}_Q(s_j) + \hat{c}_j - c_i, \tag{4.2}$$

where c_i is the actual cost of agent i. We show that reporting the true cost is the dominant strategy in the first stage. If $\hat{c}_i > c_i$, there are two possibilities: 1) agent i is the selected agent, but since its utility does not depend on $\hat{c}_i$, it will receive the same utility as when it reports the true cost, and 2) agent i is not selected and hence receives no utility, but it would be selected if the agent reported the actual cost. Therefore, agent i does not have any incentive to exaggerate its cost. If $\hat{c}_i < c_i$, there are again two possibilities: 1) agent i would be selected even if it told the truth and therefore, agent i does not increase its utility, and 2) agent i would not be selected if it reported its true cost. From $\bar{v}_Q(s_i) - c_i < \bar{v}_Q(s_j) - \hat{c}_j$ it follows that:

$$\bar{u}_i = \bar{v}_Q(s_i) - \bar{v}_Q(s_j) + \hat{c}_j - c_i < 0.$$

Hence, agent i does not have any incentive to under-report its cost. This proves that reporting true costs in the first stage is the dominant strategy.

Reporting true measurements in the second stage is the dominant strategy since the only way agent i can increase its utility is to report a value as close as possible to the true value (which will later be determined by the center). If agent i is not selected then it gets zero utility, otherwise it will receive in expectation $\bar{u}_i = (\bar{v}_Q(s_i) - c_i) - (\bar{v}_Q(s_j) - c_j) > 0$. Hence, SQ is individually rational in expectation. Note that it is possible that after reporting its measurement, agent i receives negative utility due to its accuracy being lower than the average accuracy expected by the center.

4.3.1.2   Multiple Query Locations - Optimal Allocation

Now we consider the more general case of having multiple query locations. The integer program (4.1) assigns sensors to queries such that the overall utility is maximized. However, for the ease of presentation we introduce a simplified notation. Let $\mathcal{K}$ be the set of possible allocation schemes. If $K \in \mathcal{K}$, then $v_q(K)$ is the valuation of query q for allocation K. Hence, if s_i is selected to provide a measurement for query q, then $\bar{v}_q(K) = \bar{v}_q(s_i)$. We denote by $S(K)$ and $L(K, i)$ the set of sensors that are allocated to queries and the set of query locations that are assigned by K to sensor s_i, respectively. The set of queries at location l is denoted by Q^l. This setting is somewhat similar to the case of multi-task assignment with non-combinatorial valuation in [102]. However, our problem differs from theirs in the sense that instead of dealing with the probability of success in performing the tasks, we are concerned with the accuracy with which the tasks are performed. Mechanism MQ_{OPT}, which is the generalized version of SQ, incentivises the agents to truthfully report their costs and then their measurements for the queries assigned to them.

MQ_{OPT} has all the economic properties of SQ. We can prove that reporting true costs in the first stage is the weakly dominant strategy for agents. Incentive compatibility regarding reporting measurements in the second stage and individual rationality in

---

Mechanism 2: $\textbf{MQ}_{\textbf{OPT}}$
___

1. First Stage

   (a) Center asks agents to report locations and costs

   (b) Center solves sensor assignment problem based on reported costs and locations. Formally, it finds allocation

   $$K^* = \underset{K \in \mathcal{K}}{\arg\max} \left(\sum_{l \in L} \bar{v}_{Q^l}(K) - \sum_{i \in S(K)} \hat{c}_i \right)$$

2. Second Stage

   (a) Each agent $i \in S(K^*)$ reports its measurement $\hat{x}_i$ to center

   (b) After observing actual outcomes x_i for all $i \in S(K^*)$, center makes following payment to each selected agent i:

   $$\pi_i = \sum_{l \in L(K^*,i)} v_{Q^l}(x_i, \hat{x}_i, s_i) + \sum_{l \in L(K^*,-i)} \bar{v}_{Q^l}(K^*) - \sum_{j \in S(K^*) \setminus \{i\}} \hat{c}_j$$
   $$- \max_{K' \in \mathcal{K}_{-i}} \left(\sum_{l \in L} \bar{v}_{Q^l}(K') - \sum_{j \in S(K')} \hat{c}_j \right),$$

   where $\mathcal{K}_{-i}$ is set of allocations excluding i and $L(K^*, -i)$ is set of locations assigned to agents other than i.

---

expectation can also be proved. We proceed by proving the incentive-compatibility of the mechanism by assuming that agent i reports $\hat{c}_i$ instead of its true cost c_i. We consider two cases. If $\hat{c}_i > c_i$, then two outcomes are possible: 1) agent i is among the selected agents. Since i's utility does not depend on its reported cost, it gains nothing, and 2) agent i is not selected but it would be selected and hence it would receive positive utility (in expectation) if it told the truth. Therefore, agent i does not have any incentives to overstate its cost. If $\hat{c}_i < c_i$, then again we consider two possibilities: 1) agent i is not selected hence receives no utility, and 2) agent i is selected but it would not be among the selected agents if it told the truth. We denote the optimal allocation by $\hat{K}^*$ given the reported costs of all agents including $\hat{c}_i$. Define $w(K) = \sum_{l \in L} \bar{v}_{Q^l}(K) - \sum_{j \in S(K)} \hat{c}_j$, where $\bar{v}_{Q^l}(K) = \sum_{q \in Q^l} \bar{v}_q(K)$. Let $w_{-i}(K)$ denote the center's utility by all agents but i, and let K^*_{-i} denote the optimal allocation ignoring agent i. The expected utility of i can be defined as follows:

$$\bar{u}_i = \sum_{l \in L(\hat{K}^*,i)} \bar{v}_{Q^l}(s_i) + w_{-i}(\hat{K}^*) - w(K^*_{-i}) - c_i.$$

But $w(\hat{K}^*) = \sum_{l \in L(\hat{K}^*,i)} \bar{v}_{Q^l}(s_i) - \hat{c}_i + w_{-i}(\hat{K}^*)$. Therefore,

$$\bar{u}_i = w(\hat{K}^*) - w(K^*_{-i}) - c_i + \hat{c}_i.$$

It can be easily seen that $w(\hat{K}^*) - w(K^*_{-i}) \leq c_i - \hat{c}_i$ (or else i wouldn't be selected by $\hat{K}^*$). It follows that $\bar{u}_i \leq 0$. Hence, agent i does not gain in utility by understating its cost.

If agent i does not get selected, it receives zero utility by participation. If it gets selected it's expected utility is $\bar{u}_i = \sum_{l \in L(K^*,i)} \bar{v}_{Q^l}(s_i) + w_{-i}(K^*) - w(K^*_{-i}) - c_i$. Since the agent is truthful, $\hat{c}_i = c_i$. Therefore, $\bar{u}_i = w(K^*) - w(K^*_{-i})$. Since i has been selected by the optimal allocation, $w(K^*) \geq w(K^*_{-i})$, which follows that $\bar{u}_i \geq 0$. This shows that the mechanism is individually rational in expectation.

4.3.1.3   Multiple Query Locations - Approximate Allocation

The linear program (4.1) can be solved in reasonable times when the problem instance is not very large. Otherwise we need to resort to approximation algorithms. The payment scheme in $\boldsymbol{MQ_{OPT}}$ does not guarantee incentive compatibility of the mechanism when the allocation scheme is not optimal. However, we can prove that our optimal allocation problem can be formulated as maximizing a *non-monotone submodular* function. More specifically, we can formulate the center's utility function as:

$$u(S') = \sum_{l \in L} \max_{s_i \in S'} v_{Q^l}(s_i) - \sum_{s_i \in S'} c_i. \tag{4.3}$$

This function states that for a set of selected senors S', the utility is calculated by assigning a sensor to each location such that the valuation of the queries at that location is maximized. The objective of the optimal allocation is to find a subset of S that maximizes $u(.)$. Algorithm 4.1 presents the deterministic $\frac{1}{3}$-approximation algorithm for maximizing (non-negative) non-monotone submodular functions that is proposed in [17].

Algorithm 4.1: DeterministicUSM

Data: Non-monotone submodular function $u(.)$, and set of all sensors S, where
$\quad\quad |S| = n$
Result: Set of selected sensors $\mathcal{X}_n$
$\mathcal{X}_0 \leftarrow \emptyset, \mathcal{Y}_0 \leftarrow S$
for $i \leftarrow 1$ **to** n **do**
$\quad\quad a_i \leftarrow u(\mathcal{X}_{i-1} \cup \{s_i\}) - u(\mathcal{X}_{i-1})$
$\quad\quad b_i \leftarrow u(\mathcal{Y}_{i-1} \backslash \{s_i\}) - u(\mathcal{Y}_{i-1})$
$\quad\quad$ **if** $a_i \geq b_i$ **then**
$\quad\quad\quad\quad \mathcal{X}_i \leftarrow \mathcal{X}_{i-1} \cup \{s_i\}, \mathcal{Y}_i \leftarrow \mathcal{Y}_{i-1}$
$\quad\quad$ **else**
$\quad\quad\quad\quad \mathcal{X}_i \leftarrow \mathcal{X}_{i-1}, \mathcal{Y}_i \leftarrow \mathcal{Y}_{i-1} \backslash \{s_i\}$
return $\mathcal{X}_n$

It is known that any (normalized [1]) auction that has the following characteristics is incentive compatible ([11]):

---
[1]An auction is normalized if a losing agent has a zero payment.

1. The allocation scheme is *bid-monotone*: if user i wins by bidding b_i, it also wins by bidding $b_i' \leq b_i$.

2. Each winner pays its *critical value*.

For a winning agent i we define c_i^c as its *critical cost* if agent i wins with any $\hat{c}_i \leq c_i^c$ and loses with any $\hat{c}_i > c_i^c$. The critical cost can be found by performing a progressive binary-like search assuming that the cost is not a continuous real value. We define the following payment scheme:

$$\pi_i = t_i + \beta \sum_{l \in L(\tilde{K}^*, i)} v_{Q^l}(x_i, \hat{x}_i, s_i), \tag{4.4}$$

where $\tilde{K}^*$ is the allocation scheme given by Algorithm 4.1, and $t_i = c_i^c - \beta \sum_{l \in L(\tilde{K}^*, i)} \bar{v}_{Q^l}(s_i)$, and $0 < \beta \leq 1$ is a tuning factor. We call the resulting mechanism $\boldsymbol{MQ_{APPROX}}$.

It is easy to prove that the above algorithm is bid-monotone: Assume agent j is among the selected agents when it reports its true cost c_j. Now assume that j reports a lower cost $\hat{c}_j < c_j$. Let $\Delta c_j = c_j - \hat{c}_j$, and consider step j of the algorithm. Denote by a_j', b_j', and u' the utility differences and the utility function when j is not truthful. We have

$$
\begin{aligned}
a_j' &= u'(\mathcal{X}_{j-1} \cup \{s_j\}) - u'(\mathcal{X}_{j-1}) \\
&= u(\mathcal{X}_{j-1} \cup \{s_j\}) - u(\mathcal{X}_{j-1}) + \Delta c_j = a_j + \Delta c_j, \\
b_j' &= u'(\mathcal{Y}_{j-1} \backslash \{s_j\}) - u'(\mathcal{Y}_{j-1}) \\
&= u(\mathcal{Y}_{j-1} \backslash \{s_j\}) - u(\mathcal{Y}_{j-1}) - \Delta c_j = b_j - \Delta c_j.
\end{aligned}
$$

From $a_j \geq b_j$ we can conclude that $a_j' \geq b_j'$. This shows that s_j will be included in $\mathcal{X}_j$, hence j will be selected.

The bounds for Algorithm 4.1 can be guaranteed when the function is non-negative, while $u(.)$ can be negative. In order to resolve this issue, we can add $\sum_{s_i \in S} c_i$ to $u(.)$ to guarantee that it will never be negative. However, in Algorithm 4.1 this modification is not necessary since this additional term will be cancelled out in all calculations.

4.3.2 Privacy Conscious Agents

So far we have assumed that the agents reveal their exact location to the center. However, it might be the case that some agents are concerned about their location privacy. In this case, they use a location privacy protection mechanism to reduce the probability of inferring their exact location. One common approach for protecting location privacy is called *location obfuscation*, which reduces the accuracy and/or precision of the reports [118]. Among different location obfuscation methods, we assume the agents use either the *perturbation* method, which adds some random locations, or *reducing precision or region merging* method which adds a set of locations around the real location of the agent. These locations along with the true location are then reported to the center. For

---

Mechanism 3: $\boldsymbol{PRIV_{STRICT}}$

---

1. First Stage

    (a) Center asks agents to report obfuscated locations and costs

    (b) Center solves sensor assignment problem based on reported costs and locations of the agents. Formally, it finds the allocation

$$K^* = \underset{K \in \mathcal{K}}{\arg\max} \sum_{i \in S(K)} \left(\sum_{l \in L(K,i)} \tilde{v}_{Q^l}(s_i, R_i) - \hat{c}_i \right),$$

    where $\tilde{v}_{Q^l}(s_i, R_i)$ gives expected valuation of a measurement of s_i located somewhere in R_i, for queries at l.

2. Second Stage

    (a) Each agent $i \in S(K^*)$ reports its measurement $\hat{x}_i$ to center

    (b) After observing actual outcomes for all $r \in R_i$, center makes following payment to each of the selected agents i:

$$\pi_i = h(K^*, R_i, \hat{x}_i) + \sum_{j \in S(K^*) \backslash \{i\}} \left(\sum_{l \in L(K^*,j)} \tilde{v}_{Q^l}(s_j, R_j) - \hat{c}_j \right)$$

$$- \max_{K' \in \mathcal{K}_{-i}} \left(\sum_{l \in L} \tilde{v}_{Q^l}(K') - \sum_{j \in S(K')} \hat{c}_j \right). \qquad (4.5)$$

---

doing so, we divide the area into cells. The value of the phenomenon is assumed to be uniform over each cell.

We denote by R_i the set of cells that agent i reports. R_i contains r_i, the actual cell in which i is located (i.e., $l_i \in r_i$). We assume that each agent has a location profile ψ, which is a probability distribution over all cells (i.e., $\psi_i(r)$ is the probability of agent i to be in cell r). We assume that this information is common between the agent and the center. Even though, location profile is time-dependent, for simplicity we ignore the time dimension and assume that $\psi_i(r)$ is the location profile at current time instance.

It is worthwhile to mention that knowing the location profile of users by adversaries, is a common assumption in the literature on location privacy, e.g., [119, 120, 122]. This knowledge does not necessarily violate privacy since it incorporates possibly high degree of uncertainty. The center can learn location profiles of users based on their previous reports, some training traces of users which might be noisy or incomplete, and/or background information about users.

4.3.2.1 No Privacy-Cost Trade-off

In the basic setting, agents have a preferred obfuscation level and a cost associated with it. They are not willing to violate their preferred obfuscation level in any circumstances.

PRIV $_{STRICT}$ is the generalized version of **MQ $_{OPT}$** which takes into account the obfuscated locations of the agents. In this mechanism, $\tilde{v}_{Q^l}(s_i, R_i) = \sum_{q \in Q^l} \tilde{v}_q(s_i, R_i)$, where $\tilde{v}_q(s_i, R_i)$ is a valuation function of query q that gives the expected valuation of a measurement from agent i knowing that i is located in a cell in R_i.

$h(K^*, R_i, \hat{x}_i)$ in payment scheme (4.5) is a function that calculates the valuation of $\hat{x}_i$ for the queries that will be answered by data provided by agent i. In order to guarantee incentive compatibility with respect to reporting true measurements, h must satisfy the following condition:

$$\bar{h}(K^*, i) = \sum_{l \in L(K^*, i)} \tilde{v}_{Q^l}(s_i, R_i), \tag{4.6}$$

where $\bar{h}(K^*, i)$ is the expected value of $h(K^*, R_i, \hat{x}_i)$ before receiving $\hat{x}_i$ from agent i.

Roughly speaking, the (risk neutral) agents do not have any incentive to not include their actual location in the reported locations for two reasons: (I) it may reduce their probability of getting selected by the center given that the agents do not have any information on the possible queries, and (II) even if such an agent is selected by the center, its utility depends on the quality of its measurement. A proper quality assessment method can detect the irrelevance of the measurement to the announced location.

Incentive compatibility in both stages and individual rationality (in expectation) of this mechanism still hold because we have only changed the valuation calculation method. In other words, the approach we took in proving the incentive compatibility and individual rationality of **MQ $_{OPT}$** can be used here. Note that π_i does not depend on the true types of other participants. This is crucial for incentive compatibility of the mechanism.

4.3.2.2 Privacy-Cost Trade-off

A more interesting scenario is when the agents are willing to give up some of their privacy in return for more payment. The agents' cost functions increase with the increase in their privacy leakage. The center needs to valuate the agents' reports, for which it ideally needs the exact locations of the agents. Therefore, the more precise the announced locations are, the higher valuation they yield. This encourages the agents to trade their privacy for more profit.

Agent i has a cost function $c_i : \Gamma_i \to \mathbb{R}$, where Γ_i is the set of i's obfuscation levels. For simplicity, we assume that an obfuscation level γ indicates the number of cells in the subregion reported by the corresponding agent. The only natural restriction that we impose on the cost function is that it must be monotonically decreasing. That is, $c_i(\gamma^{(1)}) < c_i(\gamma^{(2)})$ if $\gamma^{(1)} > \gamma^{(2)}$.

We propose **PRIV $_{TRADE}$** for truthfully eliciting cost and data reports from the agents with the objective of maximizing the center's utility. In the first stage, the center selects a set of agents $S(K^*)$, and for each agent $i \in S(K^*)$ calculates an obfuscation level $\Gamma(K^*, i)$. In the second stage, each selected agent i, reports its measurement and its region R_i such that $|R_i| = \Gamma(K^*, i)$. For calculating the expected valuation of a measurement from agent i for the queries at l, the center utilizes the function

Mechanism 4: **$PRIV_{TRADE}$**

1. First Stage

   (a) Each agent i reports cost function $c_i(.)$ and region R_i^0 corresponding to its highest obfuscation level

   (b) Center solves sensor assignment problem based on reported cost functions and initial regions. We define

   $$w(K, Q, \mathbf{R}^0, \mathbf{C}) = \sum_{i \in S(K)} \left(\sum_{l \in L(K,i)} \tilde{v}_{Q^l}\left(s_i, \Gamma(K, i), R_i^0\right) - \hat{c}_i\left(\Gamma(K, i)\right) \right), \quad (4.7)$$

   where $\mathbf{R}^0$ and $\mathbf{C}$ denote the set of initial regions and the set of cost functions. $\Gamma(K, i)$ denotes the obfuscation level selected by K for agent i. Obfuscated location of i will be in a subregion of R_i^0 according to γ.

   Then center finds the best allocation scheme K^*:

   $$K^* = \arg\max_{K \in \mathcal{K}} w(K, Q, \mathbf{R}^0, \mathbf{C}). \quad (4.8)$$

2. Second Stage

   (a) Each selected agent $i \in S(K^*)$ reports its region R_i and its measurement $\hat{x}_i$ to center

   (b) After observing actual outcomes for all $r \in R_i$, center makes following payment to each i:

   $$\pi_i = h(K^*, R_i, \hat{x}_i) + w_{-i}(K^*) - w(K^*_{-i}), \quad (4.9)$$

   where $w_{-i}(K^*) = w_{-i}(K^*, Q, \mathbf{R}^0, \mathbf{C})$ is center's utility achieved by K^* excluding agent i. $w(K^*_{-i}) = w(K^*_{-i}, Q, \mathbf{R}^0_{-i}, \mathbf{C}_{-i})$, is center's utility achieved by K^*_{-i}, which is the best allocation scheme when agent i is excluded from the list of agents. h is similar to the function that is used in **$PRIV_{STRICT}$**.

$\tilde{v}_{Q^l}\left(s_i, \gamma, R_i^0\right) = \sum_{q \in Q^l} \tilde{v}_q\left(s_i, \gamma, R_i^0\right)$. Function $\tilde{v}_q\left(s_i, \gamma, R_i^0\right)$ gives the valuation of a measurement from i for query q, knowing that i is located somewhere in R_i^0 and, if it gets selected, it will report a subregion $R_i \subseteq R_i^0$, where $|R_i| = \gamma$ and $l_i \in R_i$. This function is useful in scenarios where queries get some initial utility from knowing about the neighborhood in which the measurements are taken, but more precise location information is needed for extra utility.

We prove that **$PRIV_{TRADE}$** is individually rational and incentive compatible regarding reporting cost functions c_i by showing that revealing true costs is the dominant strategy of each agent. It is straightforward to prove incentive compatibility regarding reporting measurements in the second stage provided that condition (4.6) holds.

Proposition 1. **$PRIV_{TRADE}$** is individually rational.

Proof. We assume that agents are truthful. If agent i is not among the winners, its

utility is zero. If i is one of the winners, then it's expected utility is given by:

$$\bar{u}_i = \sum_{l \in L(K^*, i)} \tilde{v}_{Q^l}\left(s_i, \Gamma(K^*, i), R_i^0\right)$$
$$+ w_{-i}(K^*) - w(K_{-i}^*) - c_i(\Gamma(K^*, i))$$
$$= w(K^*) - w(K_{-i}^*).$$

Since i is a winner, the center's utility when i participates is greater than when i is excluded. Therefore, $\bar{u}_i = w(K^*) - w(K_{-i}^*) > 0$. This shows that the agents receive non-negative utility, in expectation, by participating in the mechanism. $\square$

Proposition 2. *$PRIV_{TRADE}$ is incentive compatible regarding reporting cost functions.*

Proof. Let K'^* denote the best allocation scheme when agent i misreports its cost function. Let γ_i and γ_i' be the obfuscation levels assigned to agent i by K^* and K'^*, respectively, if it wins. Let us denote by $\bar{u}_i$ and $\bar{u}_i'$, the i's expected utility when i is truthful and when it is not truthful, respectively. We can distinguish four cases:

(I) Regardless of being truthful or not, i loses. In this case, the expected utility of i is $\bar{u}_i = 0$.

(II) i loses with untruthful cost function, but it would win with true cost function. In this case, $\bar{u}_i' - \bar{u}_i < 0$.

(III) i wins with untruthful cost function, but it would lose with true cost function. This happens only when $\hat{c}_i(\gamma_i') < c_i(\gamma_i')$. The expected utility of i is:

$$\bar{u}_i' = \sum_{l \in L(K'^*, i)} \tilde{v}_{Q^l}(s_i, \gamma_i', R_i^0) + w_{-i}(K'^*) - w(K_{-i}^*) - c_i(\gamma_i')$$
$$= w(K'^*) - w(K_{-i}^*) + \hat{c}_i(\gamma_i') - c_i(\gamma_i')$$
$$= w(K'') - w(K_{-i}^*),$$

where K'' is the non-optimal allocation that selects i when it is truthful. We have used the fact that $w(K'^*) = w(K'') - \hat{c}_i(\gamma_i') + c_i(\gamma_i')$. It follows from $w(K_{-i}^*) > w(K'')$ that $\bar{u}_i' < 0$.

(IV) Regardless of being truthful or not, i wins. When $\gamma_i' = \gamma_i$, since the expected utility does not depend on the reported costs, $\bar{u}_i' - \bar{u}_i = 0$. When $\gamma_i' \neq \gamma_i$:

$$\bar{u}_i' - \bar{u}_i = \sum_{l \in L(K'^*, i)} \tilde{v}_{Q^l}(s_i, \gamma_i', R_i^0) + w_{-i}(K'^*) - w(K_{-i}^*) - c_i(\gamma_i')$$
$$- \left(\sum_{l \in L(K^*, i)} \tilde{v}_{Q^l}(s_i, \gamma_i, R_i^0) + w_{-i}(K^*) - w(K_{-i}^*) - c_i(\gamma_i) \right)$$
$$= w(K'^*) + \hat{c}_i(\gamma_i') - c_i(\gamma_i') - w(K^*)$$
$$= w(K'') - w(K^*),$$

where K'' is the non-optimal allocation that assigns γ_i' to agent i when it is truthful. Note that $w(K'') = w(K'^*) + \hat{c}_i(\gamma_i') - c_i(\gamma_i')$ regardless of the relation between γ_i' and

γ_i and the relation between $\hat{c}_i(\gamma_i')$ and $c_i(\gamma_i')$. It follows from $w(K_{-i}^*) > w(K'')$ that $\bar{u}_i' - \bar{u}_i < 0$.

Therefore, agent i gains zero or negative utility in expectation by misreporting its cost function. $\qquad\square$

The optimal allocation problem (4.8) can be formulated as a binary ILP. Assume n sensors are available and L locations are queried. For each queried location l, by m_l queries, we define a binary variable $Y_i^l \in \{0,1\}$ for each $i = 1, \ldots, n$, which states if sensor s_i is assigned to location l. For each sensor s_i, let $X_i \in \{0,1\}$ denote if s_i is assigned to any location. Let $\Gamma_i^j \in \{0,1\}$ denote if obfuscation level γ_j is selected for s_i, where $j = 1, \ldots, J_i$. Note that we substitute $Y_i^l \Gamma_i^j$ by $Z_{i,j}^l \in \{0,1\}$ to keep the linearity. We denote by c_i^j the cost of sensor s_i for obfuscation level γ_i^j. The optimal allocation problem (4.8) can be expressed as the following binary ILP:

$$max \sum_{l=1}^{L} \sum_{i=1}^{n} \sum_{j=1}^{J_i} Z_{i,j}^l \tilde{v}'_{Q^l}(s_i, \gamma_i^j, R_i^0) - \sum_{i=1}^{n} \sum_{j=1}^{J_i} \Gamma_i^j c_i^j,$$

$$s.t. \tag{4.10}$$

$$Y_i^l \leq X_i \ \forall i, l, \qquad \sum_{i=1}^{n} Y_i^l \leq 1 \ \forall l, \qquad \sum_{j=1}^{J_i} \Gamma_i^j \leq X_i \ \forall i,$$

$$Z_{i,j}^l \leq Y_i^l, \qquad Z_{i,j}^l \leq \Gamma_i^j, \qquad Z_{i,j}^l \geq Y_i^l + \Gamma_i^j - 1,$$

$$\forall i, l, j = 1, \ldots, J_i.$$

In the above formula, $\tilde{v}'_{Q^l}(s_i, \gamma_i^j, R_i^0)$ is given by:

$$\tilde{v}'_{Q^l}(s_i, \gamma_i^j, R_i^0) = \begin{cases} \tilde{v}_{Q^l}(s_i, \gamma_i^j, R_i^0) & \text{if } \tilde{v}_{Q^l}(s_i, \gamma_i^j, R_i^0) > 0 \\ -1 & \text{otherwise.} \end{cases}$$

4.4   Evaluation

We conducted extensive simulations using synthetic datasets to evaluate the performance of our mechanisms.

4.4.1   Setup

We use *random waypoint model* [57] to simulate the mobility of agents in a region of 20×20 grid. In this model, each agent moves from its current location with a speed randomly selected between zero and a sensor-specific maximum speed. The direction of the movement is either up, down, left, or right, and is randomly selected. The sensors are randomly spread in the region. The maximum speed of each sensor is set randomly to 4 or 5. We consider a simulation period of 50 time slots in all the experiments. A two-dimensional function is used to generate real data for each grid in the region. The measurement of a sensor located at a specific grid is simulated by adding to its real data a noise generated randomly from a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, with $\mu = 0$ and $\sigma = 1$.

We define the following function for computing the quality of a measurement from sensor s_i for query q, before receiving the measurement from the sensor.

$$\bar{\theta}(l_i, l_q) = \begin{cases} 1 - \frac{|l_s - l_q|}{d_{max}} & \text{if } |l_i - l_q| \leq d_{max}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.11}$$

In the above formula, l_i is the location of s_i, l_q is the queries location, and d_{max} is the maximum distance at which the agents can be considered to provide data. According to this function, the quality of a measurement linearly depends on the distance of the sensor and the queried location. Note that in (4.11) we assume that the agents will provide truthful measurements and the noise in the measurements is ignored.

In the experiments for privacy oblivious agents, we use the following as the valuation function of every query q:

$$\bar{v}_q(s_i) = \begin{cases} B_q \theta_{q,i} & \theta_{min} \leq \theta_{q,i} \leq 1 \\ 0 & \theta_{q,i} < \theta_{min}, \end{cases} \tag{4.12}$$

where B_q is the query budget, $\theta_{q,i} = \bar{\theta}(l_i, l_q)$, and θ_{min} is the minimum acceptable quality.

Point query locations are selected randomly in the simulation region. For all the queries, we set $\theta_{min} = 0.2$, $B_q = 10$, and $d_{max} = 5$.

4.4.2 Privacy Oblivious Agents

In the following experiments we compare the performance of MQ_{OPT} and MQ_{APPROX} regarding average utility and payment.

We use the following function to compute the quality of a measurement $\hat{x}_i$ reported by agent i knowing that the true value is x:

$$\theta(x, \hat{x}_i) = \begin{cases} (1 - \frac{|x - \hat{x}_i|}{5}) & \text{if } |x - \hat{x}_i| \leq 5, \\ 0 & \text{otherwise.} \end{cases} \tag{4.13}$$

According to this function the quality of a reported measurement decreases linearly as its difference with the ground truth value increases (up to a threshold of 5). Then, the formula of $v_q(x, \hat{x}_i, s_i)$ is similar to (4.12), except that the quality is given by (4.13). Agents (sensors) are initially placed at randomly selected grids in the simulation region. A cost value uniformly randomly picked from the interval $[5, 15]$ is assigned to each agent.

Varying the number of agents. Figure 4.1(a) and 4.1(b) show the average utility achieved by MQ_{OPT} and MQ_{APPROX} and the payment issued by these mechanisms as the number of agents increases. It can be seen that MQ_{APPROX} results in more utility compared to MQ_{OPT} in most cases. The obvious reason, as is seen in 4.1(b), is that MQ_{OPT} pays more to the agents compared to MQ_{APPROX}. It can also be noticed that the utility difference between these mechanisms decreases as the number of agents increases. This is due to the fact that, while the number of queries is fixed, the optimal allocation achieves higher valuation as the number of agents increases. Figure 4.1(c)

(a) Utility

(b) Payment

(c) Overpayment

Figure 4.1: **(a)** Average utility, **(b)** Average payment, **(c)** Average overpayment, by MQ_{OPT} and MQ_{APPROX} for different number of sensors (agents). 50 queries exist in each time slot. MQ_{APPROX} achieves more utility because it pays less to the agents.

shows the average *overpayment* of the two mechanisms. Overpayment is the amount paid to the winners in excess of their announced costs.

Varying the number of queries. The average utility and payment resulting from MQ_{OPT} and MQ_{APPROX} are compared in Figure 4.2(a) and 4.2(b). It can be noted that the utility and payment are inversely related. Moreover, both utility and payment go up as the number of queries increases. The reason is that more queries imply more budget. In addition, having fixed number of agents, with an increasing number of queries, the measurements taken from agents can be shared by more queries, which results in more utility. Figure 4.2(c) shows that MQ_{OPT} overpays the winners more than MQ_{APPROX} when less than 150 queries exists. For more than 150 queries, the opposite behavior is observed.

4.4.3 Privacy Conscious Agents

In the following experiments, we compare the performance of $PRIV_{TRADE}$ regarding average utility acquired and average payment issued in three different settings: (I) when agents are willing to trade their privacy (Trade-off); (II) when agents are privacy conscious but are not willing to trade their privacy as in $PRIV_{STRICT}$ (No trade-off); (III)
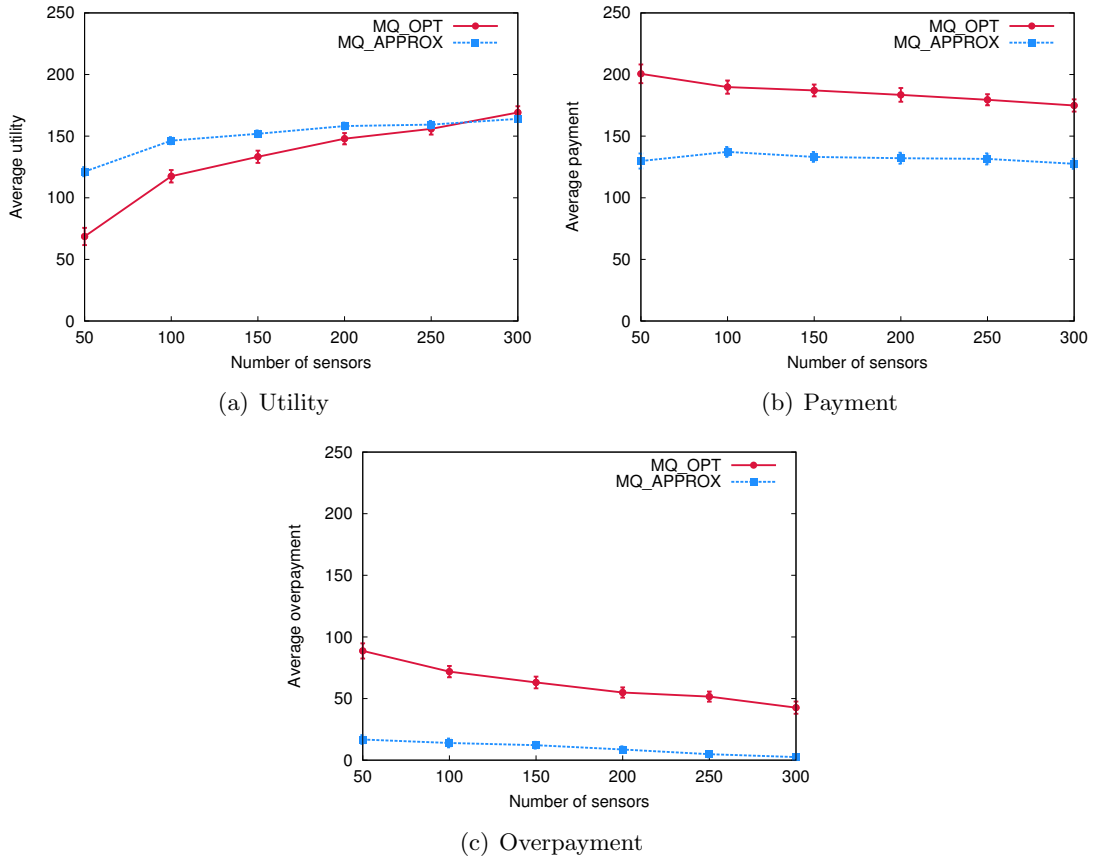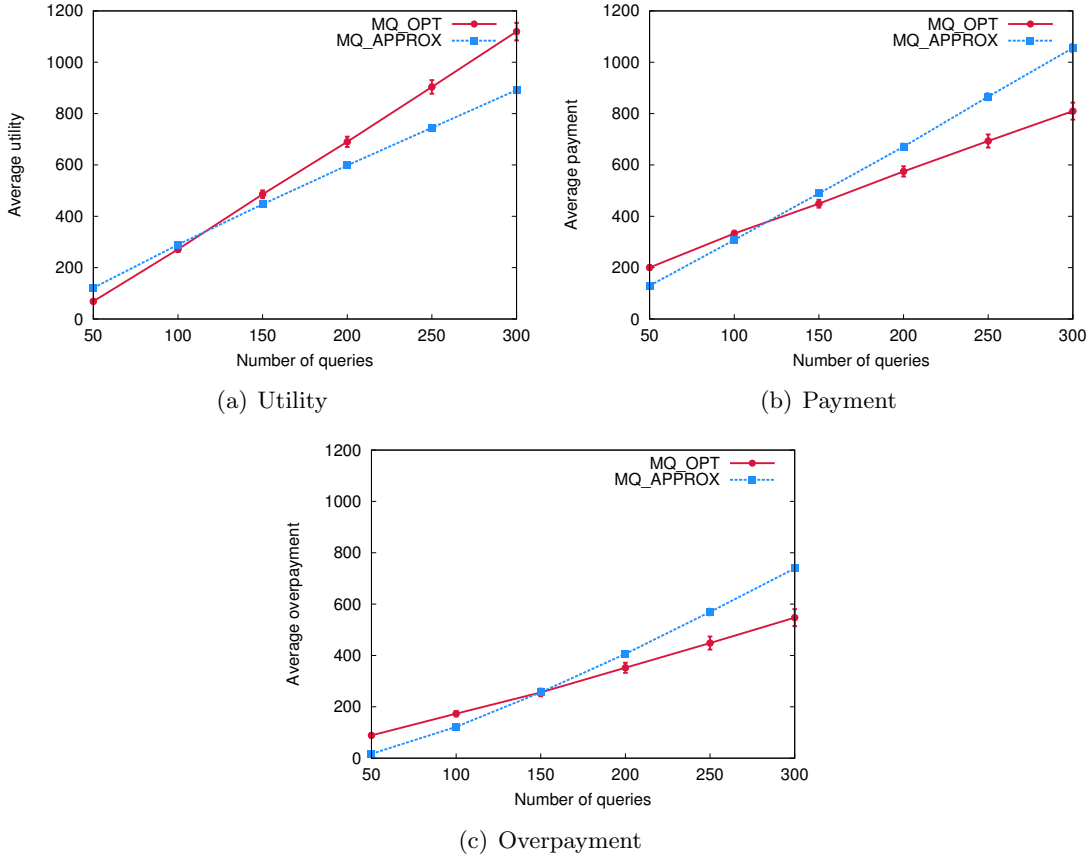
Figure 4.2: **(a)** Average utility, **(b)** Average payment, **(c)** Average overpayment, by MQ_{OPT} and MQ_{APPROX} for different number of queries. 50 sensors exist in the simulation region. MQ_{OPT} achieves more utility and it pays less as we increase the number of queries.

when agents reveal their exact location as in MQ_{OPT} (No obfuscation).

We consider three levels of privacy sensitivity for privacy conscious agents, namely, *low*, *moderate*, and *high*. The agents are split in three groups of equal size. One sensitivity level is assigned to each group. The initial size of the obfuscated region (i.e., the highest obfuscation level) is 4, 6, and 9 cells respectively for low, moderate, and high sensitivity levels. For computational reasons, we restrict the agents to report only rectangular regions. The initial obfuscated region is selected randomly. However, it contains the agent's location. For each agent i a base cost c_i^b is randomly considered from the interval $[5, 15]$. When there is no obfuscation, $c_i = c_i^b$. With privacy conscious agents, the cost corresponding to the highest obfuscation level is calculated as $c_i^h = \alpha c_i^b$, where $\alpha = \frac{1}{3}, \frac{1}{4}, \frac{1}{5}$ respectively for low, moderate, and high sensitivity levels. The cost of the lowest obfuscation level (i.e., no obfuscation) is computed as $c_i^l = 2c_i^h$. The cost linearly increases between these two values as the obfuscation level decreases. Therefore, the following cost function is assigned to agent i:

$$c_i(\gamma) = \frac{c_i^h - c_i^l}{|R_i^0|}\gamma + c_i^l. \tag{4.14}$$

(a) Utility   (b) Payment

Figure 4.3: **(a)** Average utility, **(b)** Average payment, by $\boldsymbol{PRIV_{TRADE}}$ with different agent privacy settings for different number of sensors (agents). 50 queries exist in each time slot.

The expected valuation of a measurement from agent i for queries at location l when the size of its reported obfuscated region is γ, is given by the following function:
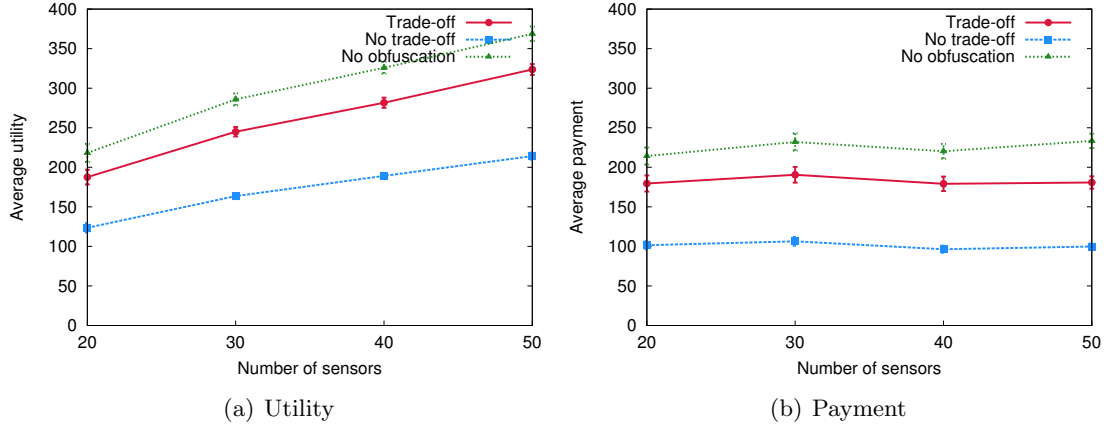
$$\tilde{v}_{Q^l}(s_i, \gamma, R_i^0) = \left(1 + \frac{|R_i^0| - \gamma}{|R_i^0| - 1}\right) \sum_{q \in Q^l} \sum_{r \in R_i^0} P(r|R_i^0)\bar{v}_q(s_i, r), \tag{4.15}$$

where $P(r|R_i^0)$ is the probability of s_i being in cell r knowing that it is somewhere in region R_i^0, and $\bar{v}_q(s_i, r)$ gives the expected valuation of s_i assuming it is located in cell r.

We use $h(K^*, R_i, \hat{x}_i) = \tilde{v}_{Q^l}(s_i, |R_i|, R_i^0)$ assuming that reported values are truthful. Using this function, we ignore the value of $\hat{x}_i$ and work with the expected quality of sensor readings from i. If $\hat{x}_i$ is totally ignored, the mechanism cannot be guaranteed to be incentive compatible for reporting true measurements. In order to alleviate this in practice, we can either use a function that considers reported values, or indirectly take into account the reported values by using them to update the reputation of the agents.

Varying the number of agents. The average achieved utility and average payment by $\boldsymbol{PRIV_{TRADE}}$ as the number of agents increases from 20 to 50 is compared for three different settings. The results are shown in Figure 4.3(a) and 4.3(b). The number of queries is set to 50 for this experiment. When the agents are willing to trade their privacy for more benefit, a higher utility for the center is achieved compared to when the agents are privacy conscious but not willing to trade their privacy. The payment to the winners follows the same behavior. However, when the agents reveal their exact location, even higher utility is acquired and more payment is issued. This is expected, as knowing the exact locations results in higher valuations for sensor readings.

Varying the number of queries. Figure 4.4(a) and 4.4(b) show the average achieved utility and average payment by $\boldsymbol{PRIV_{TRADE}}$ as the number of queries increases from 20 to 50. The number of agents is set to 50. Similarly to the experiment for varying number of agents, higher utility is achieved when the agents are privacy oblivious. When the agents trade their privacy, the utility of center is higher than when they
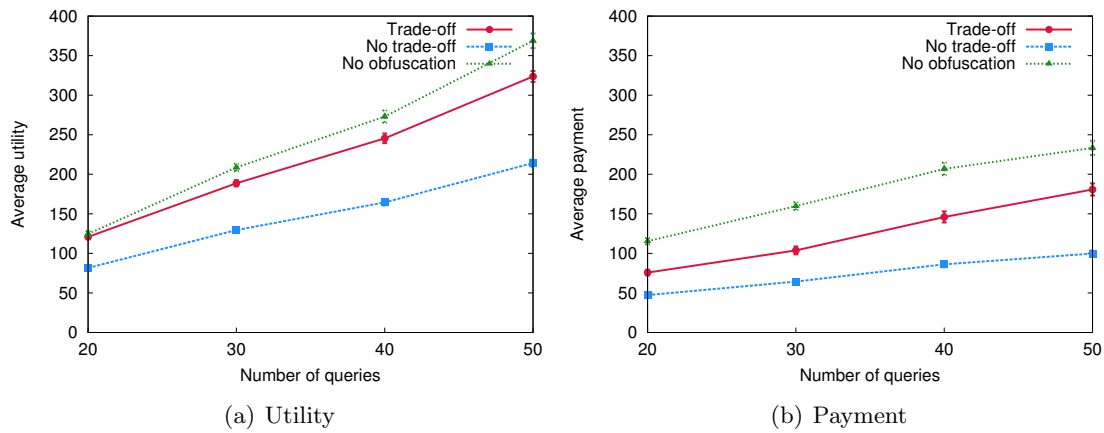
Figure 4.4: **(a)** Average utility, **(b)** Average payment, by $PRIV_{TRADE}$ with different agent privacy settings for different number of queries. 50 sensors exist in the simulation region.

are strict on revealing their private information. It can also be seen that as we increase the number of queries, the utility and payment increases. This is due to the fact that more queries result in firstly, more budget and secondly, more senor reading sharing.

4.5 Conclusion

In this chapter we proposed incentive compatible and individually rational mechanisms for eliciting truthful information from agents in participatory sensing. In our scenario, multiple applications request sensor measurements at several locations. The valuation functions for valuating the quality of measurements are provided by the applications. The agents are compensated for the measurements that they provide. We considered two cases where agents are privacy oblivious and privacy conscious. When agents are privacy oblivious, our mechanisms incentivize them to truthfully report their costs and measurements. We also provided an incentive compatible mechanism for enabling agents to trade privacy for monetary benefit.

Quality Assessment of Sensor Data Based on Frequent Patterns

5.1 Introduction

One of the most important tasks of sensor networks (SN), which include participatory sensing systems, is to detect occurrences of interesting events in the monitored environment (e.g., forest fire, chemical spill, leak of poisonous gases). Such events usually span some geographic region and involve simultaneous changes in values of several sensors.

However, data measured by SN is often affected by errors, i.e., noisy (incorrect) values resulting from faults of sensors caused by resource constraints (energy and bandwidth), calibration problems, or exposure to harsh environmental conditions (e.g., floods). Therefore, ensuring reliability of sensor data is a fundamental task for delivering actionable knowledge for a proper decision-making.

We investigate the problem of assessing quality of a sensor value (tested value) in the presence of events and errors when the sensors are stationary. A usual approach is to express the quality as a deviation of the tested value from a reference value (a normal value in SN data). Defining such a reference value is difficult for SN for the following reasons: (I) interaction between events and errors, where both of them may exhibit similar sensor values that can be considered as outliers with respect to the normal state of SN [143] and (II) differing characteristics of the monitored events, where they can have differing shapes and corresponding magnitude (e.g., *diffusion events*, where the magnitude of the observed event decreases with distance from its source [87]). Figure 5.1 presents a contour map of an example diffusion event. The event is additionally displaced to the right (east) of the source (e.g., a poisonous cloud that is displaced by westward wind).

State of the art approaches to quality assessment aim at defining the reference value in terms of a *context* consisting of values of spatially close sensors that are correlated with
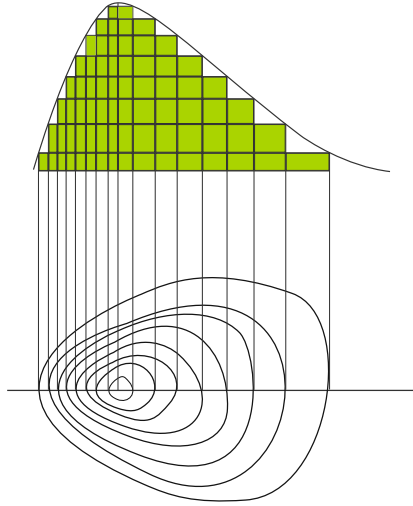
Figure 5.1: A contour map of a diffusion event (the magnitude of the observed event decreases with distance from its source). The event is additionally displaced to the left of the source (e.g., a poisonous cloud that is displaced by westward wind).

the tested value (e.g., [30, 76, 137]). This is based on the assumption that events, unlike errors, tend to involve spatio-temporal correlations with neighboring sensors, where the spatial correlations are stronger than the temporal [143]. Spatial correlations may result from dense sensor deployments that may cause spatially proximate sensor values to be correlated when they capture the same event [132]. Temporal correlations occur when consecutive sensor observations are related by referring to an evolution of the same event (e.g., tracking spread of a leak of a poisonous gas cloud over time). However, the state of the art approaches trade accuracy for simplicity and use a fixed context consisting of all values of a fixed neighborhood that occur simultaneously (e.g., all values within a circular neighborhood of radius r) and define the reference value as the average of the context [75, 76, 143]. Clearly, such a fixed context is only similar to the tested value if they both are subject to a single homogeneous event. Therefore, the state of the art approaches for quality assessment in SN suffer from the choice of inappropriate neighborhood and fail in many practical cases by under or overestimating the reference values [143].

Example 5.1. Figure 5.2 illustrates the problem of using fixed neighborhood for quality assessment for a set of six sensors $\mathcal{S} = \{s_1, s_2, \ldots, s_6\}$ that are located in a square region. The four squares present snapshots of the network corresponding to four different events (e.g., a movement of a pollution cloud or an oil spill region). The black dots represent the sensors, the dashed lines around the sensors represent their sensing ranges. The solid lines of differing width with associated values represent contour lines (magnitude of the event) of contour maps of the corresponding events. Let the task be to compute the quality of the current value of sensor s_3 (tested value). We use the *average-based method* (AB) that computes the quality as the difference between the tested value and the reference value defined as the average of the values of the sensors in a fixed circular
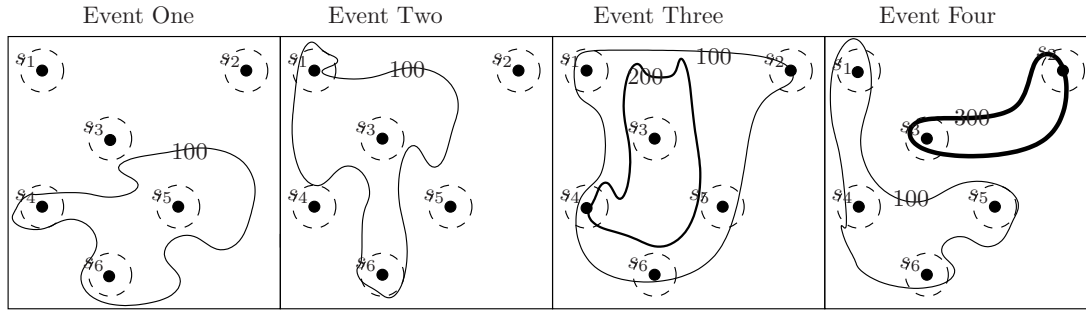
Figure 5.2: A set of six sensors $\{s_1, s_2, \ldots, s_6\}$ that are located in a square region. The four squares present snapshots of the sensor network corresponding to four differing events (e.g., a movement of a pollution cloud or an oil spill region). The black dots represent the sensors, the dashed lines around the sensors represent their sensing ranges. The solid lines of differing width with associated values represent contour lines of contour maps of the corresponding events.

neighborhood of radius r of sensor s_3. We assume that the neighborhood contains all other sensors from $\mathcal{S}$. Then the presence of the events will imply the following problems while using AB for computing quality of s_3.

Event One presents a case where s_3 lies outside of the event and three against two of its neighbors lie inside the event with value equal to 100. So the majority of neighbors are inside of the event and AB would give small quality to s_3.

Event Two presents a case where s_3 lies inside of the event with value equal to 100 and three against two of its neighbors lie outside of the event. So this case is complementary to the case in Event One.

Event Three presents a case where the event has two contour lines corresponding to values 200 and 100 respectively. s_3 and s_4 lie inside of the value 200 and the rest of the four sensors lie inside the value 100. Thus, the majority of sensors in region 100 would imply a low quality value of s_3 using AB.

Finally, *Event Four* presents a case where the event consists of two separate regions, one with value 100 and the other with value 300. Then clearly, four of the neighbors of s_3 lie inside the region with value 100. Thus, again AB would assume the value of s_3 deviates from the majority.

5.1.1 Overview of the Approach

We present the first pattern-wise method (PW) for quality assessment of sensor data that addresses the limitations of the state of the art approaches by departing from the idea of a fixed neighborhood. We take a realistic approach and assume that there does not exist a correct training set of a normal behavior for the neighborhood. Thus, the neighborhood can also contain faulty values. We proceed as follows: (I) we consider a *variable neighborhood* defined as an arbitrary subset of spatially close sensors in order to deal with arbitrary events, and (II) we define the context as a frequent spatial pattern, consisting of values of the variable neighborhood, that frequently co-occur with the tested

value in the stream of sensor values in order to deal with faulty neighborhood values and overlapping events.

We define the quality as *the belief (probability) that the tested value is correct given selected features of a frequent pattern consisting of the context and the tested value.*

We compute the quality of a given sensor value (tested value) as the output of the logistic regression, where the input variables consist of features of the pattern consisting of the context and the tested value. We use the logistic regression to combine the features such that the output is a probability value. Clearly, in our case, the output of the logistic regression corresponds to the probability of the binary random variable that the value is correct given the input variables [7]. We obtain the parameters of the logistic regression using a user evaluation, where we ask assessors to assess quality for a sample of the feature space that is subsequently used for learning the parameters.

Given the parameters of the logistic regression the algorithm proceeds as follows:

1. We use itemset mining to find, in the sensor data stream seen so far, a frequent correlated pattern, consisting of the tested value and a context, that maximizes the logistic function.

2. We compute quality using the following features of the pattern:

    (a) The relative frequency of the pattern.
    (b) The conditional probability of the tested value given the context.
    (c) The relative size of the pattern with respect to the number of streams.

The following example illustrates the quality computation process.

Example 5.2. Consider a sensor network consisting of a set of four sensors $\mathcal{S} = \{s_1, s_2, s_3, s_4\}$ monitoring daily occurrences of rainfall in corresponding four locations, where the value $s_i^{(t)} = 1/0$ denotes a presence/absence of rainfall on day t in sensor s_i. Consider the task to compute the quality of value $s_0^{(6)} = 1$. Figure 5.3 shows the corresponding sensor data streams.

Using frequent itemset mining we discover that, for $s_0^{(t)} = 1$, $\left[s_1^{(t)} = 0, s_3^{(t)} = 1\right]^T$ is the frequent correlated context (the variable neighborhood is $\{s_1, s_3\}$) that maximizes the logistic function given the parameters, where:

1. the relative frequency of the pattern $y_1 = \frac{2}{6}$.

2. the conditional probability

$$y_2 = P\left(s_0^{(t)} = 1 | s_1^{(t)} = 0, s_3^{(t)} = 1\right) = \frac{2}{3}.$$

3. the relative size of the pattern $y_3 = \frac{3}{4}$.

Given the parameters of the logistic regression model $[\beta_0 = -6, \beta_1 = 5, \beta_2 = 6, \beta_3 = 3]$ we compute the quality as $\mathcal{T}_0^{(6)} = \frac{1}{1+e^{-(\beta_0+\beta_1*y_1+\beta_2*y_2+\beta_3*y_3)}} = 0.87.$
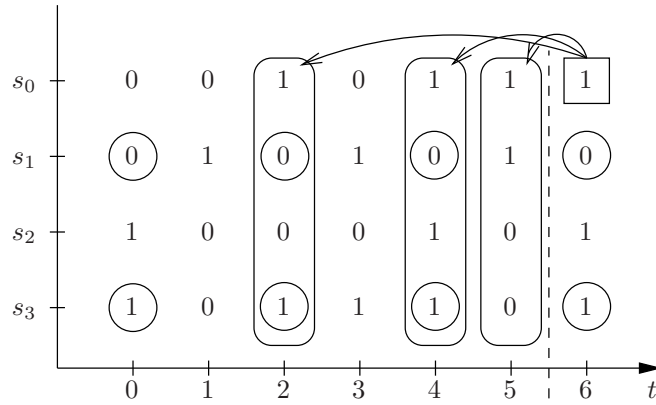
Figure 5.3: Quality computation for sensor value $s_0^{(6)} = 1$ (in the square) in the sensor multi-stream $\mathcal{S} = \{s_1, s_2, s_3, s_4\}$, where $s_i^{(t)} = 1/0$ means an occurrence/absence of rainfall on day t in sensor s_i. The circles around some values denote the occurrences of the frequent correlated context of the value in the square. The vertical dashed line separates processed (to the left) from the unprocessed multi-values. The backward arrows from the value in the square to the previous occurrences of the same value point to the projected multi-stream (in rounded rectangles).

Thus, the computed quality value given the values of the features $[y_1, y_2, y_3]$ reflects user beliefs, captured in the user evaluation, that the value is correct given the features of the frequent pattern. Please note that in most practical applications, given a sensor data stream, instead of computing quality for a single value, we will be interested in obtaining a corresponding *quality stream*.

Clearly, the appeal of using the frequent pattern approach to finding the context is as follows: (I) the frequent patterns capture differing cases of recurrent events (e.g., recurrent rainfall patterns associated with yearly cycle of recurring configurations of atmospheric fronts [47, 48]) as well as a "recurring normal values" of SN, and (II) frequent patterns filter out noisy values. Furthermore, note that the method also takes an advantage of temporal correlations implicitly by discovering the frequent patterns in the segment of the stream that precedes the tested value.

5.1.2 Motivating Application

The pattern-wise quality assessment of sensor data was inspired by project OpenIoT [1] as part of the corresponding Quality-Module (QM). OpenIoT develops an open source cloud-enabled middle-ware for Internet of Things (IoT). Figure 5.4 presents the main components of OpenIoT architecture that are as follows:

1. *GSN (Global Sensor Networks)*: is a sensor middle-ware that obtains sensor data streams from sensors [5].

2. *LSM (Linked Stream Middleware)*: is a cloud databases that stores sensor data streams from GSN [72].
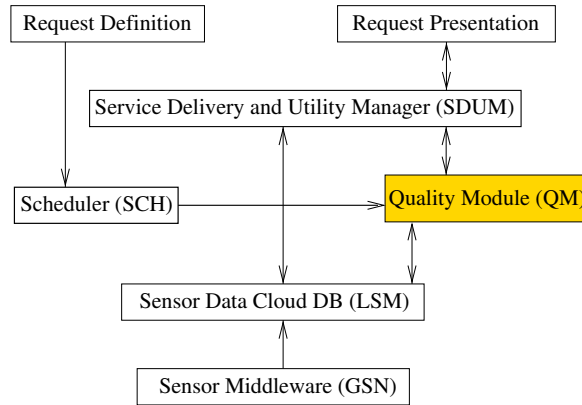
Figure 5.4: The pattern-wise quality assessment approach as part of the quality module in the OpenIoT platform.

3. *Scheduler* (SCH): processes all the requests (queries) and ensures their proper access to the resources (e.g., data streams) that they require.

4. *Service Delivery and Utility Manager* (SDUM): performs two tasks: (I) it combines data streams in order to deliver the requested services, and (II) it acts as a service metering facility, which keeps track of utility metrics for each individual service.

5. *Request definition*: provides a graphical user interface for on-the-fly specification of service requests.

6. *Request presentation*: provides a graphical user interface that enables visualization of the outputs of services.

QM is an independent module in OpenIoT. It computes quality in a *centralized way* by obtaining the sensor data streams from LSM and storing the corresponding quality streams back in LSM. There are the following ways of computing the quality stream given available sensor data streams in LSM:

1. On-line (immediate): while storing the sensor data streams to LSM. The disadvantage of this approach is a heavy overloading of QM, LSM and the communication link between them.

2. Off-line on demand: when a query needs recent quality streams. This approach has the same disadvantage as the on-line approach. Moreover, quality computation in this case is a blocking operation. The advantage of this approach is that LSM is not populated with quality streams that may never be used.

3. Off-line periodically (deferred): periodically after the sensor data streams have been stored in LSM.

We adopt the off-line solution combined with caching mechanism to optimize the computational and storage resources. QM obtains the data from LSM and periodically outputs the corresponding quality streams back to LSM (stored in a separate entity that

references the corresponding sensor data stream). SCH and SDUM communicates with QM to process queries. In particular, SCH may trigger an on-demand computation of a quality stream if this is necessary for a given query (e.g., the query specifies a minimum quality threshold for a sensor data stream from a given area), while SDUM monitors the performance of QM and triggers periodic computation of quality streams.

5.1.3 Contributions

Our contributions in this chapter are are the following: (I) we present the first method for quality assessment of sensor data that addresses the limitations of the state of the art approaches by departing from the idea of a fixed circular neighborhood. Our method is the first pattern-wise method that defines the context as a spatial frequent pattern consisting of values of the variable neighborhood that frequently co-occurs with the tested value in the stream of sensor values; and (II) we use the logistic regression to define quality in terms of parameters obtained from a user evaluation.

The rest of this chapter is organized as follows. Section 5.2 presents the theoretical foundations of our approach. Section 5.3 presents the pattern-wise solution. In Section 5.4 we present experiments for evaluating the proposed approach. We review the related work in Section 5.5. Finally we conclude the chapter in Section 5.6.

5.2 Theoretical Foundations

In this section we present the notation and review some concepts that are necessary in order to explain our framework.

5.2.1 Notation

We use subscript i to refer to the i-th sensor data stream. We assume that the streams are quantized and $\mathcal{A}_i = \{a_{i,1}, a_{i,2}, \ldots, a_{i,m_i}\}$ is an alphabet in stream i. $\mathcal{S} = \{s_1, s_2, \ldots, s_{|\mathcal{S}|}\}$ is a sensor *multi-stream* defined as a set of input streams of length n. The i-th stream (i-th attribute sequence) is defined as $s_i = \left[s_i^{(1)}, s_i^{(2)}, \ldots, s_i^{(n)}\right]$. Every *stream tuple* (stream element) has three attributes: (I) timestamp $s_i^{(t)}.timestamp = t$, where $t \in \{1, 2, \ldots\}$; (II) stream identifier $s_i^{(t)}.stream = i$; and (III) value denoted $s_i^{(t)}.value$, where $s_i^{(t)}.value \in \mathcal{A}_i$. For simplicity we just use $s_i^{(t)}$ to mean $s_i^{(t)}.value$. $\overline{s_i} = \{s_j | i \neq j\}$ is the set of complementary streams of stream s_i. $\mathcal{S}^{(t)} = \left[s_1^{(t)}, s_2^{(t)}, s_3^{(t)}\right]^T$ is a *multi-value* that is a simultaneous occurrence of symbols $s_1^{(t)}, s_2^{(t)}, s_3^{(t)}$ for a given time point t. $[e_1, e_2, \ldots, e_m]$ is a sequence of elements. $\{e_1, e_2, \ldots, e_m\}$ is a set of elements. $\mathcal{T}_k = \left[\mathcal{T}_k^{(t_s)}, \mathcal{T}_k^{(t_s+1)}, \ldots, \mathcal{T}_k^{(t_e)}\right]$ is the quality stream of the k-th sensor for the time window $[t_s, t_e]$, where t_s is the start time and t_e is the end time. $\mathcal{T}_k^{(t)}$ is the quality score for sensor k at time t, where $\mathcal{T}_k^{(t)} \in [0, 1]$, and 0 and 1 are the lowest and the highest quality scores.

5.2.2 System Model

Each sensor can sense the value of the phenomenon in its *sensing range* (measuring range). The sensing range of a sensor s_i is a circle centered at itself with radius r_i [140]. An *event* $\mathcal{E}$ is a subset of $\mathcal{R}^2$ such that readings of the sensors in $\mathcal{E}$ are different from the sensors that are not in $\mathcal{E}$ [30]. A faulty sensor can be considered as a special event which contains only one sensor. $\mathcal{S}$ is the set of all sensors. We also use $\mathcal{S}$ to denote the set of corresponding sensor data streams (multi-stream).

$\mathcal{N}_i$ is a neighborhood of a sensor s_i defined as a bounded closed set of $\mathcal{R}^2$ that contains sensor s_i and some number of other sensors. As an example of $\mathcal{N}_i$ consider a closed disk centered at s_i with radius r. We can distinguish two types of events: atomic events and composite events. An *atomic event* can be detected merely based on the observation of one attribute. For example, if the sensed temperature value exceeds a predefined threshold, an atomic event of "high temperature" is detected. A *composite event* is the combination of different atomic events. For example, the composite event fire may be defined as the combination of the temperature and light. The composite event fire occurs only when both the temperature and the light exceed some predefined thresholds. For clarity of the presentation we consider only atomic spatial events in this chapter. We do not make any further assumption on the structure of the network.

5.2.3 Itemset Mining

Let $\mathcal{A} = \{a_1, a_2, \ldots, a_{|\mathcal{A}|}\}$ be a set of items (alphabet). A subset $\mathcal{I} \subseteq \mathcal{A}$, where $\mathcal{I} = \{a_1, a_2, \ldots, a_{|\mathcal{I}|}\}$ is called an *itemset* or *element* and is also denoted by $(a_1, a_2, \ldots, a_{|\mathcal{I}|})$, where $|\mathcal{I}|$ denotes the size of the set. Thus, $\mathcal{I}$ is a *subitemset* of $\mathcal{A}$ and $\mathcal{A}$ is the *superitemset* of $\mathcal{I}$. Given a *collection of itemsets* $\mathcal{D} = \{\mathcal{I}^{(1)}, \mathcal{I}^{(2)}, \ldots, \mathcal{I}^{(|\mathcal{D}|)}\}$ (a multiset of sequences) the *support* (frequency) of an itemsets $\mathcal{I}$, denoted by $sup_\mathcal{D}(\mathcal{I})$, is defined as the number of itemsets $\mathcal{I}^{(i)} \in \mathcal{D}$ that contain $\mathcal{I}$ as a subset. The *relative support* (relative frequency) $rsup_\mathcal{D}(\mathcal{I}) = \frac{sup_\mathcal{D}(\mathcal{I})}{|\mathcal{D}|}$ is the fraction of itemsets that contain $\mathcal{I}$ as a subset.

Given a relative support threshold $minRelSup$ an itemset $\mathcal{I}$ is called a *frequent itemset* if $rsup_\mathcal{D}(\mathcal{I}) \geq minRelSup$. The problem of frequent itemset mining is to find all frequent itemsets in $\mathcal{D}$ given $minRelSup$. The support has the *downward-closure property* (also called *apriori* or *anti-monotonic* property), meaning that $sup_\mathcal{D}(\mathcal{I}) \geq sup_\mathcal{D}(\mathcal{I}')$ if and only if $\mathcal{I} \subseteq \mathcal{I}'$. Thus, for a frequent itemset, all its subsets are also frequent and thus for an infrequent itemset, all its supersets must also be infrequent. An itemset $\mathcal{I}$ is called a *frequent closed itemset (FCI)* if none of its frequent superitemsets has the same support. Thus, mining closed itemset reduces the number of discovered patterns and provides a more compact representation. Several efficient algorithms, such as [80, 101], exist for finding frequent closed itemsets.

Table 5.1 presents an example collection of itemsets, where for $minRelSup = 0.5$, $\mathcal{I} = (a_1, a_2)$ is a frequent itemset, where $rsup_\mathcal{D}(\mathcal{I}) = 0.5$ and it is contained in itemsets 0 and 3.

id	itemsets			
	a_1	a_2	a_3	a_4
0	1	1	0	0
1	0	1	1	0
2	0	0	0	1
3	1	1	1	0

Table 5.1: A collection of itemsets, where 1/0 means the presence/absence of items.

Given an itemset $\mathcal{I}$, an *itemset rule* or *itemset association rule* is defined as an implication of the form $X \Rightarrow Y$ where $X, Y \subseteq \mathcal{I}$ and $X \cap Y = \emptyset$. The sets of itemsets X and Y are called *antecedent* (left-hand-side or LHS) and *consequent* (right-hand-side or RHS) of the rule respectively. The confidence (conditional probability) of a rule is defined as

$$conf_\mathcal{D}(X \Rightarrow Y) = \frac{sup_\mathcal{D}(X \cup Y)}{sup_\mathcal{D}(X)} = P(Y|X),$$

where $X \cup Y$ means that both X and Y are present, i.e., $sup_\mathcal{D}(X \cup Y) = sup_\mathcal{D}(\mathcal{I})$. For example, in Figure 5.3 rule $(s_3, s_2) \Rightarrow s_0$ has confidence equal to $\frac{2}{3}$.

Although mining closed itemsets reduces the number of discovered patterns, that number may still be too large for an appropriately low value of $minRelSup$. In general given two itemsets X and Y there are the following three correlation relationships between them:

1. $P(Y|X) = P(Y)$, then Y and X are independent

2. $P(Y|X) > P(Y)$, then Y is positively dependent on X, and $X \Rightarrow Y$ is a *positive association rule*

3. $P(Y|X) < P(Y)$, then Y is negatively dependent on X and $X \Rightarrow \neg Y$ is a *negative association rule* (or $\neg Y$ is positively dependent on X)

A high value of $P(Y|X)$ alone is not enough to determine significance of $X \Rightarrow Y$ because $P(X|Y)$ can be small. We can fix the problem by requiring that $P(X|Y)$ is comparable to $P(Y|X)$. This observation leads to *all-confidence* rule-wise significance measure [98] defined as follows:

$$allConfidence(X \Rightarrow Y) = \min\{P(Y|X), P(X|Y)\}. \tag{5.1}$$

Thus, (5.1) leverages the rank of rules where the antecedent and consequent occur exclusively together.

All-confidence can be generalized to itemset-wise significance measure as follows

$$allConfidence(\mathcal{I}) = \min_{a_i \in \mathcal{I}} \left\{ \frac{sup(\mathcal{I})}{sup(a_i)} \right\}, \tag{5.2}$$

where the right hand side of (5.2) computes the confidence of the least favorable rule (when $|X| = 1$). Thus (5.2) leverages the rank of items that frequently co-occur. Clearly,

all-confidence has the downward-closure property meaning that $allConfidence(\mathcal{I}) \geq allConfidence(\mathcal{I}')$ if and only if $\mathcal{I} \subseteq \mathcal{I}'$. Thus, for a significant itemset, all its subsets are also significant and thus for an insignificant itemset, all its supersets must also be insignificant.

5.2.4 Average-based Quality Model (AB)

In the baseline solution ([75, 76]) the quality score of value $s_t^{(k)}$ is expressed as a *p-value* as follows:

$$\mathcal{T}_k^{(t)} = P\left(Z > Z\left(s_k^{(t)}\right)\right),$$

where:

$$Z\left(s_k^{(t)}\right) = \frac{\sqrt{n}\left(s_k^{(t)} - \mathbf{E}\left(\mathcal{N}_k^{(t)}\right)\right)}{\sqrt{\mathbf{Var}\left(\mathcal{N}_k^{(t)}\right)}},$$

$$\mathbf{E}\left(\mathcal{N}_k^{(t)}\right) = \frac{1}{\left|\mathcal{N}_k^{(t)}\right|} \sum_{value \in \mathcal{N}_k^{(t)}} value$$

and

$$\mathbf{Var}\left(\mathcal{N}_k^{(t)}\right) = \frac{\sum_{value \in \mathcal{N}_k^{(t)}} \left(value - \mathbf{E}\left(\mathcal{N}_k^{(t)}\right)\right)^2}{\left(\left|\mathcal{N}_k^{(t)}\right| - 1\right)},$$

where $\mathcal{N}_k^{(t)}$ is the collection of values (context) of a fixed circular neighborhood of sensor s_k of radius r.

5.2.5 Problem Definition

The general problem of quality assessment of sensor data stream s_k is to compute the quality sequence (stream) $\mathcal{T}_k = \left[\mathcal{T}_k^{(t_s)}, \mathcal{T}_k^{(t_s+1)}, \ldots, \mathcal{T}_k^{(t_e)}\right]$, where the following input is given:

- an input collection of sensor data streams $\mathcal{S} = \left\{s_1, s_2, \ldots, s_{|\mathcal{S}|}\right\}$, where $s_i = \left[s_i^{(t_s)}, s_i^{(t_s+1)}, \ldots, s_i^{(t_e)}\right]$

- t_s and t_e are the start and end timestamps of the multi-stream, which are not necessarily the same for all streams

- sensor identifier k

In the remainder of the chapter, we present our approach for calculating $\mathcal{T}_k^{(t)}$, the quality score of stream s_k at time t.

5.3 Pattern-wise Quality Assessment

In this section we provide details of the algorithm for pattern-wise quality assessment of sensor data (PW).

We aim at discovering the real correlations between neighbors and the tested value. For clarity of the presentation we assume that all the streams are over the same discrete alphabet $\mathcal{A} = \{0, 1, |\mathcal{A}| - 1\}$. However, for the purpose of itemset mining, to comply with the set semantics we differentiate between symbols of $\mathcal{A}$ in different streams leading to stream-wise alphabets $\mathcal{A}_i = \{0_i, 1_i, (|\mathcal{A}_i| - 1)_i\}$. For example, given the current multi-value on the streams $\mathcal{S}^{(t)} = \left[s_0^{(t)} = 0, s_1^{(t)} = 1, s_2^{(t)} = 1, \ldots \right]^T$, we write it alternatively as $[0_0, 1_1, 1_2, \ldots]^T$. We represent the corresponding itemset as $\mathcal{I}^{(t)} = \{0_0, 1_1, 1_2, \ldots\}^T$, where $\overline{\mathcal{I}_i^{(t)}}$ is the set of values of the complementary streams of sensor s_i. Let $\mathcal{I}_i^* \subseteq \overline{\mathcal{I}_i^{(t)}}$ be a subset of correlated stream values.

The main idea of our approach of assessing quality of a value $s_k^{(t)}$ is to find the most correlated subset of values on other streams $\mathcal{I}_k^*$ (the context), as observed from the beginning of the stream, and express the quality in terms of conditional probability $P\left(s_k^{(t)}|\mathcal{I}_k^*\right)$. However, using $P\left(s_k^{(t)}|\mathcal{I}_k^*\right)$ alone, for computing quality, is not meaningful since it does not consider other important features of the pattern such as the relative support and the context length. Therefore, we consider quality in terms of the following features of the pattern from the $[0, 1]$ interval:

1. $y_1 = rsup\left(\left\{s_k^{(t)}, \mathcal{I}_k^*\right\}\right)$, the relative support, that gives more importance to more frequent itemsets.

2. $y_2 = P\left(s_k^{(t)}|\mathcal{I}_k^*\right)$, the conditional probability of the tested value given the context, that gives more importance to correlated items.

3. $y_3 = \frac{|\mathcal{I}_k^*| + 1}{|\mathcal{S}|}$, the relative pattern size with respect to the number of streams, that gives more importance to larger itemsets.

Given the feature vector $\mathbf{y} = [y_1, y_2, y_3]$, we combine the features using the logistic function, [7], as follows:

$$\mathcal{T}_k^{(t)}(\mathbf{y}) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 y_1 + \beta_2 y_2 + \beta_3 y_3)}}, \tag{5.3}$$

where $[\beta_0, \beta_1, \beta_2, \beta_3]$ are predefined optimal parameters of the logistic regression that are learned from a user evaluation. Given the parameters $[\beta_0, \beta_1, \beta_2, \beta_3]$ we clearly select an itemset with such features $\mathbf{y}$ that maximizes the following score

$$\beta(\mathbf{y}) = \beta_0 + \beta_1 y_1 + \beta_2 y_2 + \beta_3 y_3. \tag{5.4}$$

Given the following input:

- $minRelSup$ is the minimum relative support threshold for itemsets.

- $minAllConf$ is the minimum all-confidence threshold for itemsets.

- $[\beta_0, \beta_1, \beta_2, \beta_3]$ that is learned from a user evaluation.

- $s_k^{(t)}$ is a stream value in stream k at time point t.

The algorithm proceeds as follows:

1. Mine frequent closed itemsets, where:

   (a) Grow only itemsets for s_k equal to $s_k^{(t)}$ (a projection of s_k on $s_k^{(t)}$).

   (b) Grow only itemsets $\mathcal{I}_k$ if

   $$allConfidence\left(\left\{s_k^{(t)}, \mathcal{I}_k\right\}\right) > minAllConf.$$

2. Find an itemset $\mathcal{I}_k^*$ such that

$$\mathcal{I}_k^* = \arg\max_{\beta(\mathbf{y})}\left\{\mathcal{I}_k^* \subseteq \overline{\mathcal{I}_k^{(t)}}\right\}, \tag{5.5}$$

   where:

$$
\begin{aligned}
y_1 &= rsup\left(\left\{s_k^{(t)}, \mathcal{I}_k^*\right\}\right), y_1 > minRelSup,\\
y_2 &= P\left(s_k^{(t)}|\mathcal{I}_k^*\right) = \frac{sup\left(\left\{s_k^{(t)}, \mathcal{I}_k^*\right\}\right)}{sup\left(\mathcal{I}_k^*\right)},\\
y_3 &= \frac{|\mathcal{I}_k^*|+1}{|\mathcal{S}|},\\
\beta(\mathbf{y}) &= \beta_0 + \beta_1 y_1 + \beta_2 y_2 + \beta_3 y_3.
\end{aligned}
$$

3. Return $\mathcal{T}_k^{(t)} = \frac{1}{1+e^{-\beta(\mathbf{y})}}$.

Clearly, the time complexity of the algorithm is dominated by mining the frequent itemsets that can be computationally expensive at low support thresholds. Therefore, we apply several optimization techniques that substantially cut the search space for itemsets and make the method applicable for a large number of sensor data streams. In particular, we apply the following techniques:

1. We operate at relative high range of $minRelSup$ in order to ensure that the discovered context is believed to be reliable as indicated in the user evaluation.

2. We run the frequent pattern mining algorithm on the subset of the input sequence corresponding to a projection of the sequence on the tested value.

3. We filter out insignificant itemsets before we extend them to supersets using the downward-closure property of all-confidence.

4. We filter out itemset based on the rank found so far according to formula (5.4).

5.4 Experiments

We evaluated our pattern-wise quality assessment using both generated and real data. The experiments were performed in Java programming language on a 2.70GHz dual core i7-2620M CPU machine running Ubuntu with 8GB of memory.

5.4.1 Parameters of the Logistic Function

Since quality is a subjective concept, we determined $[\beta_0, \beta_1, \beta_2, \beta_3]$, the parameter vector of our logistic regression model, by conducting a user evaluation as follows: (I) we generated an appropriate sample of the feature space by creating 3000 feature vector reflecting combinations of values for y_1, y_2, y_3; (II) we asked four human assessors to assign a quality value to each feature vector; and (III) we averaged quality scores form the four human assessors to obtain a single quality score per feature vector.

Given the user evaluation, we learned the parameter vector using function *glm* from the free software environment for statistical computing and graphics R [2]. We obtained the following parameter vector $[\beta_0 = -6.86, \beta_1 = 6.64, \beta_2 = 5.68, \beta_3 = 0.157]$, that is used throughout the experiments.

Correctness and meaning of the learned parameters can be verified. Recall, that the quality value $\mathcal{T}(\mathbf{y})$ corresponds to the probability that the output of the logistic regression is one given the input parameters $\mathbf{y}$ [7]. Then clearly the odds of $\mathcal{T}(\mathbf{y}) = 1$ is equal to

$$\frac{\mathcal{T}(\mathbf{y})}{1 - \mathcal{T}(\mathbf{y})} = e^{\beta(\mathbf{y})}, \tag{5.6}$$

and the *odds ratio (OR)* between odds when $y_1 = a + \Delta$ and odds when $y_1 = a$ for fixed y_2 and y_3 can be expressed as follows:

$$OR = \frac{\text{odds of quality for } y_1 = a + \Delta}{\text{odds of quality for } y_1 = a}$$
$$= \frac{e^{\beta_0 + \beta_1(a+\Delta) + \beta_2 y_2 + \beta_3 y_3}}{e^{\beta_0 + \beta_1 a + \beta_2 y_2 + \beta_3 y_3}} = e^{\beta_1 \Delta}.$$

Thus, $e^{\beta_1 \Delta}$ is the change in the odds of quality when y_1 is increased by Δ and y_2 and y_3 are fixed. Clearly, using this reasoning we can make sure that for a given Δ the increase in OR is appropriate.

5.4.2 Methodology

Another implication of the fact that quality of a sensor value (in the presence of errors and events) is a subjective measure is the lack of a *baseline stream* with quality assessment (*ground truth*). To overcome this difficulty, given a stream s_k, we constructed a *generated baseline stream* $\hat{s}_k$ and a *generated baseline quality stream* $\widehat{\mathcal{T}}_k$ using the following approach:

1. We assumed that s_k is clean (either generated or real data stream).

2. We generated $\hat{s}_k$ by injecting errors and events into s_k.

3. We generated $\widehat{\mathcal{T}}_k$ by assigning a quality score for the data points of $\hat{s}_k$ given the full knowledge of the injected distortions.

In particular, we evaluated the quality of each data point $\hat{s}_k^{(t)}$, depending on whether $\hat{s}_k^{(t)}$ contains an error or event, as follows:

1. Clean: i.e., $s_k^{(t)} = \hat{s}_k^{(t)}$. Then clearly $\widehat{\mathcal{T}}_k^{(t)} = 1$.

2. Error: i.e., $s_k^{(t)} \neq \hat{s}_k^{(t)}$. Then $\widehat{\mathcal{T}}_k^{(t)}$ reflects the difference between $s_k^{(t)}$ and $\hat{s}_k^{(t)}$.

3. Event: i.e., $s_k^{(t)} \neq \hat{s}_k^{(t)}$. Then clearly $\widehat{\mathcal{T}}_k^{(t)} = 1$.

4. Error and event: then $\widehat{\mathcal{T}}_k^{(t)}$ reflects the difference between $s_k^{(t)}$ given the event and $\hat{s}_k^{(t)}$.

In cases 2 and 4 the contribution of the difference to the decrease in quality is based on a user evaluation, where the assessors were asked to evaluate the decrease in quality given the amount of noise injected into the values.

Given the generated baseline quality stream $\widehat{\mathcal{T}}_k$ we used the average-based method (AB) as a *baseline method* to compare with the pattern-wise method (PW). For this purpose we used RMSE (Root Mean Squared Error) expressed as follows:

$$RMSE\left(\mathcal{T}_k\right) = \sqrt{\frac{1}{n}\sum_{t=1}^{|\mathcal{T}_k|}\left(\widehat{\mathcal{T}}_k^{(t)} - \mathcal{T}_k^{(t)}\right)^2}, \tag{5.7}$$

where $\widehat{\mathcal{T}}_k$ is the generated ground truth quality stream and $\mathcal{T}_k$ is the tested quality stream (e.g., obtained from PW or AB), both referring to the same generated baseline stream $\hat{s}_k$.

The purpose of the simulations was to show superiority of PW over AB since PW uses a more reliable context to deal with errors and events. In all experiments with generated data $minAllConf = 0.25$ is used.

We considered two types of errors: *offset errors* and *variance degradation errors* [41], where (I) an offset error is generated by adding a random value from interval $[-2.5, 2.5]$ to a sensor value and (II) variance degradation error is generated, by sampling from $\mathcal{N}(0, 1)$ (i.e., normal distribution with mean 0 and variance 1) and adding it to a sensor value. For experiments with errors (faulty sensors), sensors are randomly selected to contain variance degradation or offset errors. Half of these sensors will contain variance degradation and the other half will contain offset faults. The total number of faulty sensors is specified in each experiment. A sensor selected as faulty remains faulty for a duration of 1 to 10 time units (randomly chosen), but with different amount of error at each time unit. Every 40 time units a new set of sensors is randomly selected as the set of faulty sensors.

We simulated events by randomly placing event sources in the experiment region. Each event e has a fixed duration τ_e and occurs every t_e time units. Each event source O_e emits a signal that is dispersed based on a two dimensional Gaussian function. Given
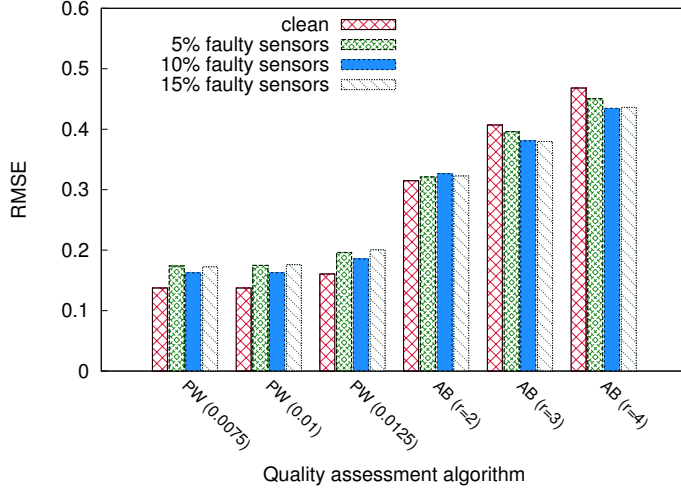
Figure 5.5: Root mean squared error of different PW and the baseline AB instances in generated streams. PW(x) means PW with $minRelSup = x$. PW results in more than three times lower RMSE compared to the baseline.

event e from an event source located at (x_e, y_e), which has started at time t_e^s, the event value at location (x, y) at time $t_e^s \leq t \leq t_e^s + \tau_e$ is given by:

$$v(x, y, t) = \alpha_{e,t} e^{\frac{(x_e - x)^2}{2\sigma_{x,t}^2} + \frac{(y_e - y)^2}{2\sigma_{y,t}^2}}, \tag{5.8}$$

where $\alpha_{e,t}$ is the magnitude of the event at time t, $\sigma_{x,t}$ and $\sigma_{y,t}$ are the dispersion parameters on the x and y axes at time t. The sensor located at (x, y) takes $v(x, y, t)$ as its value if $v(x, y, t) > \gamma_e$, where γ_e is the event impact threshold (i.e., the sensor cannot detect the event otherwise). α_e, σ_x, σ_y are the original values at event start time for magnitude, and dispersion parameters over x and y axes. Event propagation is modeled by increasing σ_x and σ_y as $t - t_e^s$ increases. Signal degradation is modeled by reducing α_e as $t - t_e^s$ increases. This model simulates diffusion events (e.g., a gas leakage) that happen in fixed intervals and are slowly dispersed around the event sources (e.g., leakage point) and vanish over time. If multiple events exist at the same time, sensors take the maximum value generated by the events at their locations.

5.4.3 Experiments with Generated Data

We evaluated the effectiveness of PW using 64 sensors in a region of 8×8 grids, where the generated sensor values were from a range $[10, 50]$ and the stream size was 10000 time units. quality values were computed for all sensor values in the stream segment $[9990, 10000]$.

5.4.3.1 Errors

In the first experiment we considered only errors (faulty sensors). PW was run with $minRelSup$ equal to 0.0075, 0.01, and 0.0125, respectively. AB was run with three neighborhood sizes $r = 2$, $r = 3$, and $r = 4$, that correspond to 12, 28, and 48 sensors,
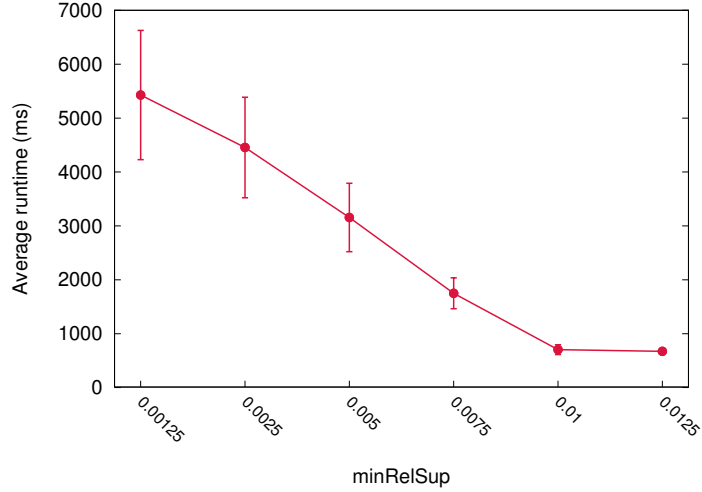
Figure 5.6: Average runtime of PW with different $minRelSup$ values in generated stream. 10% of sensors are faulty.

respectively. Clearly, PW considers all the 64 sensors for finding context. Figure 5.5 presents the results, where the percentage of faulty sensors was 0% (clean data), 5%, 10%, and 15%. PW significantly outperforms the baseline, where RMSE of PW is more than two times lower than that of AB with $r = 2$ and more than three times lower than that of AB with $r = 4$. AB with $r = 2$ performs better than AB with $r = 3$ and $r = 4$. The reason is that the probability of having multiple faulty sensors in a small neighborhood is low. Clearly, PW performs better in terms of RMSE as we decrease the $minRelSup$ parameter since PW is able to find a longer context. The reduction in error is achieved at the cost of higher execution time as Figure 5.6 illustrates.

5.4.3.2 Events

In the second experiment we calculated the quality for the stream segment $[9900, 10000]$ of the sensor located in grid $(3, 3)$ in the presence of only events. Four event sources $O_{e_1}, .., O_{e_4}$ were placed in the simulation region that generate non overlapping events every 50 time units. We set $\tau_{e_1} = 5$, $\tau_{e_2} = 10$, $\tau_{e_3} = 12$, and $\tau_{e_4} = 12$ as duration of events, the magnitudes of the events are α_e and dispersion parameters σ_x and σ_y. The event impact threshold γ_e, was set to 12 for all events. With respect to event e_1, as an example, the number of sensors affected by the event is 3, 3, 7, 11, 17 at $t = t_{e_1}^s, (t_{e_1}^s + 1), \ldots, (t_{e_1}^s + \tau_{e_1} - 1)$, respectively. Figure 5.7 presents the results, where PW outperforms AB. In particular, when the sensor value is affected by events (denoted by disconnected horizontal lines), PW assigns a high quality while AB treats events as errors and assigned a low quality. This confirms our intuition since PW uses a more reliable context than AB.
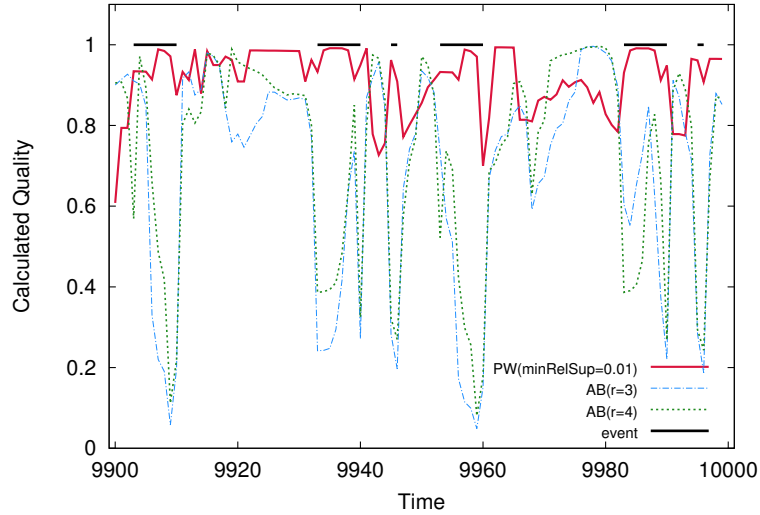
Figure 5.7: Quality calculated for sensor values from the sensor located at $(3, 3)$ by PW and AB in the presence of events in generated streams. Horizontal discontinuous lines indicate the presence of events. No noise is introduced in the sensor values. Contrary to the baseline, PW assigns high quality to sensor values when events occur.

5.4.3.3 Errors and events

In the third experiment, we combined the error and event models as defined for the first and the second experiment respectively. Figure 5.8 presents the results, where PW performs more than two times better than the baseline with or without errors in data. The reason is that introducing events reduces the strong spatial correlation between neighboring sensors, which is the essential requirement of AB. RMSE for AB is almost unaffected by increasing the noise level. This can be explained by the fact that AB already gives a low quality to the values affected by the events.

5.4.4 Experiments with Real Data

We used temperature readings from a collection of 64 sensors deployed in Switzerland as part of *SwissEx* project [4]. The dataset contains 15000 records sampled every 30 minutes. Missing values were replaced by interpolation using available values of the sensors. In this experiment we assumed that the data was clean in order to be able to use the generated ground truth quality.

In this experiment we considered the influence of both errors and events. We set $misRelSup = 0.0075$ and $minAllConf = 0.1$. One simulated event source with duration of 5 time units was placed into the region, which generated a new event every 50 time units. Figure 5.9 presents the results, where PW performs two times better than AB in terms of RMSE. PW results in slightly higher RMSE as we increase the number of faulty sensors. This is due to the fact that injecting noise in the data reduces the frequency of interesting patterns and leads to a drop in quality, which is desired for noisy tested values.
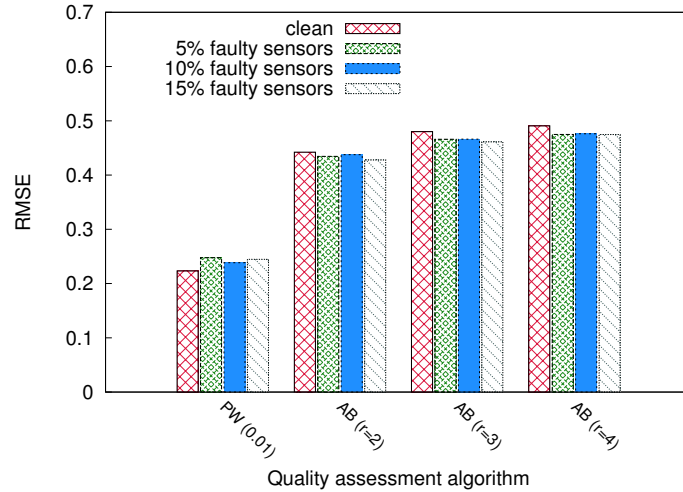
Figure 5.8: Root mean squared error of PW when simulated events and faults are introduced in generated streams, where $minRelSup = 0.01$. PW achieves two times lower error than the baseline.

5.5 Related Work

In Chapter 2 we provided an extensive review of the related work. We can distinguish our approach from the existing approaches for quality assessment and outlier detection in sensor networks as the following. (I) Contrary to the previous methods such as [30, 71, 75, 76, 96], which suffer from the choice of the fixed neighborhood, we do not restrict our method to using a predefined fixed neighborhood. In our approach, the sensors in a neighborhood are not necessarily spatially proximate. The neighborhood is dynamically determined based on the frequent patterns identified in the sensor readings over time. (II) While model-based methods such as [15], and classification-based approaches such as [36, 71, 96, 133] require the knowledge of the statistical model of the sensor readings, our pattern-wise method only works based on the value of the sensors and no background information is needed. (III) We do not make any assumption on the structure of the sensor network. The only input from the sensor network to the pattern-wise approach is the sensor data streams.

In [138] an algorithm for event detection that is based on contour map matching was presented. Thus, it converted the event detection problem into a pattern matching problem. The motivation was to address disadvantages of the threshold-based methods for event detection, where the threshold-based methods are based on the assumption that if sensor values exceed a certain (user defined) threshold then an event has occurred. It pointed out that such threshold, although simple, are inappropriate for the following reasons: (I) it is difficult to specify proper thresholds given differing environments to be monitored and application semantics and (II) events, where the magnitude of the observed event decreases with distance from its source (diffusion events) cannot be easily captures by discrete threshold values. The proposed method constructs and incrementally updates a number of contour maps that are used as building blocs for con-
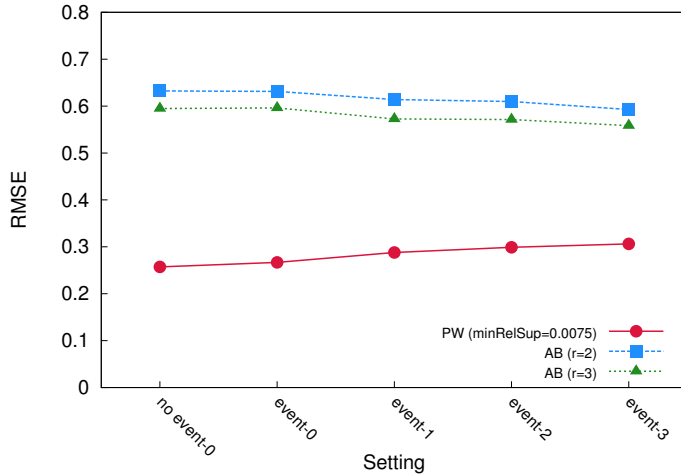
Figure 5.9: Root mean squared error of PW algorithm in presence of events and faults on real dataset. 'event-x' means that events are present and x signifies the number of faulty sensors. 'No event-0' means that no event exists in the dataset and no sensor is faulty.

structing spatio-temporal patterns exhibited in contour maps. The paper also identified three types of common events with respect to the shape of their contours in the map: pyramid, fault and island.

In [77] different optimization techniques for speeding up mining frequent value set in sensor networks were presented, using interval lists consisting of intervals during which the sensor assumes a given value. They coupled the idea of interval list with approximate itemset mining and derived an online algorithm for mining frequent itemsets from large a SN.

In [108] an approach for distributed mining of spatio-temporal event patterns in SN was presented. The algorithm proceeds as follows: (I) every sensor in the network continuously collects user defined events from neighboring sensors within a fixed distance and keeps a history of a fixed size of these events and (II) every sensor runs a mining algorithm for discovering patterns among these collected events. The main idea of the approach was to transmit to the sink only the compact patterns mined at sensors.

5.6 Conclusion

In this chapter we presented the first pattern-wise method for quality assessment of sensor data that addresses the limitations of the state of the art approaches by departing from the idea of a fixed neighborhood. Using frequent itemset mining, the method finds the values from multiple sensor data streams that frequently co-occur with the tested value. The logistic regression function was used to produce the quality score of the tested value given specific features of the frequent itemset. The performance of the pattern-wise approach regarding quality score computation error was compared to the performance of the common average-based approach. Experimental results confirmed superiority of the proposed method over the average-based approach.

Conclusion and Future Directions

6.1 Conclusion

In this thesis we considered some important data management problems in participatory sensing systems and proposed efficient solutions for those problems. In particular, we looked at the problem of efficient data acquisition in participatory sensing, where several factors must be considered concurrently while collecting data from participants and answering user queries. Incentivizing participants to truthfully provide their private cost information and measurements was another important problem that we considered in this thesis. Finally, we proposed a novel approach towards assessing the quality of sensor readings based on frequent pattern mining methods.

We proposed a holistic data acquisition framework for participatory sensing environments, in which we incorporated the most important parameters pertinent to this paradigm, such as uncontrolled mobility, privacy, trust, costs, and utility. Based on the argument that in such systems, the type of applications and queries that are posed by the applications can be diverse, the proposed framework was designed to be as generic as possible. We formulated the problem of optimal multi-query data acquisition with the objective of maximizing the total utility for the applications. Since finding the optimal solution is computationally intractable in many cases, efficient heuristic algorithms were proposed to myopically maximize the total utility for some of the most important query types and their combinations. In particular, we proposed utility-driven data acquisition algorithms for point and aggregate queries, which are examples of one-shot queries, location and region monitoring queries, which are examples of continuous queries, and the combination of these individual query types.

The proposed utility-driven framework for data acquisition in participatory sensing would not be useful if the participants misreport their cost information and their measurements. In order to incentivize the participants to truthfully report their data, we designed incentive compatible and individually rational mechanisms for data acquisition

in participatory sensing as part of this thesis. The proposed mechanisms were designed for data acquisition for point queries with the objective of maximizing the utility of the center (or applications). We considered two cases, where the participants are privacy oblivious, i.e., they are willing to report their exact location, and where the participants are privacy conscious, i.e., they are not willing to reveal their exact locations. In case of privacy conscious participants, we proposed mechanisms for enabling them to trade their privacy for more monetary incentives.

Lastly, we presented the first pattern-wise method for quality assessment of sensor data that addresses the limitations of the state of the art approaches by departing from the idea of a fixed neighborhood. Using frequent itemset mining, the method finds the values from multiple sensor data streams that frequently co-occur with the tested value. The logistic regression function was used to generate a quality score for each sensor value, given carefully chosen features of their frequent itemset on other sensor data streams. Experimental results confirmed superiority of the proposed method over the commonly used average-based approach.

6.2   Future Directions

The work described in this thesis can be extended and enhanced in several different ways. We suggest the following research directions as future work.

6.2.1   Data Acquisition in Participatory Sensing

In relation to the proposed utility-driven approach for data acquisition in participatory sensing systems (Chapter 3):

- We can take advantage of the mobility knowledge of sensors having controlled or semi-controlled mobilities, such as the sensors mounted on public transport vehicles, whenever such sensors exists. The existence of such sensors can significantly help acquiring data for all query types, in particular for continuous queries, and therefore increase the utility of the system.

- Event detection queries are one important class of queries in participatory sensing with different requirements compared to monitoring queries. Efficient data acquisition mechanisms for event detection queries should be investigated as an extension to the presented data acquisition approach for monitoring queries.

- In the presented data acquisition algorithms, we assumed that a trust assessment or reputation management mechanism was in place that assigns trust values to the sensors. However, these mechanisms often have direct impact on data acquisition because they might require redundant measurements. Therefore, the proposed algorithms should be extended to account for the data requirements of trust or reputation management mechanisms.

- A common approach towards protecting location privacy of the participants is that they employ a privacy protection mechanism that obfuscates their locations. In the proposed approach, we assumed that the participants precisely report their location to the aggregator. In presence of this sort of privacy protection mechanism, the proposed approach needs to be extended to account for the obfuscated location reports.

- In scenarios where the participatory sensing system forms a distributed data collection and processing environment without a centralized entity, the proposed mechanisms must be adapted. Efficient solutions should be provided for distributed utility-driven sensor discovery, data acquisition, and query processing in participatory sensing.

6.2.2 Truthful Data Acquisition

With regard to the proposed mechanisms for truthful data elicitation in participatory sensing (Chapter 4):

- As an important extension to the proposed mechanisms, we should relax the assumption that the ground truth is always available. Using redundancy is a potential solution, which introduces interdependent valuations for agents. Another solution is to employ environmental models as the ground truth. Combining these two solutions would certainly yield better results.

- The presented mechanisms incentivize the participants to truthfully reveal their data by providing monetary incentives to them. However, there might exist some malicious participants who obtain more benefit by reporting falsified data. Therefore, it is important to combine the proposed approach with trust assessment or reputation mechanisms to identify malicious participants and to promote honest data reporting.

- Extending the proposed mechanisms for truthful data elicitation for answering other query types introduced in Chapter 3 is another important future work.

6.2.3 Quality Assessment of Sensor Data Streams

In relation to the proposed approach for quality assessment of sensor data streams (Chapter 5):

- Fast quality score computation in high rate sensor data streams is an essential requirement for applicability of the proposed frequent pattern-based approach. Therefore, the algorithm should be further optimized. The optimization can be done, for example, by applying techniques including: (I) incremental itemset mining, (II) approximate itemset mining, (III) segmentation of sensors and sub-sampling, and (IV) interval lists in the spirit of [77].

- The proposed approach for quality assessment assumes that the sensor are stationary. In participatory sensing, however, most of the sensors are mobile. Therefore, the proposed approach cannot be readily applied for quality assessment in participatory sensing and needs to be adapted to the case of mobile sensors.

- Inconsistencies in sensor data can be due to malicious acts of some users who obtain benefits by tampering with the sensor readings with the goal of breaking the trust assessment mechanism, e.g., by introducing artificial repeated patterns in the data. As a future research direction, the impact of this kind of malicious behavior and the countermeasures against it can be investigated.

Bibliography

[1] Openiot. Online: `http://www.openiot.eu`. Accessed: 2014-06-04. 87

[2] The r project for statistical computing. `http://www.r-project.org/`. Accessed: 2014-06-04. 95

[3] Safecast project. Online: `http://blog.safecast.org/`. Accessed: 2014-06-04. 3, 9

[4] Swiss experiment. `http://www.swiss-experiment.ch`. Accessed: 2014-06-04. 99

[5] K. Aberer, M. Hauswirth, and A. Salehi. A middleware for fast and flexible sensor network deployment. In *Proceedings of the 32Nd International Conference on Very Large Data Bases*, VLDB '06, pages 1199–1202. VLDB Endowment, 2006. 87

[6] K. Aberer, S. Sathe, D. Chakraborty, A. Martinoli, G. Barrenetxea, B. Faltings, and L. Thiele. Opensense: Open community driven sensing of environment. In *Proceedings of the ACM SIGSPATIAL International Workshop on GeoStreaming*, IWGS '10, pages 39–42, New York, NY, USA, 2010. ACM. 1, 9, 11

[7] A. Agresti. *An Introduction to Categorical Data Analysis (Wiley Series in Probability and Statistics)*. Wiley-Interscience, 2 edition, 2007. 86, 93, 95

[8] G.-S. Ahn, M. Musolesi, H. Lu, R. Olfati-Saber, and A. T. Campbell. Metrotrack: Predictive tracking of mobile events using mobile phones. In *Proceedings of the 6th IEEE International Conference on Distributed Computing in Sensor Systems*, DCOSS'10, pages 230–243, Berlin, Heidelberg, 2010. Springer-Verlag. 9

[9] M. Annavaram, N. Medvidovic, U. Mitra, S. S. Narayanan, G. Sukhatme, Z. Meng, S. Qiu, R. Kumar, G. Thatte, and D. Spruijt-Metz. Multimodal sensing for pediatric obesity applications. In *Proceedings of the International Workshop on Urban, Community, and Social Applications of Networked Sensing Systems (UrbanSense)*, pages 21–25, Raleigh, NC, Nov. 2008. 11

107

[10] R. Avnur and J. M. Hellerstein. Eddies: Continuously adaptive query processing. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, pages 261–272, New York, NY, USA, 2000. ACM. 22

[11] M. Babaioff and N. Nisan. Concurrent auctions across the supply chain. In *Proceedings of the 3rd ACM conference on Electronic Commerce*, EC '01, pages 1–10, New York, NY, USA, 2001. ACM. 70

[12] X. Bao and R. Roy Choudhury. Movi: Mobile phone based video highlights via collaborative sensing. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pages 357–370, New York, NY, USA, 2010. ACM. 9, 12

[13] A. Behzadan and A. Anpalagan. Optimization of multiple overlapping queries for energy efficient sensor communication. In *Communications (QBSC), 2010 25th Biennial Symposium on*, pages 181–186, May 2010. 23

[14] U. Berkeley/Nokia/NAVTEQ. Mobile millennium. Online: `http://traffic.berkeley.edu`. Accessed: 2014-06-04. 1, 10

[15] L. Bettencourt, A. Hagberg, and L. Larkey. Separating the wheat from the chaff: Practical anomaly detection schemes in ecological applications of distributed sensor networks. In J. Aspnes, C. Scheideler, A. Arora, and S. Madden, editors, *Distributed Computing in Sensor Systems*, volume 4549 of *Lecture Notes in Computer Science*, pages 223–239. Springer Berlin Heidelberg, 2007. 33, 100

[16] F. Bian, D. Kempe, and R. Govindan. Utility based sensor selection. In *Proceedings of the 5th International Conference on Information Processing in Sensor Networks*, IPSN '06, pages 11–18, New York, NY, USA, 2006. ACM. 16, 35, 61

[17] N. Buchbinder, M. Feldman, J. Naor, and R. Schwartz. A tight linear time (1/2)-approximation for unconstrained submodular maximization. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 649–658, Oct 2012. 44, 70

[18] J. A. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava. Participatory sensing. 2006. 1, 2, 8

[19] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, and R. A. Peterson. People-centric urban sensing. In *Proceedings of the 2Nd Annual International Workshop on Wireless Internet*, WICON '06, New York, NY, USA, 2006. ACM. 1, 2, 8

[20] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, R. A. Peterson, H. Lu, X. Zheng, M. Musolesi, K. Fodor, and G.-S. Ahn. The rise of people-centric sensing. *IEEE Internet Computing*, 12(4):12–21, July 2008. 2

[21] J. Carrapetta, N. Youdale, A. Chow, and V. Sivaraman. Haze watch project. Online: `http://www.pollution.ee.unsw.edu.au/`, 2010. Accessed: 2014-06-04. 2, 9

[22] A. K. Chorppath and T. Alpcan. Trading privacy with incentives in mobile commerce: A game theoretic approach. *Pervasive and Mobile Computing*, 9(4):598 – 612, 2013. 28

[23] D. Christin, A. Reinhardt, S. S. Kanhere, and M. Hollick. A survey on privacy in mobile participatory sensing applications. *Journal of Systems and Software*, 84(11):1928–1946, Nov. 2011. 8

[24] S. Consolvo, D. W. McDonald, T. Toscos, M. Y. Chen, J. Froehlich, B. Harrison, P. Klasnja, A. LaMarca, L. LeGrand, R. Libby, I. Smith, and J. A. Landay. Activity sensing in the wild: A field trial of ubifit garden. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 1797–1806, New York, NY, USA, 2008. ACM. 1, 11, 12

[25] T. Das, P. Mohan, V. N. Padmanabhan, R. Ramjee, and A. Sharma. Prism: Platform for remote sensing using smartphones. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pages 63–76, New York, NY, USA, 2010. ACM. 10

[26] R. Dash, A. Rogers, N. Jennings, S. Reece, and S. Roberts. Constrained bandwidth allocation in multi-sensor information fusion: a mechanism design approach. In *Information Fusion, 2005 8th International Conference on*, volume 2, pages 8 pp.–, July 2005. 25

[27] L. Deng and L. P. Cox. Livecompare: Grocery bargain hunting through participatory sensing. In *Proceedings of the 10th Workshop on Mobile Computing Systems and Applications*, HotMobile '09, pages 4:1–4:6, New York, NY, USA, 2009. ACM. 13

[28] T. Denning, A. Andrew, R. Chaudhri, C. Hartung, J. Lester, G. Borriello, and G. Duncan. Balance: Towards a usable pervasive wellness application with accurate activity inference. In *Proceedings of the 10th Workshop on Mobile Computing Systems and Applications*, HotMobile '09, pages 5:1–5:6, New York, NY, USA, 2009. ACM. 11

[29] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*, VLDB '04, pages 588–599. VLDB Endowment, 2004. 19, 41

[30] M. Ding, D. Chen, K. Xing, and X. Cheng. Localized fault-tolerant event boundary detection in sensor networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 2, pages 902–913 vol. 2, March 2005. 31, 34, 84, 90, 100

[31] Y. F. Dong, S. Kanhere, C. T. Chou, and N. Bulusu. Automatic collection of fuel prices from a network of mobile cameras. In *Proceedings of the 4th IEEE International Conference on Distributed Computing in Sensor Systems*, DCOSS '08, pages 140–156, Berlin, Heidelberg, 2008. Springer-Verlag. 13

[32] P. Dutta, P. M. Aoki, N. Kumar, A. Mainwaring, C. Myers, W. Willett, and A. Woodruff. Common sense: Participatory urban sensing using a network of handheld air quality monitors. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys '09, pages 349–350, New York, NY, USA, 2009. ACM. 1, 9

[33] S. B. Eisenman and A. T. Campbell. Skiscape sensing. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, SenSys '06, pages 401–402, New York, NY, USA, 2006. ACM. 12

[34] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell. The bikenet mobile sensing system for cyclist experience mapping. In *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*, SenSys '07, pages 87–101, New York, NY, USA, 2007. ACM. 1, 12

[35] C. Ellul and M. Haklay. Creating community maps for the london thames gateway. IBG Annual International Conference. 1, 9

[36] E. Elnahrawy and B. Nath. Cleaning and querying noisy sensors. In *Proceedings of the 2Nd ACM International Conference on Wireless Sensor Networks and Applications*, WSNA '03, pages 78–87, New York, NY, USA, 2003. ACM. 32, 34, 100

[37] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan. The pothole patrol: Using a mobile sensor network for road surface monitoring. In *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*, MobiSys '08, pages 29–39, New York, NY, USA, 2008. ACM. 10

[38] B. Faltings, J. J. Li, and R. Jurca. Eliciting truthful measurements from a community of sensors. In *3rd IEEE International Conference on the Internet of Things*, IOT'12, pages 47–54. IEEE, 2012. 28, 64

[39] M. Faulkner, M. Olson, R. Chandy, J. Krause, K. M. Chandy, and A. Krause. The next big one: Detecting earthquakes and other rare events from community-based sensors. In *Proc. ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2011. 10

[40] U. Feige, V. S. Mirrokni, and J. Vondrak. Maximizing non-monotone submodular functions. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '07, pages 461–471, Washington, DC, USA, 2007. IEEE Computer Society. 44, 45

[41] S. Ganeriwal, L. K. Balzano, and M. B. Srivastava. Reputation-based framework for high integrity sensor networks. *ACM Trans. Sen. Netw.*, 4(3):15:1–15:37, June 2008. 96

[42] R. Ganti, F. Ye, and H. Lei. Mobile crowdsensing: current state and future challenges. *Communications Magazine, IEEE*, 49(11):32–39, November 2011. 2, 8

[43] R. K. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. F. Abdelzaher. Greengps: A participatory sensing fuel-efficient maps application. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pages 151–164, New York, NY, USA, 2010. ACM. 1, 10

[44] C. Gao, F. Kong, and J. Tan. Healthaware: Tackling obesity with health aware smart phone systems. In *Proceedings of the 2009 International Conference on Robotics and Biomimetics*, ROBIO'09, pages 1549–1554, Piscataway, NJ, USA, 2009. IEEE Press. 11

[45] S. Gaonkar, J. Li, R. R. Choudhury, L. Cox, and A. Schmidt. Micro-blog: Sharing and querying content through mobile phones and social participation. In *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*, MobiSys '08, pages 174–186, New York, NY, USA, 2008. ACM. 12

[46] D. Golovin, M. Faulkner, and A. Krause. Online distributed sensor selection. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, IPSN '10, pages 220–231, New York, NY, USA, 2010. ACM. 17, 61

[47] R. Gwadera. Multi-stream join answering for mining significant cross-stream correlations. *2013 IEEE 13th International Conference on Data Mining*, 0:851–856, 2010. 87

[48] R. Gwadera. Mdl-based segmentation of multi-attribute sequences. In *Spatial Data Mining and Geographical Knowledge Services (ICSDM), 2011 IEEE International Conference on*, pages 106–111, June 2011. 87

[49] D. Hasenfratz, O. Saukh, and L. Thiele. Model-driven accuracy bounds for noisy sensor readings. In *DCOSS*, pages 165–174, 2013. 33

[50] C.-F. Huang and Y.-C. Tseng. The coverage problem in a wireless sensor network. In *Proceedings of the 2Nd ACM International Conference on Wireless Sensor Networks and Applications*, WSNA '03, pages 115–121, New York, NY, USA, 2003. ACM. 17

[51] M. Huber, A. Kuwertz, F. Sawo, and U. Hanebeck. Distributed greedy sensor scheduling for model-based reconstruction of space-time continuous physical phenomena. In *Information Fusion, 2009. FUSION '09. 12th International Conference on*, pages 102–109, July 2009. 16, 61

[52] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden. Cartel: A distributed mobile sensor computing system. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, SenSys '06, pages 125–138, New York, NY, USA, 2006. ACM. 1, 10

[53] V. Isler and R. Bajcsy. The sensor selection problem for bounded uncertainty sensing models. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, IPSN '05, Piscataway, NJ, USA, 2005. IEEE Press. 14

[54] P. Jarvinen, T. Jarvinen, L. Lahteenmaki, and C. Sodergard. Hyperfit: Hybrid media in personal nutrition and exercise management. In *Pervasive Computing Technologies for Healthcare, 2008. PervasiveHealth 2008. Second International Conference on*, pages 222–226, Jan 2008. 11

[55] P. Jehiel and B. Moldovanu. Efficient design with interdependent valuations. *Econometrica*, 69(5):pp. 1237–1259, 2001. 25

[56] M. Jiang and W. McGill. Participatory risk management: Managing community risk through games. In *Social Computing (SocialCom), 2010 IEEE Second International Conference on*, pages 25–32, Aug 2010. 10

[57] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, pages 153–181. Kluwer Academic Publishers, 1996. 52, 76

[58] S. Joshi and S. Boyd. Sensor selection via convex optimization. *Trans. Sig. Proc.*, 57(2):451–462, Feb. 2009. 16, 61

[59] R. Jurca and B. Faltings. Minimum payments that reward honest reputation feedback. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, EC '06, pages 190–199, New York, NY, USA, 2006. ACM. 27

[60] E. Kalyvianaki, W. Wiesemann, Q. Vu, D. Kuhn, and P. Pietzuch. Sqpr: Stream query planning with reuse. In *Proceedings of the IEEE International Conference on Data Engineering*, 2011. 23, 62

[61] S. Kanhere. Participatory sensing: Crowdsourcing data from mobile smartphones in urban spaces. In C. Hota and P. Srimani, editors, *Distributed Computing and Internet Technology*, volume 7753 of *Lecture Notes in Computer Science*, pages 19–26. Springer Berlin Heidelberg, 2013. 3

[62] E. Kanjo, J. Bacon, D. Roberts, and P. Landshoff. Mobsens: Making smart phones smarter. *IEEE Pervasive Computing*, 8(4):50–57, Oct. 2009. 1, 9, 11

[63] A. Kapadia, D. Kotz, and N. Triandopoulos. Opportunistic sensing: Security challenges for the new paradigm. In *Communication Systems and Networks and Workshops, 2009. COMSNETS 2009. First International*, pages 1–10, Jan 2009. 3

[64] H. Ke. Two-phase query optimization in mobile ad hoc wireless networks. In *Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on*, volume 3, pages 515–520, Nov 2009. 23

[65] W. Khan, Y. Xiang, M. Aalsalem, and Q. Arshad. Mobile phone sensing systems: A survey. *Communications Surveys Tutorials, IEEE*, 15(1):402–427, First 2013. 8

[66] M. Klein, G. A. Moreno, D. C. Parkes, D. Plakosh, S. Seuken, and K. Wallnau. Handling interdependent values in an auction mechanism for bandwidth allocation in tactical data networks. In *NetEcon, 2008 Proceedings*, pages 73–78, New York, NY, USA, 2008. ACM. 25

[67] I. Koutsopoulos. Optimal incentive-driven design of participatory sensing systems. In *INFOCOM, 2013 Proceedings IEEE*, pages 1402–1410, April 2013. 64

[68] A. Krause, E. Horvitz, A. Kansal, and F. Zhao. Toward community sensing. In *Proceedings of the 7th International Conference on Information Processing in Sensor Networks*, IPSN '08, pages 481–492, Washington, DC, USA, 2008. IEEE Computer Society. 2

[69] A. Krause, E. Horvitz, A. Kansal, and F. Zhao. Toward community sensing. In *Proceedings of the 7th International Conference on Information Processing in Sensor Networks*, IPSN '08, pages 481–492, Washington, DC, USA, 2008. IEEE Computer Society. 18, 35, 41, 62

[70] A. Krause, R. Rajagopal, A. Gupta, and C. Guestrin. Simultaneous optimization of sensor placements and balanced schedules. *Automatic Control, IEEE Transactions on*, 56(10):2390–2405, Oct 2011. 17, 35, 61

[71] B. Krishnamachari and S. Iyengar. Distributed bayesian algorithms for fault-tolerant event region detection in wireless sensor networks. *IEEE Trans. Comput.*, 53(3):241–250, Mar. 2004. 31, 32, 34, 100

[72] D. Le-Phuoc, H. Q. Nguyen-Mau, J. X. Parreira, and M. Hauswirth. A middleware framework for scalable management of linked streams. *Web Semantics: Science, Services and Agents on the World Wide Web*, 16(0):42 – 51, 2012. 87

[73] Y. W. Lee, K. Y. Lee, and M. H. Kim. Energy-efficient multiple query optimization for wireless sensor networks. In *Proceedings of the 2009 Third International Conference on Sensor Technologies and Applications*, SENSORCOMM '09, pages 531–538, Washington, DC, USA, 2009. IEEE Computer Society. 21, 35, 62

[74] M. Li, T. Yan, D. Ganesan, E. Lyons, P. Shenoy, A. Venkataramani, and M. Zink. Multi-user data sharing in radar sensor networks. In *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*, SenSys '07, pages 247–260, New York, NY, USA, 2007. ACM. 22

[75] H.-S. Lim, G. Ghinita, E. Bertino, and M. Kantarcioglu. A game-theoretic approach for high-assurance of data trustworthiness in sensor networks. In *Proceedings of the 2012 IEEE 28th International Conference on Data Engineering*, ICDE '12, pages 1192–1203, Washington, DC, USA, 2012. IEEE Computer Society. 84, 92, 100

[76] H.-S. Lim, Y.-S. Moon, and E. Bertino. Provenance-based trustworthiness assessment in sensor networks. In *Proceedings of the Seventh International Workshop on Data Management for Sensor Networks*, DMSN '10, pages 2–7, New York, NY, USA, 2010. ACM. 29, 30, 84, 92, 100

[77] K. K. Loo, I. Tong, B. Kao, and D. Cheung. Online algorithms for mining inter-stream associations from large sensor networks. In *PAKDD '05: Proceedings of the 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, volume 3518 of *Lecture Notes in Computer Science*, pages 143–149, Hydarabad, India, June 2005. 101, 105

[78] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell. Soundsense: Scalable sound sensing for people-centric applications on mobile phones. In *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services*, MobiSys '09, pages 165–178, New York, NY, USA, 2009. ACM. 9

[79] J. Lu, L. Bao, and T. Suda. Coverage-aware sensor engagement in dense sensor networks. *J. Embedded Comput.*, 3(1):3–18, Jan. 2009. 17

[80] C. Lucchese, S. Orlando, and R. Perego. Dci-closed: A fast and memory efficient algorithm to mine frequent closed itemsets. In *In Proc. of the IEEE ICDM 2004 Workshop on Frequent Itemset Mining Implementations (FIMI04)*, 2004. 90

[81] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. The design of an acquisitional query processor for sensor networks. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, SIGMOD '03, pages 491–502, New York, NY, USA, 2003. ACM. 23, 62

[82] S. Madden, M. Shah, J. M. Hellerstein, and V. Raman. Continuously adaptive continuous queries over streams. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*, SIGMOD '02, pages 49–60, New York, NY, USA, 2002. ACM. 23

[83] N. Maisonneuve, M. Stevens, M. Niessen, and L. Steels. Noisetube: Measuring and mapping noise pollution with mobile phones. In *Information Technologies in Environmental Engineering*, Environmental Science and Engineering, pages 215–228. Springer Berlin Heidelberg, 2009. 1, 9

[84] A. Mas-Colell, M. Whinston, and J. Green. *Microeconomic Theory*. Oxford, 1995. 25

[85] E. Maskin and P. Dasgupta. Efficient auctions. *Quarterly Journal of Economics*, 115:341–388, 2000. 25

[86] S. Mathur, T. Jin, N. Kasturirangan, J. Chandrasekaran, W. Xue, M. Gruteser, and W. Trappe. Parknet: Drive-by sensing of road-side parking statistics. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pages 123–136, New York, NY, USA, 2010. ACM. 1, 10

[87] X. Meng, T. Nandagopal, L. Li, and S. Lu. Contour maps: Monitoring and diagnosis in sensor networks. *Comput. Netw.*, 50(15):2820–2838, Oct. 2006. 83

[88] C. Mezzetti. Mechanism design with interdependent valuations: Efficiency. *Econometrica*, 72(5):1617–1626, 09 2004. 25, 26

[89] N. Miller, P. Resnick, and R. Zeckhauser. Eliciting informative feedback: The peer-prediction method. *Manage. Sci.*, 51(9):1359–1373, Sept. 2005. 27, 28, 64

[90] E. Miluzzo, N. D. Lane, S. B. Eisenman, and A. T. Campbell. Cenceme: Injecting sensing presence into social networking applications. In *Proceedings of the 2Nd European Conference on Smart Sensing and Context*, EuroSSC'07, pages 1–28, Berlin, Heidelberg, 2007. Springer-Verlag. 12

[91] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell. Sensing meets mobile social networks: The design, implementation and evaluation of the cenceme application. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, SenSys '08, pages 337–350, New York, NY, USA, 2008. ACM. 12

[92] P. Mohan, V. N. Padmanabhan, and R. Ramjee. Nericell: Rich monitoring of road and traffic conditions using mobile smartphones. In *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, SenSys '08, pages 323–336, New York, NY, USA, 2008. ACM. 1, 10

[93] R. Muller and G. Alonso. Efficient sharing of sensor networks. In *Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on*, pages 109–118, Oct 2006. 20, 62

[94] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda. Peir, the personal environmental impact report, as a platform for participatory sensing systems research. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*, MobiSys '09, pages 55–68, New York, NY, USA, 2009. ACM. 11

[95] L. Nachman, A. Baxi, S. Bhattacharya, V. Darera, P. Deshpande, N. Kodalapura, V. Mageshkumar, S. Rath, J. Shahabdeen, and R. Acharya. Jog falls: A pervasive healthcare platform for diabetes management. In *Proceedings of the 8th International Conference on Pervasive Computing*, Pervasive'10, pages 94–111, Berlin, Heidelberg, 2010. Springer-Verlag. 11

[96] K. Ni and G. Pottie. Bayesian selection of non-faulty sensors. In *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*, pages 616–620, June 2007. 32, 34, 100

[97] D. Offenhuber and D. Lee. Putting the informal on the map: Tools for participatory waste management. In *Proceedings of the 12th Participatory Design Conference: Exploratory Papers, Workshop Descriptions, Industry Cases - Volume 2*, PDC '12, pages 13–16, New York, NY, USA, 2012. ACM. 10

[98] E. R. Omiecinski. Alternative interest measures for mining associations in databases. *IEEE Transactions on Knowledge and Data Engineering*, 15:57–69, January 2003. 91

[99] A. Papakonstantinou, A. Rogers, E. H. Gerding, and N. R. Jennings. Mechanism design for the truthful elicitation of costly probabilistic estimates in distributed information systems. *Artif. Intell.*, 175(2):648–672, Feb. 2011. 27, 64

[100] E. Paulos, R. J. Honicky, and B. Hooker. Citizen Science: Enabling Participatory Urbanism. *Handbook of Research on Urban Informatics: The Practice and Promise of the Real-Time City*, pages 414–436, 2009. 9

[101] J. Pei, J. Han, and R. Mao. Closet: An efficient algorithm for mining frequent closed itemsets. In *Proceedings of the 2000 ACM-SIGMOD international workshop of data mining and knowledge discovery (DMKD'00)*, pages 21–30, 2000. 90

[102] R. Porter, A. Ronen, Y. Shoham, and M. Tennenholtz. Fault tolerant mechanism design. *Artif. Intell.*, 172(15):1783–1799, Oct. 2008. 26, 64, 67, 68

[103] B. Predic, Z. Yan, J. Eberle, D. Stojanovic, and K. Aberer. Exposuresense: Integrating daily activities with air quality using mobile participatory sensing. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on*, pages 303–305, March 2013. 11

[104] S. Rajasegarar, C. Leckie, M. Palaniswami, and J. Bezdek. Distributed anomaly detection in wireless sensor networks. In *Communication systems, 2006. ICCS 2006. 10th IEEE Singapore International Conference on*, pages 1–5, Oct 2006. 34

[105] S. D. Ramchurn, C. Mezzetti, A. Giovannucci, J. A. Rodriguez-Aguilar, R. K. Dash, and N. R. Jennings. Trust-based mechanisms for robust and efficient task allocation in the presence of execution uncertainty. *J. Artif. Int. Res.*, 35(1):119–159, June 2009. 26, 64

[106] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu. Ear-phone: An end-to-end participatory urban noise mapping system. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, IPSN '10, pages 105–116, New York, NY, USA, 2010. ACM. 9

[107] S. Reddy, A. Parker, J. Hyman, J. Burke, D. Estrin, and M. Hansen. Image browsing, processing, and clustering for participatory sensing: Lessons from a dietsense prototype. In *Proceedings of the 4th Workshop on Embedded Networked Sensors*, EmNets '07, pages 13–17, New York, NY, USA, 2007. ACM. 1, 11

[108] K. Römer. Discovery of frequent distributed event patterns in sensor networks. In *Proceedings of the 5th European Conference on Wireless Sensor Networks*, EWSN'08, pages 106–124. Springer-Verlag, Berlin, Heidelberg, 2008. 101

[109] H. Rowaihy, S. Eswaran, M. Johnson, D. Verma, A. Bar-noy, and T. Brown. A survey of sensor selection schemes in wireless sensor networks. In *In SPIE Defense and Security Symposium Conference on Unattended Ground, Sea, and Air Sensor Technologies and Applications IX*, 2007. 16

[110] L. J. Savage. Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association*, 66(336):pp. 783–801, 1971. 26

[111] S. Sehgal, S. S. Kanhere, and C. T. Chou. Mobishop: Using mobile phones for sharing consumer pricing information. In *Demo Session of the Intl. Conference on Distributed Computing in Sensor Systems*, 2008. 13

[112] T. K. Sellis. Multiple-query optimization. *ACM Trans. Database Syst.*, 13(1):23–52, Mar. 1988. 20, 62

[113] S. Seshadri, V. Kumar, B. Cooper, and L. Liu. Optimizing multiple distributed stream queries using hierarchical network partitions. In *Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International*, pages 1–10, March 2007. 23

[114] K. Sha, G. Zhan, W. Shi, M. Lumley, C. Wiholm, and B. Arnetz. Spa: A smart phone assisted chronic illness self-management system with participatory sensing. In *Proceedings of the 2Nd International Workshop on Systems and Networking Support for Health Care and Assisted Living Environments*, HealthNet '08, pages 5:1–5:3, New York, NY, USA, 2008. ACM. 11

[115] M. Shamaiah, S. Banerjee, and H. Vikalo. Greedy sensor selection: Leveraging submodularity. In *CDC*, pages 2572–2577. IEEE, 2010. 16, 35, 61

[116] K.-P. Shih, Y.-D. Chen, C.-W. Chiang, and B.-J. Liu. A distributed active sensor selection scheme for wireless sensor networks. In *Computers and Communications, 2006. ISCC '06. Proceedings. 11th IEEE Symposium on*, pages 923–928, June 2006. 17, 35

[117] K. Shilton. Four billion little brothers?: Privacy, mobile phones, and ubiquitous data collection. *Commun. ACM*, 52(11):48–53, Nov. 2009. 11

[118] R. Shokri, J. Freudiger, and J. pierre Hubaux. A unified framework for location privacy. In *3rd Hot Topics in Privacy Enhancing Technologies (HotPETs)*, 2010. 71

[119] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux. Quantifying location privacy. In *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, SP '11, pages 247–262, Washington, DC, USA, 2011. IEEE Computer Society. 72

[120] R. Shokri, G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux, and J.-Y. Le Boudec. Protecting location privacy: Optimal strategy against localization attacks. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, pages 617–627, New York, NY, USA, 2012. ACM. 72

[121] A. Silberstein and J. Yang. Many-to-many aggregation for sensor networks. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 986–995, April 2007. 23

[122] A. Singla and A. Krause. Incentives for privacy tradeoff in community sensing. In *HCOMP*. AAAI, 2013. 28, 72

[123] S. Stein, E. Gerding, A. Rogers, K. Larson, and N. Jennings. Algorithms and mechanisms for procuring services with uncertain durations using redundancy. *Artificial Intelligence*, 175(14-15):2021–2060, September 2011. 26

[124] A. Strehl, E. Strehl, J. Ghosh, and R. Mooney. Impact of similarity measures on web-page clustering. In *In Workshop on Artificial Intelligence for Web Search (AAAI 2000*, pages 58–64. AAAI, 2000. 31

[125] E. P. Stuntebeck, J. S. Davis, II, G. D. Abowd, and M. Blount. Healthsense: Classification of health-related sensor data through user-assisted machine learning. In *Proceedings of the 9th Workshop on Mobile Computing Systems and Applications*, HotMobile '08, pages 1–5, New York, NY, USA, 2008. ACM. 11

[126] L.-A. Tang, X. Yu, S. Kim, J. Han, C.-C. Hung, and W.-C. Peng. Tru-alarm: Trustworthiness analysis of sensor networks in cyber-physical systems. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, pages 1079–1084, Washington, DC, USA, 2010. IEEE Computer Society. 32

[127] A. Thiagarajan, J. Biagioni, T. Gerlich, and J. Eriksson. Cooperative transit tracking using smart-phones. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 85–98, New York, NY, USA, 2010. ACM. 10

[128] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson. Vtrack: Accurate, energy-aware road traffic delay estimation using mobile phones. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys '09, pages 85–98, New York, NY, USA, 2009. ACM. 1, 10

[129] W. H. Tok and S. Bressan. Efficient and adaptive processing of multiple continuous queries. In *Advances in Database Technology EDBT 2002*, pages 215–232, 2002. 22, 35, 62

[130] N. Trigoni, Y. Yao, A. Demers, J. Gehrke, and R. Rajaraman. Multi-query optimization for sensor networks. In *Proceedings of the First IEEE International Conference on Distributed Computing in Sensor Systems*, DCOSS'05, pages 307–321, Berlin, Heidelberg, 2005. Springer-Verlag. 21, 35, 62

[131] N. Trigoni, Y. Yao, A. Demers, J. Gehrke, and R. Rajaraman. Multi-query optimization for sensor networks. TR2005-1989, Cornell Univ, 2005. 21

[132] M. C. Vuran, O. B. Akan, and I. F. Akyildiz. Spatio-temporal correlation: Theory and applications for wireless sensor networks. *Comput. Netw.*, 45(3):245–259, June 2004. 84

[133] X. R. Wang, J. T. Lizier, O. Obst, M. Prokopenko, and P. Wang. Spatiotemporal anomaly detection in gas monitoring sensor networks. In *Proceedings of the 5th European Conference on Wireless Sensor Networks*, EWSN'08, pages 90–105, Berlin, Heidelberg, 2008. Springer-Verlag. 32, 34, 100

[134] J. Witkowski. Eliciting honest reputation feedback in a markov setting. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, IJCAI'09, pages 330–335, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc. 27, 64

[135] J. Wolf, N. Bansal, K. Hildrum, S. Parekh, D. Rajan, R. Wagle, K.-L. Wu, and L. Fleischer. Soda: An optimizing scheduler for large-scale stream-based distributed computer systems. In *Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware*, Middleware '08, pages 306–325, New York, NY, USA, 2008. Springer-Verlag New York, Inc. 23

[136] S. Xiang, H.-B. Lim, K.-L. Tan, and Y. Zhou. Two-tier multiple query optimization for sensor networks. In *Distributed Computing Systems, 2007. ICDCS '07. 27th International Conference on*, pages 39–39, June 2007. 20

[137] X.-Y. Xiao, W.-C. Peng, C.-C. Hung, and W.-C. Lee. Using sensorranks for in-network detection of faulty readings in wireless sensor networks. In *Proceedings of the 6th ACM International Workshop on Data Engineering for Wireless and Mobile Access*, MobiDE '07, pages 1–8, New York, NY, USA, 2007. ACM. 30, 32, 84

[138] W. Xue, Q. Luo, L. Chen, and Y. Liu. Contour map matching for event detection in sensor networks. In *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*, SIGMOD '06, pages 145–156, New York, NY, USA, 2006. ACM. 100

[139] Z. Yan, J. Eberle, and K. Aberer. Optimos: Optimal sensing for mobile sensors. In *Proceedings of the 2012 IEEE 13th International Conference on Mobile Data Management (Mdm 2012)*, MDM '12, pages 105–114, Washington, DC, USA, 2012. IEEE Computer Society. 56

[140] Y. Yang, A. Ambrose, and M. Cardei. Coverage for composite event detection in wireless sensor networks. *Wirel. Commun. Mob. Comput.*, 11(8):1168–1181, Aug. 2011. 90

[141] Y. Yao and J. Gehrke. The cougar approach to in-network query processing in sensor networks. *SIGMOD Rec.*, 31(3):9–18, Sept. 2002. 23

[142] A. Yu and A. Nahapetian. Participatory sensing for fighting food deserts. In *Proceedings of the 8th International Conference on Body Area Networks*, BodyNets '13, pages 221–224, ICST, Brussels, Belgium, Belgium, 2013. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). 12

[143] Y. Zhang, N. Meratnia, and P. Havinga. Outlier detection techniques for wireless sensor networks: A survey. *Commun. Surveys Tuts.*, 12(2):159–170, Apr. 2010. 33, 83, 84

[144] Z. Zhang, A. D. Kshemkalyani, and S. M. Shatz. Dynamic multiroot, multiquery processing based on data sharing in sensor networks. *ACM Trans. Sen. Netw.*, 6(3):25:1–25:38, June 2010. 23

[145] A. Zohar and J. S. Rosenschein. Mechanisms for information elicitation. *Artificial Intelligence*, 172(1617):1917 – 1939, 2008. 26, 64

Avenue de Cour 15
1007 Lausanne
Switzerland
✆ +41 78 865 0271
☎ +41 21 693 1314
✉ meriahi@gmail.com
in mehdiriahi
mdii

Mehdi Riahi

PhD, EPFL, Switzerland

Strengths

- Skilled in design and implementation of distributed and scalable systems
- Expert in design and implementation of scalable data stream processing systems
- Experienced in teamworking in successful industrial and semi-industrial projects

Education

2010–2015 **PhD**, *School of Computer and Communication Sciences*, EPFL, Switzerland.
- Research topics: Stream data processing, quality assessment of sensor data, data management in participatory sensing
- Advisor: Prof. Karl Aberer

2005–2008 **MSc, Computer Architecture**, *Isfahan University of Technology*, Isfahan, Iran.

2000–2005 **BSc, Software Engineering**, *Isfahan University of Technology*, Isfahan, Iran.

Publications

- R. Gwadera, M. Riahi, and K. Aberer. Pattern-wise trust assessment of sensor data. *MDM 2014 - 15th IEEE International Conference on Mobile Data Management*, Brisbane, Australia, 2014.

- M. Riahi, T. G. Papaioannou, I. Trummer and K. Aberer. Utility-driven Data Acquisition in Participatory Sensing. *16th International Conference on Extending Database Technology (EDBT)*, Genoa, Italy, 2013.

- T. G. Papaioannou, M. Riahi and K. Aberer. Towards Online Multi-Model Approximation of Time Series. *MDM 2011 - 12th International Conference on Mobile Data Management*, Lulea, Sweden, 2011.

- A. Salehi, M. Riahi, S. Michel and K. Aberer. GSN, Middleware for Streaming World (Best Demo Award). *MDM 2009 - 10th International Conference on Mobile Data Management*, Taipei, Taiwan, 2009.

- A. Salehi, M. Riahi, S. Michel and K. Aberer. Knowing When to Slide: Efficient Scheduling for Sliding Window Processing. *MDM 2009 - 10th International Conference on Mobile Data Management*, Taipei, 2009.

Projects

2013–present **OpenIoT**, `https://github.com/OpenIotOrg/openiot`.
Open source cloud-based solution for Internet of Things (IoT). Identified as one of the top ten new open source projects of 2013 ◪.
- Designed and implemented the security and privacy module based on CAS and OAuth2.0

2010–2013 **OpenSense**, `http://opensense.epfl.ch`.
Community-based wireless sensor network for monitoring air pollution.
- Proposed a multi-model based technique for efficient storage and querying of large amount of sensor data achieving up to 80% storage gain
- Proposed a utility-driven data collection framework for community sensing systems

2008–2010 **GSN**, `https://github.com/LSIR/gsn`.
Middleware for collecting and processing sensor data in the Internet, used by several academic and industrial institutes.
- Designed and implemented the sliding window component for efficiently answering continuous queries
- Co-designed and implemented push and pull based stream data publishing

2007 **Master project**, *Isfahan University of Technology*.
Off-line Persian handwriting recognition system.
- Proposed and prototyped a novel system for recognizing handwritten text in Persian language with acceptable accuracy given the complexity of the Persian script

2005 **Bachelor project**, *Isfahan University of Technology*.
General-purpose network service configuration utility.
- Developed an integrated and extendable solution for facilitating configuration of network services, such as web servers, routers, etc., through a graphical interface

Experience

2012–2013 **Lead developer**, *Scalendo (startup)*, Lausanne, Switzerland.
Scalendo is a novel and cost-effective cloud brokerage solution for satisfying customer requirements on data durability, availability and vendor lock-in avoidance.
- Co-designed a scalable architecture capable of handling billions of objects
- Developed scalable and real-time analytics, object placement and replacement engine

2009–2010 **Internship**, *Distributed Information Systems Laboratory*, EPFL, Switzerland.
- Improved features and performance of the stream processing middleware GSN
- Co-designed and prototyped back-end and front-end of an innovative system for facilitating strategic trading for professional and non-professional stock traders

2008–2009 **Architect/Developer**, *Infoprosys (startup)*, Isfahan, Iran.
- Co-designed and developed a, first-in-kind in the country, online school application for facilitating school-parents communication and for educational purposes (courses, homework, grades, etc.)

2007–2008 **Project manager/Lead developer**, *Integrated Computing Systems Co.*, Isfahan, Iran.
- Led design and development of an office automation web portal timely delivered to Isfahan science and technology town
- Designed and implemented a, first-in-kind in country, pipeline inspection and management application for oil companies based on uDig GIS framework