Learning and Incentives in Crowd-Powered Systems

A thesis submitted to attain the degree of
DOCTOR OF SCIENCES of ETH ZURICH
(Dr. sc. ETH Zurich)

presented by

ADISH KUMAR SINGLA
M. Sc. in Computer Science, EPFL, Switzerland
born on 12.03.1984
citizen of India

accepted on the recommendation of
Prof. Dr. Andreas Krause (ETH Zurich), examiner
Prof. Dr. Donald Kossmann (ETH Zurich), co-examiner
Prof. Dr. Carla Gomes (Cornell University), co-examiner
Dr. Eric Horvitz (Microsoft Research), co-examiner
Prof. Dr. David Parkes (Harvard University), co-examiner

2016
The ongoing technological revolution—fueled by the Internet, mobile computing, and advances in AI—is leading to deep-seated economic and societal changes, and fundamentally altering our lives in unprecedented ways. An underlying theme of this revolution is that the users are increasingly becoming an integral part of computational systems; prominent examples include community-driven services like Airbnb, shared mobility based bike sharing systems, citizen science projects like eBird, communitiesensing applications like Waze, online tutoring systems like Coursera, and many more. These emerging crowd-powered systems are opening up exciting new opportunities for industry and science by providing the ability to harness the knowledge/expertise of people at scale and by collecting/analyzing the unprecedented volume of users’ data. However, the machine learning algorithms and AI techniques behind these crowd-powered systems are increasingly interacting with and seeking information from people. This raises fundamentally new issues and technical challenges including concerns about users’ privacy, incentivizing users, and the robustness of the deployed algorithms against the intricacies of human behavior (e.g., users acting strategically). Furthermore, these systems typically face various operational challenges on a daily basis—for instance, imbalance of the bikes across stations in bike sharing systems and the huge disproportion of the revenue across hosts in services like Airbnb—as their users act primarily in their own self-interest.

In this dissertation, we develop novel techniques at the interplay of learning and incentives to tackle some of the above-mentioned challenges with the goal of improving the overall effectiveness of crowd-powered systems. Part II of this dissertation explores the first facet of this interplay—Learning about Incentives—where we develop new online learning mechanisms using the approach of regret minimization for learning about users’ preferences and offering them optimized incentives. In Part III, we study the
aspect of Incentives for Learning, in particular for data collection, exploring the role of designing/engineering the incentives in crowd-powered systems for incentivizing users to solving complex sensing and computation tasks. We design a new class of privacy-aware truthful mechanisms suitable for a wide range of sensing applications dealing with information acquisition from strategic agents under uncertainty and enjoying provable guarantees on the acquired utility. Finally, Part IV of this dissertation tightly couples this interplay and explores Learning as an Incentive, motivated by applications to educating volunteers in citizen science. We introduce a new stochastic model of human learners and then develop a novel teaching algorithm that selects a sequence of training examples to the human learners in order to steer them towards the true hypothesis. We theoretically analyze our approach, proving strong approximation guarantees, and thereby providing—one of the first—teaching complexity results for a practically applicable approach towards teaching the crowd.

As part of the research carried out in this dissertation, we collaborate with various industrial/academic partners and interact directly with their end users (via conducting surveys, running online experiments, or deploying smartphone apps). Through this process, we gain deep insights into real-world problems, a key towards developing practically applicable techniques. In this dissertation, we apply our techniques across three important application domains presented in three different parts of this dissertation. Part II tackles the imbalance problem in bike sharing systems by developing a new crowdsourcing scheme to incentivize users in the bike repositioning process (in collaboration with a bike sharing company in Zurich, Switzerland and a public transport company in Mainz, Germany). In Part III, we develop practical solutions for community-based air-quality sensing (in the context of OpenSense, a Swiss nationwide project). In Part IV, we present our results on teaching participants on a crowdsourcing platform for different tasks of classifying animal species with applications to biodiversity monitoring via citizen science.

By grounding our work in three prototypical real-world applications, this dissertation explores several fundamental challenges that are prevalent in the ongoing technological revolution: incentivizing users in the era of the sharing economy, ensuring the privacy of users in the digital world, and designing personalized teaching policies for e-education. A central theme of our research is empowering users and actively engaging them to contribute to the system, with a goal towards building self-sustainable and intelligent crowd-powered systems.
Zusammenfassung


In dieser Dissertation entwickeln wir neue Methoden am Schnittpunkt zwischen dem Lernen und Anreizen, um die zuvor beschriebenen Herausforderungen zu lösen und die Effektivität von crowd-basierten Systemen zu verbessern. Teil II dieser Dissertation beschäftigt sich mit dem ersten Aspekt dieser Interaktion, dem Lernen von

First and foremost, I want to express my deepest gratitude to my advisor Andreas Krause. Andreas has been the best advisor I could hope for. He has always inspired me through his immense enthusiasm for research and has provided relentless support in the past five years. I want to thank Andreas for giving me freedom to pursue various research ideas, providing me detailed guidance along the way, while at the same time ensuring that I stay focused.

I am indebted to the members of my dissertation committee—Andreas Krause, Donald Kossmann, Carla Gomes, Eric Horvitz, and David Parkes—for providing me valuable feedback and a lot of encouragement.

I am thankful to my collaborators and co-authors at ETH Zurich, and beyond—each one of you contributed uniquely towards this dissertation. I am also thankful for the opportunity to work with amazing Master students at ETH Zurich. Thanks to my mentors and colleagues at Microsoft Research—Eric Horvitz, Ece Kamar, Pushmeet Kohli, Ryen White, and Dengyong Zhou—for long-lasting collaborations, research guidance, and career advice.

A special thanks to our group admin, Rita Klute, for immense support and smilingly taking on every administrative task. Her warm personality made life so much easier at ETH Zurich. I also want to thank all the current and former members of the Learning and Adaptive Systems Group, especially to my office mates Sebastian, Gábor, Yuxin, and Hasta.

Finally, I am indebted to my parents, my family, my little nieces/nephews, and my dear friends for their endless support and inspiration. I know you all have been waiting very eagerly for me to finish this dissertation, you made me push forward.
# Contents

I Background and Contributions

1 Introduction

1.1 Crowd-Powered Systems

1.1.1 Technical and Operational Challenges

1.1.2 Interplay of Learning and Incentives

1.2 Key Ingredients of Our Approach

1.2.1 Identifying Technical and Operational Challenges

1.2.2 Engaging and Empowering the Crowd

1.2.3 Techniques at the Interplay of Learning and Incentives

1.3 Overview of Main Contributions

1.3.1 Learning about Incentives with Applications to Balancing Bike Sharing Systems

1.3.2 Incentives for Learning with Applications to Community-Based Air Quality Sensing

1.3.3 Learning as an Incentive with Applications to Biodiversity Monitoring via Citizen Science

1.3.4 Summary of Theoretical Results

1.4 Organization of this Dissertation
2 Research Publications and Collaborations 17
  2.1 Publications Covered in this Dissertation ......................... 17
  2.2 Publications Not Covered in this Dissertation .................... 19

II Learning about Incentives 21

3 Overview of Part II 23
  3.1 Relevant Research Problems and Our Contributions ............... 24
      3.1.1 Learning Hemimetrics Encoding Switching Preferences ........ 24
      3.1.2 Learning Skills and Expertise ................................ 25

4 Truthful Incentives via Online Learning Mechanisms 27
  4.1 Overview of Our Approach ........................................... 29
      4.1.1 Pricing Models and Interaction with Users ..................... 29
      4.1.2 Our Results .................................................. 29
  4.2 The Model ............................................................ 30
      4.2.1 Task Requester and Users ..................................... 30
      4.2.2 Online Arrival of Users ....................................... 31
      4.2.3 Optimal Benchmarks ........................................... 32
      4.2.4 Utility and Regret ............................................ 32
  4.3 Our Mechanisms ....................................................... 33
      4.3.1 Background and High-Level Ideas ............................. 33
      4.3.2 Mechanism BP-DGREEDY for Bidding Model ................... 35
      4.3.3 Mechanism BP-UCB for Posted-Price Model ................... 37
  4.4 Performance Analysis ................................................ 37
      4.4.1 Components Contributing to the Regret ....................... 37
      4.4.2 Regret Bounds for BP-DGREEDY ................................ 38
      4.4.3 Regret Bounds for BP-UCB .................................... 39
## Contents

4.5 Related Work ................................................. 40  
4.5.1 Multi-Armed Bandits and Regret Minimization ............ 40  
4.5.2 Learning in Online Auctions ............................. 40  
4.5.3 Online Procurement Auctions ............................ 41  
4.6 Summary .................................................. 42  

5 Applications to Procurement in Crowdsourcing Marketplaces 43  
5.1 Experimental Setup ........................................ 44  
5.1.1 Benchmarks and Baselines ............................... 44  
5.1.2 Metrics and Types of Experiments ...................... 46  
5.1.3 Parameter Choices ..................................... 46  
5.1.4 Cost Distributions ...................................... 46  
5.2 Experimental Results on Simulated Distributions .......... 46  
5.2.1 Acquired Utility ....................................... 47  
5.2.2 Getting Deeper Insights ................................. 48  
5.3 Survey Study on the Mechanical Turk Platform ............. 50  
5.3.1 Setup of the Survey Study .............................. 50  
5.3.2 Statistics of the Data ................................... 51  
5.4 Experimental Results on Real Workers’ Distribution ..... 52  
5.4.1 Acquired Utility ....................................... 53  
5.4.2 Robustness in Different Online Settings ................. 54  
5.5 Related Work ............................................. 54  
5.5.1 Incentives in Crowdsourcing Platforms .................. 54  
5.6 Summary .................................................. 55  

6 Applications to Balancing Bike Sharing Systems ............. 57  
6.1 Overview of Our Approach ................................ 59  
6.1.1 Architecture of Our Incentives System ................. 60
6.2 The Model ................................................................. 61
  6.2.1 Bike Sharing Systems ............................................ 61
  6.2.2 Quality of Service ................................................. 61
  6.2.3 Truck-Based Repositioning Policies ................................. 62
  6.2.4 User Model ......................................................... 62
  6.2.5 Incentivizing Users for Repositioning ............................. 63
6.3 The Incentives System .................................................. 63
  6.3.1 Incentives Deployment Schema .................................... 63
  6.3.2 Pricing Mechanism: Dynamic BP-UCB ............................... 65
  6.3.3 Truthfulness of the Incentives System ............................. 66
6.4 Experimental Setup ...................................................... 67
  6.4.1 Historical BSS Dataset from Boston’s Hubway .................... 68
  6.4.2 Survey Study Among BSS Customers in Mainz, Germany ........ 68
  6.4.3 BSS Simulator ...................................................... 69
6.5 Experimental Results on BSS Simulator ................................. 69
  6.5.1 Varying Budget ...................................................... 69
  6.5.2 Budget Tradeoff Between Incentives and TRUCKS ................. 71
  6.5.3 Varying User Participation ....................................... 71
6.6 Deployment in Mainz, Germany ........................................... 71
  6.6.1 Participation and Reaction to Incentives ............................ 72
  6.6.2 Temporal and Spatial Distribution of Accepted Offers ............ 73
6.7 Related Work ............................................................ 73
  6.7.1 Truck-Based Repositioning in BSS ................................ 73
  6.7.2 Crowd-Based Repositioning in BSS ................................ 73
6.8 Summary ................................................................. 74
8.5.3 Budget Feasibility ................................................. 95
8.5.4 Guarantees on the Utility ................................. 95
8.6 Related Work .................................................. 96
  8.6.1 Privacy, Incentives, and Sensing ................. 96
8.7 Summary ..................................................... 97

9 Applications to Community-Based Air Quality Sensing 99
  9.1 Survey Study ................................................. 100
    9.1.1 Setup of the Survey Study ............................. 100
    9.1.2 Statistics of the Data .................................. 101
  9.2 The OpenSense Project: Data Collection and Spatial Modeling .... 101
    9.2.1 Calibrating NODE+ CO2 Gas Sensors ................. 102
    9.2.2 Data Collection from NODE+ Sensors Mounted on a Bike ... 102
  9.3 Experimental Setup ........................................ 104
    9.3.1 Sensing Locations using the OpenStreetMap Data .... 104
    9.3.2 Population of Users using the Strava Data ........... 105
    9.3.3 Utility Function Based on the Spatial Model of CO2 .... 105
    9.3.4 Simulating User Profiles ............................... 106
    9.3.5 Benchmarks and Baselines ............................. 107
    9.3.6 Metrics and Types of Experiments .................... 107
  9.4 Experimental Results ....................................... 108
    9.4.1 Computing Tighter Bounds on Payment ............... 108
    9.4.2 Varying the Given Budget and Specified Utility .... 109
    9.4.3 Varying Obfuscation Levels ........................... 109
  9.5 Related Work .............................................. 111
    9.5.1 Applications of Community Sensing ................. 111
  9.6 Summary ................................................... 111
IV Learning as an Incentive

10 Overview of Part IV

10.1 Relevant Research Problems and Our Contributions

10.1.1 Personalized Teaching Policies

11 Near-Optimally Teaching the Crowd to Classify

11.1 Overview of Our Approach

11.2 Teaching Process

11.2.1 The Learning Domain

11.2.2 The Teaching Protocol

11.3 Model of the Learner

11.4 Teaching Algorithm: STRICT

11.5 Performance Analysis

11.5.1 Approximation Guarantees

11.5.2 Teaching Complexity for Linear Separators

11.6 Related Work

11.6.1 Non-Interactive vs. Interactive Teaching

11.6.2 Noise-Free vs. Noise-Tolerant Teaching

11.7 Summary

12 Applications to Biodiversity Monitoring via Citizen Science

12.1 Teaching Task: Identifying Vespula vs. Weevil

12.2 Teaching Task: Identifying Butterflies vs. Moths

12.3 Teaching Task: Identifying Endangered Woodpecker Bird Species

12.4 Experimental Results on Simulated Learners

12.4.1 Test Error

12.4.2 Robustness against Learner’s Mismatched Parameters

12.4.3 On the Difficulty Level of Teaching
A.3 Regret Bounds for BP-UCB
   Proof of Theorem 4.2 .................................................. 163

B Proofs of Part III ......................................................... 167
   B.1 Truthfulness of the Mechanism
       Proof of Theorem 8.1 ................................................. 167
   B.2 Individually Rationality
       Proof of Theorem 8.2 ................................................. 169
   B.3 Budget Feasibility
       Proof of Theorem 8.3 ................................................. 170
   B.4 Guarantees on the Utility
       Proof of Theorem 8.4 ................................................. 172

C Proofs of Part IV ......................................................... 177
   C.1 Hardness Results
       Proof of Proposition 11.1 ............................................ 177
   C.2 Useful Lemmas ....................................................... 178
   C.3 Guarantees on the Convergence of STRICT
       Proof of Theorem 11.1 ............................................. 178
   C.4 Teaching Complexity for Linear Separators
       Proof of Theorem 11.2 ............................................. 180
Part I

Background and Contributions
The landscape of interaction between computational systems and their users is evolving fast, primarily fueled by recent technological advancements as well as deep-seated economic and societal changes. Users are increasingly becoming an integral part of computational systems, thereby playing a pivotal role in their functionality, driving their growth, and deciding on the eventual success (or failure) of these systems. Influential examples range from Internet-based community-edited encyclopedia Wikipedia [Wik] to cyber-physical systems such as the community traffic-sensing application Waze [Waz]. In this dissertation, we study these emerging crowd-powered systems and develop novel techniques at the interplay of learning and incentives with the goal of improving their overall effectiveness.

1.1 Crowd-Powered Systems

Let us begin by looking at the spectrum of crowd-powered systems. Technological advancements, primarily the growth of the Internet and mobile computing in the last decade, have created numerous opportunities to fuse computing with the capabilities of people at scale. This has led to the development of several crowd-powered online services including the community-edited encyclopedia Wikipedia [Wik], the collaboratively edited question-and-answer site StackOverflow [Sta],
and many more. The proliferation of smartphones and sensor-equipped devices such as GPS has created unique opportunities to sense the environment around us. This has led to the emergence of cyber-physical systems—a popular example is the community traffic-sensing application Waze [Waz] used by over 50 million users worldwide.

In science, citizen science projects such as eBird [Ebi; Sul+09] or GalaxyZoo [Gal; Lin+08] empower ordinary citizens to contribute to scientific endeavors such as biodiversity monitoring and classifying astronomical observations. The scale of annotations made available by these projects is unprecedented: within half a year of the release of the first GalaxyZoo project, over 40 million annotations were provided by the volunteers.

A sub-class of crowd-powered systems, particularly popular in the industry, includes the microwork based crowdsourcing platforms like Amazon’s Mechanical Turk [Mtu] and CrowdFlower [Cro]. These platforms have triggered a rise of crowdsourcing approaches enabling us to systematically coordinate the crowd to perform micro-tasks in exchange for small amounts of monetary reward. This, in turn, has fueled the recent successes in large-scale machine learning systems by providing access to training data at scale for learning tasks such as computer vision and natural language processing. Going beyond microwork, online outsourcing and freelancing platforms are erasing international borders for accessing skilled workers. For instance, Upwork [Upw]—the world’s largest freelancer marketplace—has a talent pool of over ten million registered freelancers with diverse skills such as web development, journalism, and marketing.

The emerging paradigm of the sharing economy is fueling collaborative consumption that is often facilitated by community-based online services. Influential examples include online vacation rental marketplace AirBnb [Air] and online transportation network company Uber [Ube]. The size and scale of these crowd-powered marketplaces now rival or even surpass most of the traditional businesses in hospitality and transportation. This model of collaborative consumption is emerging in many other sectors including social lending, crowdfunding campaigns like Kickstarter [Kic], shared-mobility systems [MD14], and peer-to-peer task assignments in online MOOCs like Coursera [Cou].

1.1.1 Technical and Operational Challenges

As people are becoming an integral part of computational systems, this raises fundamentally new questions and technical challenges for the AI and machine learning
1.1. Crowd-Powered Systems

Techniques behind these systems. Consider, for example, a typical active learning algorithm designed for querying labels from domain experts or an optimization technique for sensor selection. In these emerging systems, these domain experts and programmable sensors are increasingly being replaced by people (i.e., the users of these systems). This presents new technical challenges regarding the robustness of deployed algorithms that typically do not account for the intricacies of human behavior (e.g., users acting strategically). This also raises new issues including concerns about users’ privacy and questions about incentivizing the users.

Furthermore, these systems usually face various operational challenges on a daily basis. Consider, for example, bike sharing systems based on shared mobility. One of the primary reasons for the popularity of these systems is the flexibility they provide: users can pick up or drop off the bikes at a station of their choice without prior notice or time planning. This increased flexibility comes with the challenge of unpredictable and fluctuating demand as well as irregular flow patterns of the bikes. As a result, these systems often become imbalanced throughout the day, such as the unavailability of bikes or parking docks at stations. In this light, operators deploy fleets of vehicles which redistribute the bikes among stations in order to guarantee a desirable service level. This requires extensive logistic work and poses major operational costs. Some of the operational challenges in similar systems based on the model of collaborative consumption can be attributed to the fact that their users act primarily in their own self-interest.

1.1.2 Interplay of Learning and Incentives

In this dissertation, we work towards the goal of building self-sustainable and intelligent crowd-powered systems by incentivizing and empowering their users to actively engage them in contributing to the system processes. Some of the fundamental challenges arise here because of the human participation: It is notoriously difficult to design and learn tractable models of human preferences, and thus difficult to predict the behavior of users in real-world systems. This leads us to explore several questions at the intersection of learning and incentives. In particular, we develop novel techniques encompassing the following three facets of this interplay:

- **Learning about incentives.** How can we simultaneously learn about users’ preferences (i.e., desired incentives) as well as their abilities and skills, while using
the obtained information in order to maximize effectiveness?

- **Incentives for learning.** How can we design privacy-aware incentive-compatible mechanisms for learning and optimal information gathering? *i.e.*, how should we recruit users to solve complex sensing and computation tasks?

- **Learning as an incentive.** How can we use learning itself as an incentive, by teaching participants in citizen science projects and crowdsourcing platforms to be more effective?

We address these fundamental questions by building on state of the art results in machine learning, probabilistic modeling, and game theory. In particular, we build on results from multi-armed bandit methods in structured spaces, submodular function optimization, active learning, and budget feasible mechanism design.

### 1.2 Key Ingredients of Our Approach

As part of the research carried out in this dissertation, we collaborate with various industrial and academic partners, as well as interact with end users via survey studies and online experiments. These collaborations and interactions allow us to gain deep insights of the real-world problems in important application domains. Before presenting our main algorithmic techniques and contributions in the next section, we now discuss key ingredients of our approach, as illustrated in the Figure 1.1. In this dissertation, we demonstrate our approach across three real-world applications including balancing bike sharing systems, a community-based system for air quality sensing, and educating the volunteers/participants in citizen science projects for biodiversity monitoring. These three applications serve as prototypical examples covering the above-mentioned spectrum of crowd-powered systems. By grounding our work in these three applications, we explore several fundamental challenges that are prevalent in the technological revolution around us: (*i*) incentivizing users in the era of the sharing economy, (*ii*) ensuring the privacy of users in this digital world, and (*iii*) designing personalized teaching policies for e-education.
1.2. Key Ingredients of Our Approach

Crowd-powered system and its users
Identifying technical and operational challenges
Engaging and empowering the crowd
Techniques at the interplay of learning and incentives
Self-sustainable and intelligent crowd-powered system
Key ingredients of our approach

Figure 1.1: Key ingredients of our approach with a goal towards building self-sustainable and intelligent crowd-powered systems.

1.2.1 Identifying Technical and Operational Challenges

Consider for example the emerging trend of shared mobility, specifically bike sharing systems (BSSs). This mobility trend has seen an exponential growth over the last few years, and over 700 cities actively operate automated bike sharing systems, deploying an estimate of 800,000 bikes worldwide [MD14]. BSS offers citizens a flexible, fast, and green alternative for mobility. This flexibility and ease of access for the end users come with many challenges that operators face on their side. Unpredictable and fluctuating demand, as well as irregular flow patterns of the bikes, leads to imbalance problems such as the unavailability of bikes or parking docks at stations (cf., Figure 1.2). A BSS user would experience an unsatisfactory quality of service when attempting to rent or drop off a bike at an empty or full station respectively. Thus, BSS operators deploy a fleet of vehicles (e.g., trucks) to redistribute bikes among stations and balance the system in order to guarantee a desirable service level. However, meeting this demand for bikes and free docks requires extensive logistic work resulting in high operational costs.

Or consider for example community sensing, a new paradigm for creating efficient and cost-effective sensing applications by harnessing the data of large populations of sensors. For instance, OpenSense [Ope] is a Swiss nationwide project, with the goal of creating a community-based solution for sensing the air quality and understanding the health impact of air pollution exposure. However, there are at least two major chal-
Chapter 1. Introduction

Challenges that one needs to tackle in the realm of this project (cf., Figure 1.3): (i) finding low-cost, light-weight sensors and new sensing modalities that can be incorporated easily into the daily lives of people; and (ii) designing privacy-aware mechanisms that can valuate and negotiate access to the participants’ sensing data in an incentive-compatible manner.

Last but not least, consider citizen science projects such as eBird or GalaxyZoo. These are crowd-powered systems through which people around the world volunteer to help scientific research. For instance, volunteers on GalaxyZoo have provided millions of annotations for classifying galaxies from telescopic images. However, the data collected from such platforms is often noisy, primarily due to a lack of expertise among the participants. This problem is further exacerbated by the heterogeneity of these participants (cf., Figure 1.4). As the accuracy of the annotated data is often crucial, the problem of tackling noise is of vital importance, and it is one of the main challenges faced by microwork based crowdsourcing platforms.

1.2.2 Engaging and Empowering the Crowd

In order to tackle the identified challenges, our approach is to engage and empower the users of the system (referred to as crowd) with a goal towards building self-sustainable and intelligent systems. The specific protocols for doing so depend on the application domain at hand, the specific challenges we seek to tackle, and the resource constraints.

For instance, to tackle the above-mentioned imbalance problem in the bike sharing systems, we propose a novel crowdsourcing scheme that engages users in the bike repositioning process by providing them with alternate choices to pick or return bikes (cf., Figure 1.2).

For community sensing applications, we develop a new obfuscation protocol allowing users to participate in the system without revealing their true locations until recruited by the system. Furthermore, in the context of the OpenSense project, we study the feasibility of a new sensing modality, equipping users’ bikes with light-weight, low-cost air quality sensors (cf., Figure 1.3).

For reducing the annotation noise in citizen science projects, we explore a novel direction of research on teaching the participants by demonstration in order to improve their accuracy. That is, instead of designing models and methods for determining
participants’ reliability (a common approach in the existing literature), our approach is focused towards developing intelligent systems that teach participants to be more effective (cf., Figure 1.4).

1.2.3 Techniques at the Interplay of Learning and Incentives

As the core of our approach, we develop novel techniques at the interplay of learning and incentives. In this dissertation, our focus is on developing techniques that are both theoretically well-founded with strong provable guarantees and practically applicable. As we discussed above, our techniques encompass three facets of this interplay, namely: (i) Learning about Incentives, (ii) Incentives for Learning, and (iii) Learning as an Incentive. In the next section, we give an overview of the learning algorithms and mechanisms we have developed in this dissertation.

1.3 Overview of Main Contributions

In this dissertation, we develop three key algorithms/mechanisms, each covering one of the facets of the interplay between learning and incentives. We provide strong theoretical guarantees for them and then apply our results to real-world applications. Table 1.1 summarizes the main technical results along with their guarantees. We give a more detailed overview of these results in the rest of this section.

1.3.1 Learning about Incentives

with Applications to Balancing Bike Sharing Systems

In this direction of research, we focus on developing new machine learning techniques to model and learn about users’ preferences as well as their skills/expertise, with the goal of learning to offer incentives to the users more efficiently. In this dissertation, we present our technical results for one specific problem concerning the design of an efficient pricing mechanism for offering monetary incentives to the users, under strict budget constraints. Some of the main challenges in designing an efficient pricing mechanism are due to constraints on the allocated budget and a dynamic pool size of users to make offers to. Furthermore, the underlying distribution of the users’
Chapter 1. Introduction

Figure 1.2: Learning about incentives with applications to balancing bike sharing systems (Part II of this dissertation). These results are based on Singla and Krause [SK13b] and Singla et al. [Sin+15a].

private costs is unknown to the mechanism and needs to be learned via interacting with them. Also, these users may act strategically in their self-interest, and we need to design incentive-compatible (truthful) mechanisms to be robust against any strategic behavior of the users.

The applicability of the existing mechanisms in the literature for this problem of budgeted procurement is limited: either they are designed for a restricted bidding model (whereby users are required to bid their costs of participation), or they perform poorly in practice. Instead of soliciting the users’ costs, we consider a more natural setting of the posted-price model where users are offered a take-it-or-leave-it price offer. We develop a novel posted-price online mechanism, BP-UCB, for online budgeted procurement, which is guaranteed to be budget feasible, achieves near-optimal utility for the system, is incentive-compatible (truthful) for the users, and makes minimal assumptions about the distribution of users’ private costs. On the theoretical side, we present a novel mathematical analysis which exploits a link between procurement auctions and multi-armed bandits—a classical problem in online learning and experimental design—to prove regret bounds for the mechanism. The main technical challenges arise because of the budget constraints that leads us to develop new tools and analysis. Our analysis further yields insights into an explicit separation of the regret in terms of the wasted budget through overpayment and rejected offers through underpayment. This mechanism is of independent interest and useful for other crowdsourcing
platforms with monetary incentives like Mechanical Turk.

Our pricing mechanism BP-UCB is the key learning component in the crowdsourcing scheme that we develop in this dissertation to tackle the imbalance problem in bike sharing systems (cf., Figure 1.2). Our crowdsourcing scheme incentivizes users in the bike repositioning process by providing them with alternate choices to pick or return bikes in exchange for monetary rewards. In collaboration with a bike sharing company ElectricFeel [Ele], we deployed the proposed incentives scheme through a smartphone app among users of a large-scale bike sharing system operated by the public transport company MVGmeinRad [Mvg] in the city of Mainz, Germany. To our knowledge, this is the first dynamic incentives scheme for bikes redistribution ever deployed in a real-world bike sharing system.

1.3.2 Incentives for Learning

with Applications to Community-Based Air Quality Sensing

Next, we explore the role of designing and engineering incentives for collecting data by incentivizing users to solving complex sensing and computation tasks. This, in turn, helps to improve the performance of large-scale machine learning systems, for instance, by providing access to training data at scale for learning tasks such as computer vision and natural language processing.

The key research question that we tackle is: How can we design privacy-aware, incentive-compatible mechanisms for learning, sensing, and optimal information gathering? We study this question in the context of community sensing applications. In the existing literature, these applications have been studied extensively by casting the underlying problem of discrete optimization as a sensor selection problem. However, these existing results do not directly apply to our setting whereby these sensors correspond to (or are held by) strategic users: we need a new suite of optimization techniques that can account for privacy concerns and are robust against the strategic behavior of these users.

Our approach towards recruiting participants for community sensing is based on the idea that users are willing to share certain private information (sensor data, private location, etc.) if compensated in terms of their utility loss, in return of, e.g., monetary or other forms of incentives. We model the users as strategic agents who are willing
Privacy concerns and bulky sensors

Inference of locations from GPS traces

Obfuscation protocol and new sensing modalities

Sensors mounted on trams and buses

Figure 1.3: Incentives for learning with applications to community-based air quality sensing (Part III of this dissertation). These results are based on Singla and Krause [SK13a].

...to negotiate access to certain private information, aiming to maximize the monetary incentives they receive in return. In our approach, the key idea towards protecting users’ privacy is to allow them to communicate only their obfuscated locations until recruited by the system. Such an obfuscation protocol is practically easy to implement, for example, via a smartphone app whereby users declare their obfuscated locations (or regions/trajectories) for which they could sense data. However, this leads to new technical challenges: the system now has to deal with the uncertainty caused by the obfuscation of users’ sensing profiles.

Our key insight is that privacy tradeoffs in community sensing can be cast as an adaptive submodular optimization problem for a wide range of sensing applications dealing with uncertainty. We then design a novel incentive-compatible (truthful) mechanism, SEQTGREEDY, for adaptive submodular maximization under budget constraints. We prove that our proposed mechanism achieves near-optimal utility for a large class of sensing applications. This mechanism is general, and of independent interest, both theoretically as well as for other potential applications (e.g., viral marketing) dealing with information acquisition from strategic agents under uncertainty.

We demonstrate the effectiveness of our approach in a case study of air quality monitoring, using data gathered from a user study. In this case study, we surveyed users to understand their willingness as well as bids to participate in such community sensing systems, and we use the collected data to perform realistic experiments. We compare
1.3. Overview of Main Contributions

against some natural baselines, allowing us to quantify the loss in utility in order to respect constraints of privacy and truthfulness for our mechanism.

Finally, in the context of the OpenSense project, we study the feasibility of a sensing modality whereby we installed air quality sensors on a bike. In the process of searching for low-cost and light-weight sensors, we study the CO2 gas sensors from the NODE+ platform [Nod] and perform calibration experiments at EMPA [Emp] (cf., Figure 1.3). We then collect data from these sensors mounted on a bike in the city of Zurich and learn a spatial model of CO2 concentration from this data. These models are important in guiding the system of how dense a sensing network work need to be and to define realistic utility functions quantifying the value of the selected sensors.

We believe that this integrated approach connecting privacy, utility, and incentives provides an important step towards developing practical, yet theoretically well-founded techniques for community sensing.

1.3.3 Learning as an Incentive

with Applications to Biodiversity Monitoring via Citizen Science

This direction of research tightly couples the interplay of learning and incentives: we explore novel techniques to leverage learning as an incentive in itself with the goal of improving the labeling accuracy in citizen science projects. For the problem of tackling noise in crowdsourcing services including citizen science, most of the work in existing literature has focused on methods for combining labels from many participants or in designing control measures by estimating the participants’ reliabilities through gold standard questions. We explore an orthogonal direction of teaching participants in crowdsourcing services in order to improve their accuracy. Our main idea is that the participants—especially volunteers and amateur scientists in citizen science projects—have strong incentives to learn to increase their knowledge. Furthermore, learning acts as an incentive to improve performance accuracy, leading to higher monetary payments and better ranking in the leaderboard in other crowdsourcing platforms.

The specific research question that we study is: How should we present training examples to learners to teach them classification rules? This is a natural problem when training participants for labeling tasks in citizen science projects or crowdsourcing platforms, and is also motivated by challenges in data-driven online education.
Figure 1.4: Learning as an incentive with applications to educating citizen scientists for biodiversity monitoring (Part IV of this dissertation). These results are based on Singla et al. [Sin+14b].

We propose a *noise-tolerant* stochastic model of human learners, modeling them as switching among hypotheses via a probabilistic model based on observed feedback. Our model generalizes existing *noise-free* models of teaching in order to increase robustness: we show that these existing models are brittle in practice via experiments on human subjects. We then develop STRICT, a novel teaching algorithm that shows a sequence of training examples to learners in order to steer them towards the true hypothesis. We theoretically analyze our approach, proving strong approximation guarantees and teaching complexity results.

For experiments, we consider different three tasks of classifying animal species, an important component in several citizen science projects such as the eBird project for biodiversity monitoring (cf., Figure 1.4). We demonstrate the effectiveness of our model and STRICT policy on these three tasks via teaching the participants from a crowdsourcing platform. More generally, our approach goes beyond solving the problem of teaching participants in crowdsourcing services. With the recent growth of online education and tutoring systems (e.g., Coursera), algorithms such as STRICT can be envisioned to aid in data-driven online education.
1.3.4 Summary of Theoretical Results

Table 1.1 summarizes the main algorithms and mechanisms presented in this dissertation, along with their theoretical guarantees.

<table>
<thead>
<tr>
<th>Part</th>
<th>Algorithm/ Mechanism</th>
<th>Theoretical Guarantees</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part II: Learning about Incentives</td>
<td>BP-UCB</td>
<td>• Online posted-price mechanism</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• No-regret guarantees, truthfulness, and budget-feasibility</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Theorem 2 from Singla and Krause [SK13b]</td>
</tr>
<tr>
<td>Part III: Incentives for Learning</td>
<td>SEQTGReedy</td>
<td>• Privacy-aware mechanism for adaptive submodular maximization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Near-optimal utility, truthfulness, and budget-feasibility</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Theorems 1, 2, 3, and 4 from Singla and Krause [SK13a]</td>
</tr>
<tr>
<td>Part IV: Learning as an Incentive</td>
<td>STRICT</td>
<td>• Algorithm for selecting training examples to teach participants</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Approximation guarantees on the convergence to a desired error rate and teaching</td>
</tr>
<tr>
<td></td>
<td></td>
<td>complexity results</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Theorems 1 and 2 from Singla et al. [Sin+14b]</td>
</tr>
</tbody>
</table>

Table 1.1: Algorithms and mechanisms presented in this dissertation, along with their theoretical guarantees.
Chapter 1. Introduction

1.4 Organization of this Dissertation

The main content of this dissertation is organized in three parts: Parts II, III, and IV. Each of these parts consists of three components: (i) an overview chapter to begin (Chapters 3, 7, and 10), (ii) the next chapter presenting the developed algorithms and mechanisms with theoretical analysis (Chapters 4, 8, and 11), and (iii) one or two subsequent chapters with experimental results (Chapters 5, 6, 9, and 12). In the three overview chapters, we also provide a summary of our relevant research work that is not covered in this dissertation. We review the related work separately within each chapter.

In Part II, we explore the first facet of the above-mentioned interplay: Learning about Incentives. Part II begins with Chapter 3 presenting a brief overview of the results to be discussed in subsequent chapters. Then, Chapter 4 presents our online learning mechanism BP-UCB and provides the no-regret guarantees of BP-UCB. Chapter 5 presents the detailed experimental evaluation of our proposed mechanisms as well as comparison with state of the art baselines on data collected from the Mechanical Turk platform. In Chapter 6, we apply this mechanism for tackling the imbalance problem in bike sharing systems. We then report on the results of real-world deployment of the proposed incentives scheme in the city of Mainz, Germany.

In Part III, we study the aspect of Incentives for Learning and consider the application of community-based air quality sensing. Chapter 8 presents our privacy-aware mechanism SeqTGreedy with formal analysis by exploiting adaptive submodularity properties of the model. In Chapter 9, we demonstrate the effectiveness of our approach via extensive experiments on real-world data.

Part IV explores Learning as an Incentive with applications to educating citizen scientists for biodiversity monitoring. Chapter 11 presents our teaching algorithm STRICT with formal guarantees on its near-optimality. In Chapter 12, we discuss the experimental results based on user studies for teaching participants to classify animal species.

Lastly, in Chapters 13 and 14, we conclude this dissertation and discuss a few interesting directions for future work. The proofs of all the theoretical results are given in Appendices at the end of this dissertation.
Research Publications and Collaborations

The research work done during the doctoral studies resulted in several fruitful collaborations with other researchers and industrial partners. In Section 2.1, we provide a list of our research publications that form the basis of the content presented in this dissertation, and we acknowledge the contributions of our collaborators.

Then, in Section 2.2, we provide a list of our other research publications tackling problems relevant to the dissertation’s topic. However, we do not cover them in detail in this dissertation, primarily to keep the content of this dissertation concise.

2.1 Publications Covered in this Dissertation

This dissertation covers material primarily from the following four publications.

Chapter 2. Research Publications and Collaborations


Acknowledging the Contributions

Prof. Andreas Krause was involved in all four above-mentioned publications, providing valuable suggestions, ideas, and technical mentoring. Below, we provide specific details of the contributions from other co-authors and collaborators.

Contributions from Co-Authors [Sin+14b]

This work was done in close collaboration with Dr. Gábor Bartók and Dr. Amin Karbasi, who helped in formalizing the problem and contributed to the theoretical analysis. Ilija Bogunovic developed the web app used for performing the experiments that involved teaching participants recruited from a crowdsourcing platform.

Collaboration with ElectricFeel [Sin+15a]

The work on balancing bike sharing systems in Singla et al. [Sin+15a] was done in collaboration with ElectricFeel (Zurich, Switzerland) [Ele] and the public transport company MVGmeinRad (Mainz, Germany) [Mvg]. Marco Santoni did his M.Sc. thesis on this topic supervised by Prof. Andreas Krause, Dr. Gábor Bartók, and Adish Singla, and in collaboration with ElectricFeel. Marco Santoni conducted the experiments and developed the smartphone app as part of his thesis. Pratik Mukerji, CTO of ElectricFeel, developed the required APIs so that our mechanisms can access the predictions about traffic load at the bike stations based on proprietary algorithms developed by ElectricFeel. The team at MVGmeinRad helped us to conduct the survey among their users in the city.
of Mainz. ElectricFeel and MVGmeinRad allocated the budget for the pilot deployment needed to offer the monetary incentives to the users. Furthermore, they helped us deploy the smartphone app in the city of Mainz, and they also allowed us to transfer the monetary incentives to the accounts of participating users.

2.2 Publications Not Covered in this Dissertation

Next, we provide a list of our other research publications relevant to the topic of this dissertation; however, we do not cover them in detail in this dissertation.

Learning Hemimetrics Encoding Switching Preferences


Privacy-Aware Information Gathering on the Web


Learning Users’ Skills/Expertise and Allocating Tasks under Skill-Matching Constraints

Chapter 2. Research Publications and Collaborations


**Budgeted Learning in Crowdsourcing**


**Novel Applications of Crowdsourcing: From Building Taxonomies to Image Collection Summarization**

• Yuyin Sun, Adish Singla, Dieter Fox, and Andreas Krause. “Building Hierarchies of Concepts via Crowdsourcing”. In: *Proc. International Joint Conference on Artificial Intelligence (IJCAI)*. 2015. [Sun+15]. (primarily based on the dissertation work of Yuyin Sun.)

• Yuyin Sun, Adish Singla, Tori Yan, Andreas Krause, and Dieter Fox. “Evaluating Task-Dependent Taxonomies for Navigation”. In: *Proc. AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*. 2016. [Sun+16]. (primarily based on the dissertation work of Yuyin Sun.)

• Adish Singla, Sebastian Tschiatschek, and Andreas Krause. “Noisy Submodular Maximization via Adaptive Sampling with Applications to Crowdsourced Image Collection Summarization”. In: *Proc. Conference on Artificial Intelligence (AAAI)*. 2016. [STK16b].
Part II

Learning about Incentives
In Part II of this dissertation, we explore the first facet of the interplay between learning and incentives, namely *Learning about Incentives*. We focus on developing new machine learning techniques for learning users’ preferences and designing efficient incentive mechanisms. Some of the key questions in this direction of research include:

*How can we simultaneously learn about users’ preferences (i.e., desired incentives) as well as their abilities and skills, while using the obtained information in order to maximize effectiveness?*

We address these fundamental questions by building on results from online learning (in particular, multi-armed bandit methods in structured spaces), budget feasible mechanism design, and by designing new models of users’ preferences.

One specific problem that we focus on in the subsequent chapters concerns designing an efficient pricing mechanism for offering monetary incentives to the users, under strict budget constraints. Our work is motivated by crowd-powered systems where monetary rewards are the primary incentives to engage the users for performing some tasks. In Chapter 4, we address this problem and present an online *posted-price* mechanism BP-UCB using the approach of regret minimization in online learning. In Chapter 5, we carry out extensive experiments to compare the performance of our proposed mechanism with optimal benchmarks as well as state of the art mechanisms. Apart
Chapter 3. Overview of Part II

from experimenting with simulated users’ cost distributions, we perform experiments using data gathered from a user study on the Mechanical Turk platform to demonstrate the effectiveness of our approach on real-world inputs. In Chapter 6, we tackle the imbalance problem across stations in bike sharing systems (BSSs)—a real-world problem faced by the operators of BSSs posing high operational costs and extensive logistic work. To address this problem, we design a novel crowdsourcing scheme that incentivizes the users in the bike repositioning process. A key component of this incentives system is the dynamic pricing mechanism based on BP-UCB. We deployed the proposed system through a smartphone app in the city of Mainz, Germany and we provide results from this pilot deployment.

3.1 Relevant Research Problems and Our Contributions

In the rest of this chapter, we provide a summary of our work that tackles interrelated research problems relevant to the aspect of Learning about Incentives. These results are not covered in the rest of this dissertation, primarily to keep the content of this dissertation concise.

3.1.1 Learning Hemimetrics Encoding Switching Preferences [STK16a]

Recall that in the above-mentioned application of bike sharing systems (cf., Chapter 6 for details), our key idea is to steer users’ actions by providing them alternate choices to pick or return bikes in exchange for monetary incentives. However, our proposed mechanisms do not explicitly distinguish the type of these choices. For instance, consider two stations in a city: station \( i \) located near the city center and station \( j \) in a suburb. Then, the private cost of a user to switch from her default choice \( i \) to an alternate choice \( j \) could be much more compared to switching from default choice \( j \) to \( i \). Encoding and learning such fine-grained users’ preferences across different choices forms the basis of our work in Singla, Tschiatschek, and Krause [STK16a].

In Singla, Tschiatschek, and Krause [STK16a], we introduce a novel framework of encoding users’ preferences across different choices via hemimetrics, i.e., a relaxed form of a metric, satisfying only non-negativity constraints and triangle inequalities. These choices take the form of \( n \) items available in a marketplace (e.g., items could corre-
3.1. Relevant Research Problems and Our Contributions

spond to different apartments on rental services like Airbnb, or restaurants in a recommendation system such as Yelp). Our goal is to learn the pairwise distances $D_{i,j}$ representing the private cost of a user for switching from her default choice of item $i$ to item $j$. Knowledge about this type of preferences can be used in e-commerce applications for marketing or for maximizing social welfare, e.g., by persuading users to change their decisions [KG09; Sin+15a]. In our work, the interaction with the users takes the form of a binary query, motivated by the posted-price model in marketplaces—our mechanism BP-UCB developed in this dissertation is also based on this model (cf., Chapter 5 for details).

We then study the problem of learning an underlying hemimetric to a desired accuracy while minimizing the sample complexity (i.e., the number of user queries). We propose an active learning algorithm that substantially reduces this sample complexity by exploiting the structural constraints on the version space of hemimetrics. The two key techniques used in the construction of our algorithm are novel projection techniques for tightening the lower and upper bounds on the solution space and a non-myopic (non-greedy) query policy. We provide a thorough analysis of the sample complexity and runtime of our algorithm. Furthermore, our experimental results on a dataset from the Yelp Dataset Challenge (round 7) [Yel] show substantial improvements over baseline algorithms in line with our theoretical findings.

3.1.2 Learning Skills and Expertise [Sin+15c]

Crowdsourcing and human computation are being employed in sophisticated projects that require the solution of a heterogeneous set of tasks. With the increasing complexity of tasks that are crowdsourced, as well as enterprises outsourcing their work, the need to hire skilled participants with an eye to considerations of complementarity and coordinative efforts in a collaboration around problem solving is becoming important. Online platforms are offering new capabilities to deal with the rise of expertise-driven crowdsourcing. For example, the Upwork platform provides opportunities for freelancers to do self-assessments via taking tests on a voluntary basis. It also provides support for recruiters to conduct interviews or perform online tests for job applicants.

In our work in Singla et al. [Sin+15c], we explore the challenge of composing or hiring an effective team from an available pool of applicants for performing tasks required for such projects on an ongoing basis. How can one optimally spend budget to learn
the expertise of participants as part of recruiting a team? How can one exploit the similarities among tasks as well as underlying social ties or commonalities among the participants for faster learning? We tackle these decision-theoretic challenges by casting them as an instance of online learning for best action selection with side-observations. We evaluate our methodology on simulated problem instances using data collected from the Upwork platform.
Truthful Incentives via Online Learning Mechanisms

In this chapter, we tackle the problem of designing an online pricing mechanism for offering monetary incentives to the strategic users, when operating under strict budget constraints and limited pool size of the users to interact with. This is a natural problem arising in various crowdsourcing marketplaces like Mechanical Turk and CrowdFlower. Furthermore, in other systems like online social networks, users can be easily compensated via monetary incentives for performing some tasks, for instance, participating in a viral marketing campaign or performing a survey. When building new crowd-powered systems, monetary incentives are often easier to design, instead of, for instance, social badges or leaderboards, as they provide immediate tangible value to the users—this is one of the primary motivation for using monetary incentives in our crowdsourcing scheme to incentivize users in the bike repositioning process, cf., Chapter 6.

Considering, for instance, popular microwork based crowdsourcing platforms like Mechanical Turk, task requesters use monetary payments to procure the crowd of workers to perform micro-tasks. In these systems, the requester generally has a limited budget for the task and needs to come up with a payment scheme for users (crowd workers in this case) in order to maximize the utility derived from the task. For users,
the main goal is to maximize their individual profit by deciding which tasks to perform and at what price. Designing optimal pricing policies and determining the right monetary incentives is central to maximizing requester’s utility and users’ profits. However, often rules of thumb are used to price tasks, for instance, on Mechanical Turk platform, requesters would set a fixed price for the tasks chosen arbitrarily via some platform guidelines. Clearly, overpricing the tasks would result in inefficient use of the requester’s budget, whereas underpricing could lead to task starvation because of unavailability of the users willing to participate.

Some of the key challenges in designing an efficient pricing mechanism are due to strict budget constraints, unknown distributions of the private costs of users, and strategic aspects as users may act in their self-interest to maximize their personal profits. In this light, the key questions that we seek to address include:

\[ \text{How can one design optimal pricing policies under strict budget constraints of the task requester? How can users communicate and negotiate the price with requester? How would the system behave if users act strategically by misreporting their costs for their benefit?} \]

These are some of the questions that naturally come to mind while studying incentive structures for online crowdsourcing marketplaces, yet they are not well understood. We address some of these questions in this chapter, and we begin with a high-level overview of our approach in the next section.

**Organization of this Chapter**

This chapter is organized as follows. In Section 4.1, we give a high-level overview of our approach. Section 4.2 formalizes the problem addressed in this chapter. In Section 4.3, we develop our mechanism BP-UCB and then prove the regret bounds in Section 4.4. Section 4.5 discusses the related work and Section 4.6 presents our conclusions.

**Relevant Publication**

Results presented in this chapter are published in Singla and Krause [SK13b].
4.1 Overview of Our Approach

In this section, we give a high-level overview of our approach and summarize our main contributions.

4.1.1 Pricing Models and Interaction with Users

A simple pricing scheme, often used in practice, is to use a predetermined single fixed price. Especially, the current crowdsourcing platforms like Mechanical Turk offer a limited capability to the task requesters in designing the pricing policies, mostly limiting them to this fixed-price model. One way to set prices under such models is to estimate users’ costs via a market analysis and then compute an optimal fixed price which would maximize the utility. However, there are many difficulties in inferring this optimal fixed price, including the high cost of market surveys, the dynamic and online nature of the marketplaces, the inexperience of the requester and challenges in soliciting true costs from users because of their self-interest. An alternate approach is to use tools of online procurement auctions where users can bid on the price they are willing to receive, and the requester’s mechanism can decide on the allocation and prices to be paid to them. In this bidding model, mechanisms need to be truthful: it should be a dominant strategy for rational users to bid their true cost. However, communicating these true costs to the requester may be challenging in real-world settings. The user may typically not trust the requester and understand the mechanism to reveal their true cost or the cost may not even be known to a user and perhaps difficult to determine. Instead of soliciting the users’ costs, an often more natural setting is the posted-price model where users are offered a take-it-or-leave-it price offer. The mechanism interacts with each user once in a sequential manner and adjusts the offered price from past responses of the users. We note that posted-price mechanisms are by design guaranteed to be dominant strategy incentive-compatible (truthful), cf., Figure 4.1.

4.1.2 Our Results

In this chapter, we present a novel posted-price mechanism, BP-UCB, for online budgeted procurement, which is guaranteed to be budget feasible, achieve near-optimal
4.2 The Model

We now formalize the problem addressed in this chapter.

4.2.1 Task Requester and Users

There is a principal agent, the requester, who is interested in procuring a set of users to perform a task. In this chapter, we consider a generic notion of the requester,
4.2. The Model

users, and the task, capturing various motivating applications mentioned above—we will apply our approach to two specific applications in the next two chapters (cf., Chapters 5 and 6). A task is composed of atomic assignments, which can be performed by individual users. The requester has a budget \( B > 0 \) and a utility function over completed assignments. In this work, we assume that each assignment performed by a user has unit value, thus the requester wishes to maximize the number of completed assignments subject to the budget constraint. There is a finite pool of users, denoted by \( W \). Each user \( w_i \in W \) is associated with a private cost \( c_i \in \mathbb{R}_{\geq 0} \) for performing an assignment. When considering the bidding model, let \( b_i \in \mathbb{R}_{\geq 0} \) be their bid or reported cost—we are interested in truthful mechanisms where it is a dominant strategy for user \( w_i \) to report \( b_i = c_i \).

We assume that costs have known bounded support, i.e., \( c_i \in [c_{\text{min}}, c_{\text{max}}] \) where \( c_{\text{min}} \) and \( c_{\text{max}} \) are the parameters of the problem, scaled such that \( c_{\text{min}} > 0 \) and \( c_{\text{max}} \leq 1 \). We note that the assumption of bounded costs naturally holds in online crowdsourcing platforms like Mechanical Turk, which generally enforce a publicly known minimal and maximal allowed payment for the assignments. We will keep the range of the costs fixed, and consider varying the budget, which ensures that this assumption can be made without loss of generality. Also, the number of assignments per user is normally set to one in Mechanical Turk platform. Furthermore, we assume that there are at least \( N \) users where \( N \geq \frac{B}{c_{\text{min}}} \). We note that limiting the pool size to a much smaller number would lead to further constraints in our mechanism design in addition to the budget constraint and is beyond the scope of this work. However, having a very large, essentially infinite pool of users would make the problem trivial as the mechanism can offer lowest possible prices without any overall loss of utility. We further discuss this issue in Section 4.6.

4.2.2 Online Arrival of Users

We are interested in online settings where users arrive one at a time. We focus on stochastic arrival of users, where their costs are i.i.d. sampled from a distribution \( f \). In this i.i.d model, we let \( F : [c_{\text{min}}, c_{\text{max}}] \to [0, 1] \) denote the cumulative distribution function (CDF) of costs associated with the users. Note that the stochastic arrival assumption may be violated in real-world services/marketplaces. This could be because of various factors, for example, if the users’ value of service increases over time and
hence so does their cost. This non-stochastic setting is often called the oblivious adversary model. In the next chapter (cf. Chapter 5), we will empirically evaluate the robustness of our mechanisms in the presence of such adversarial noise.

4.2.3 Optimal Benchmarks

Consider an (unrealistic) offline mechanism with complete access to the pool of users’ true costs. The maximal utility in this setting can be achieved by sorting the users by their increasing costs and offering each user their true cost until the budget is exhausted. We denote this benchmark by OPT-Var, i.e., the optimal variable price benchmark. An alternate benchmark of interest is a mechanism that is limited to offer a single fixed price to all users, though this price is computed optimally assuming full knowledge of users’ costs. We denote this by OPT-Fix. Note that these benchmarks are offline, untruthful, and assume full knowledge of the users’ costs. It seems natural to compare our online truthful mechanisms to the optimal truthful mechanism in offline settings. The utility of any single priced offline truthful mechanism is bounded by OPT-Fix and Goldberg, Hartline, and Wright [GHW01] shows that the performance of OPT-Fix is close to that of OPT-Var. Further, recent results in procurement auctions from Singer [Sin10] and Badanidiyuru, Kleinberg, and Singer [BKS12] show that OPT-Fix is only a factor of 2 away from OPT-Var for modular as well as symmetric submodular functions and this is the best approximation that any truthful mechanism can achieve. Therefore we will compare our mechanisms against this benchmark. This optimal offline fixed price denoted by $p^*$, as illustrated in Figure 4.2, is given by:

$$p^* = \arg \max_p \min \left\{ \frac{F(p)}{N}, \frac{B}{np} \right\} \text{ s.t. } p \in [c_{\min} \ldots c_{\max}].$$

4.2.4 Utility and Regret

For a fixed budget $B$, let $U(M, B)$ denote the expected utility of mechanism $M$ and $U(p, B)$ denote the expected utility of fixed price $p$. In the regret minimization framework, we are interested in comparing the regret w.r.t. the best single price $p^*$ offered in hindsight. The expected regret of mechanism $M$ is given by

$$R_M(B) = U(p^*, B) - U(M, B),$$
4.3. Our Mechanisms

We begin by describing the high-level ideas behind our mechanisms. Then, we design BP-DGREEDY for the bidding model and subsequently extend it to arrive at our main mechanism, BP-UCB, for posted prices. In the next section (cf., Section 4.4), we will analyze the mechanisms and prove the regret bounds.

4.3.1 Background and High-Level Ideas

Background on Classical Multi-Armed Bandits

In the classical multi-armed bandits (henceforth MAB) setting [BM07; ACBF02], there are $K$ independent choices (arms) associated with unknown reward distributions. A
MAB algorithm operates in discrete timesteps (rounds) and pulls an arm in each round to get a stochastic reward associated with that arm. The algorithm needs to explore by experimenting with potentially suboptimal arms so as to learn about the optimal arm. Meanwhile, to maximize the reward, it has to exploit its learning by pulling the arm that appears best. The goal of the algorithm is to minimize the regret by quickly converging to the optimal arm.

Learning the Cost Curve and Connection to Multi-Armed Bandits

The main challenge in deciding the payments in our problem is the unknown distribution of the users’ cost (cost curve). The mechanism interacts with users sequentially in discrete timesteps denoted by $t$, offering a price $p^t$ at each timestep to user $w^t$ and adjusting the estimates of the cost curve based on observed feedback. In order to cast our problem in the MAB framework, we discretize the prices by creating a set of $K$ price arms using a multiplicative factor of $(1 + \alpha)$, where $\alpha$ is a parameter of the mechanism, similar to Blum et al. [Blu+03] and Babaioff et al. [Bab+12], as illustrated in Figure 4.2. For these $K$ arms, we maintain $F^t_i$ as an estimate of the CDF of users’ costs for price $p^i$ at time $t$. The mechanism stops execution when the budget or the pool of users is exhausted. At each timestep, our mechanisms will pick the arm $i^t$ based on some optimization criterion. Unfortunately, the presence of the budget constraint breaks the standard MAB algorithms: The optimal arm in terms of utility is the one corresponding to the maximal price, though it would quickly exhaust the available budget, leading to diminished utility. Further, we can exploit the fact that the price arms are correlated in our case: Acceptance at an offered price means acceptance for all more expensive arms and rejection at an offered price means rejection for all cheaper arms.

Additional Notation

We need to introduce some more notation in order to describe our mechanisms. Let $i' \in \{1, \ldots, K\}$ denote the index of the optimal price among the $K$ discretized prices, $p'$ denote the corresponding price and $F'$ the value of function $F_p = F(p')$. As illustrated in Figure 4.2, $i'$ is given by:

$$i' = \arg \max_i \min \left\{ F_i, \frac{B}{N \cdot p_i} \right\} \forall i \in [1 \ldots K].$$
4.3. Our Mechanisms

**Algorithm 4.1: Mechanism BP-DGREEDY**

1. **Parameters**: $B; N; \alpha = (0, 1]; c_{\min}, c_{\max};$
2. **Initialize**:
   - **Prices**: $p_0 = c_{\min}; p_i = (1 + \alpha) \cdot p_{i-1} \cdot p_K = c_{\max};$
   - **Variables**: time $t = 0; \text{budget } B^t = B; \text{utility } U = 0;$
   - **Value estimates**: $F^t_i = 0;$

begin
   while $B^t > c_{\min} \& t < N$ do
      $i^t = \arg \max_i V^t_i \text{ s.t. } p_i \leq B^t;$
      /* ties broken by picking lowest $i */;
      Offer price $p^t = p_{i^t}$ to user $u_{i^t};$
      Observe bid $b^t;$
      $\forall i$, Update $F^t_i = F^t_i + \frac{(S^t_i - F^t_i)}{(t+1)};$
      Set $U = U + S^t_i; B^{t+1} = B^t - p^t \cdot S^t_i; t = t + 1;$
   end

Output: $U$

We let $B^t$ be the budget remaining at time $t$, $N^t_i$ be the number of times $p_i$ price has been offered, $S^t_i$ be the indicator random variable indicating $b^t \leq p_i$ and $T$ be the total number of rounds of execution of the mechanism (until the budget is exhausted or all the users have been seen by the mechanism). Let us define $V^t_i = \min \{ F^t_i, \frac{B}{N^t_i p_i} \}$, $V^t_i$ be the estimate of $V_i$ at round $t$. We further use $\Delta_i = V^t_i - V_i$.

4.3.2 Mechanism BP-DGREEDY for Bidding Model

We begin by designing a mechanism, BP-DGREEDY (deterministic greedy for budgeted procurement), for the bidding model (see Mechanism 4.1) in order to gain insights into the problem and develop mathematical foundations for the posted-price model. The mechanism solicits the user’s bid $b^t$ about service cost and then offers them a price $p^t$, based on past observations of users’ bids, which the current user can accept or reject. A natural approach towards ensuring truthful bids is to make the offered price
Algorithm 4.2: Mechanism BP-UCB

1 Parameters: \( B; N; \alpha = (0, 1]; c_{\min}; c_{\max}; \)

2 Initialize:
   - Prices. \( p_0 = c_{\min}; p_i = (1 + \alpha) \cdot p_{i-1}; p_K = c_{\max}; \)
   - Variables. time \( t = 0; \) budget \( B^t = B; \) utility \( U = 0; \)
   - Value estimates. \( N_i^t = 0; F_i^t = 0; \)

begin
   while \( B^t > c_{\min} \) & \( t < N \) do
      \( \tilde{F}_i^t \) = \( F_i^t + \sqrt{\frac{2 \cdot \ln(t)}{N_i^t}}; \)
      \( \tilde{V}_i^t \) = \( \min \{\tilde{F}_i^t, \frac{B}{N_p^t}p_i\}; \)
      \( i^t = \arg \max_i \tilde{V}_i^t \) s.t. \( p_i \leq B^t; \)
      /* ties broken by picking lowest \( i */ ;
      Offer price \( p^t = p_{i^t} \) to user \( w^t; \)
      Observe acceptance decision \( y^t; \)
      Update \( F_i^t = F_i^t + (y^t - F_i^t) \frac{N_i^t}{(N_i^t+1)}; N_i^t = N_i^t + 1; \)
      Update \( U = U + y^t; B^{t+1} = B^t - p^t \cdot y^t; t = t + 1; \)
   end

Output: \( U \)

independent of the bid of the current user. Because of this truthfulness, we have \( b^t = c^t \), which makes this mechanism resemble online learning with full information (cf., [BM07; LW94; CB+97]). This intuitively means that the mechanism gets to compute response feedback it would have received for any possible action.

We now discuss how to pick the arm \( i^t \) to make a price offer \( p^t \) independent of \( b^t \). The intuition is simple: BP-DGREEDY just tracks the expected utility \( V_i^t \) of the arms based on the estimated \( F_i^t \) and offers the price corresponding to the best arm at time \( t \). Based on the observed bid, it updates the estimates of \( F_i^t \) for all the arms by simply maintaining the average response of acceptance at arm \( i \).
4.4. Performance Analysis

4.3.3 Mechanism BP-UCB for Posted-Price Model

Next, we present our main mechanism, BP-UCB (UCB for budgeted procurement), for the posted-price model (see Mechanism 4.2), where we get to see acceptance or rejection feedback only for the price offered. This limited feedback leads to a natural exploration-exploitation tradeoff as in MAB problems. We tackle this problem by modifying BP-DGREEDY, whereby we maintain upper confidence bounds on $F^t_i$, denoted by $\tilde{F}^t_i$, which are then used to optimistically estimate the values $\tilde{V}^t_i$. The mechanism then picks the best arm based on $\tilde{V}^t_i$ and offers a take-it-or-leave it price $p^t_i$. Based on the feedback from the user, it updates the parameters only for the arm $i^t$. This approach is inspired from the classical UCB1 algorithm [ACBF02]. However, the budget constraints make the analysis of regret bounds non-trivial.

We can further exploit the correlation between the arms, as a rejection response actually means rejection for all cheaper arms. Similar to Blum and Hartline [BH05], we can use this correlation to further improve the execution performance of the mechanism by keeping an estimate of the lower bound of the cost curve’s support and keeping all the arms (except one) below this estimate as inactive. This modification does not hurt the theoretical guarantees described in Section 4.4.

4.4 Performance Analysis

We now prove regret bounds for our mechanisms BP-DGREEDY and BP-UCB. We begin with the analysis of BP-DGREEDY and develop the mathematical tools that will be useful for the analysis of BP-UCB. A crucial challenge in dealing with the budget constraint lies in the fact that while higher prices are more effective since more users would accept the offer, they would quickly exhaust the budget leading to reduced overall utility.

4.4.1 Components Contributing to the Regret

There are essentially three components contributing to the regret. The first is the discretization of the prices: Since the mechanism does not have access to the optimal price $p^*$, $p'$ is the best price available. After accounting for the regret of discretization, we can consider an alternative mechanism $M'$ which has access to an additional arm.
corresponding to $p^*$. Considering $M'$, the second component of the regret is attributed to pulling arms with prices $p_i < p^*$ as cheaper arms are less effective and result in more rejected offers. The third component of the regret is attributed to pulling arms with prices $p_i > p^*$. Though these expensive arms are more effective than the price $p^*$, they overpay and quickly exhaust the budget. We formalize the above discussion in Lemma 4.1.

**Lemma 4.1.** The expected regret $R_M(B)$ of any mechanism $M$ can be expressed in terms of three components as follows:

$$R_M(B) < \left( \frac{B}{p^*} - U(p', B) \right) + \sum_{i:p_i < p'} \mathbb{E}[N_i^T] \cdot (F^* - F_i)$$

$$+ \sum_{i:p_i > p'} \mathbb{E}[N_i^T] \cdot \left( \frac{(p_i F_i - p^* F_i)}{p^*} \right) + \frac{c_{\min}}{p^*}$$

The proof is given in the Appendix A.1.

### 4.4.2 Regret Bounds for BP-DGREEDY

To obtain the desired regret bounds for mechanism BP-DGREEDY from Lemma 4.1, we need to bound $N_i^T$, as well as the regret of discretization. We bound the $N_i^T$ for an execution of BP-DGREEDY using the Chernoff-Hoeffding concentration inequalities as in Auer, Cesa-Bianchi, and Fischer [ACBF02]. By exploiting the ordering of arms, we are able to separately provide bounds for $N_i^T$ for arms with prices $p_i < p'$ and $p_i > p'$. For our analysis, we consider four separate cases based on whether $F_i$ is less or greater than $\frac{B}{N \cdot p'}$ and based on the relative ordering of $p'$ compared to $p^*$. This insight crucially simplifies the analysis, enabling us to use the tools from the original UCB1 analysis in bounding each one of these four cases separately. Theorem 4.1 provides the desired regret bounds for BP-DGREEDY.

**Theorem 4.1.** The expected regret of mechanism BP-DGREEDY is upper-bounded as follows:

$$R_{BP-DGREEDY}(B) < \frac{\alpha \cdot B}{p^*} + \sum_{i:p_i < p'} \frac{8 \cdot (F^* - F_i)}{\Delta_i^2}$$

$$+ \sum_{i:p_i > p'} \frac{(p_i F_i - p^* F_i)}{2 \cdot \Delta_i^2 \cdot p^*} + \frac{c_{\min}}{p^*}$$
4.4. Performance Analysis

The proof is given in the Appendix A.2. Next, we prove the no-regret property of mechanism BP-DGREEDY by using an appropriate choice of the discretization factor $\alpha$, similar to the choice made for the problem of online auctions with limited supply in Babaioff et al. [Bab+12].

**Corollary 4.1.** The expected average regret of mechanism BP-DGREEDY w.r.t. the budget size $B$ goes to zero asymptotically for an appropriate choice of $\alpha$.

**Proof.** By setting $\alpha$ to $O\left(\frac{\ln(B/\epsilon_{\text{min}})}{B}\right)$, the expected average regret of BP-DGREEDY w.r.t. $B$ in the limit $\lim_{B \to \infty}$ is given as:

$$
\lim_{B \to \infty} \mathbb{E}\left[\frac{R_{\text{BP-DGREEDY}}(B)}{B}\right] = \lim_{B \to \infty} O\left(\frac{\ln(B/\epsilon_{\text{min}})}{B}\right) = 0.
$$

\hfill \Box

### 4.4.3 Regret Bounds for BP-UCB

We now extend the analysis of BP-DGREEDY to BP-UCB. Theorem 4.2 provides the desired regret bounds for BP-UCB.

**Theorem 4.2.** The expected regret of mechanism BP-UCB is upper-bounded as follows:

$$
R_{\text{BP-UCB}}(B) < \frac{\alpha \cdot B}{p^*}
+ \sum_{i : p_i < p'} \left(8 \cdot \frac{\ln(B/\epsilon_{\text{min}})}{\Delta_i^2} + \frac{\pi^2}{2} + 1\right) \cdot (F^* - F_i)
+ \sum_{i : p_i > p'} \frac{\pi^2 \cdot (p_i \cdot F_i - p^* \cdot F^*)}{6 \cdot p^*} + \frac{\epsilon_{\text{min}}}{p^*}
$$

The proof is given in the Appendix A.3. Corollary 4.2 proves the no-regret property of the mechanism.

**Corollary 4.2.** The expected average of regret of mechanism BP-UCB w.r.t. budget size $B$ goes to zero asymptotically for an appropriate choice of $\alpha$.

**Proof.** The proof follows by using exactly the same arguments as in Corollary 4.1. \hfill \Box
4.5 Related Work

In this section, we give a background on multi-armed bandits problem, and then review some relevant literature on learning in online auctions.

4.5.1 Multi-Armed Bandits and Regret Minimization

The multi-armed bandits (MAB) problem is a natural formalism for studying settings where an agent repeatedly chooses among actions with uncertain rewards, and must trade exploration (gathering information about rewards) and exploitation (maximizing rewards obtained). A primary objective is to design no-regret algorithms which guarantee that the average regret approaches zero asymptotically over time when compared to the single best action in hindsight. MAB and regret minimization algorithms have been studied extensively and Blum and Monsour [BM07] gives a good overview. Auer, Cesa-Bianchi, and Fischer [ACBF02] introduces the UCB1 algorithm, which maintains an index (known as Upper Confidence Bound) on the actions and avoids explicit separation of exploration and exploitation by picking the action with the highest index. Kleinberg [Kle04], Auer, Ortner, and Szepesvari [AOS07], Besbes and Zeevi [BZ09b], and Srinivas et al. [Sri+10] extend this approach to handle complex (possibly infinite) action spaces. Recently, budgeted variants of the MAB problem, where actions have different known costs, have been considered [TT+10; TT+12b; TT+13]. Tran-Thanh et al. [TT+12a] solves the crowdsourcing task whereby the goal is to learn users’ effectiveness as part of exploration. In our setting, in contrast, the costs are unknown and the budget is utilized only in rounds when the offer is accepted by the user—none of the standard approaches apply to this setting.

4.5.2 Learning in Online Auctions

Competitive online auctions were introduced by Lavi and Nisan [LN00] and Bar-yossef, Hildrum, and Wu [ByHW02]. These results were further extended and improved by using insights from regret minimization algorithms [Blu+03; BH05; KL03]. Chawla et al. [Cha+10] further extend the online posted-price mechanisms for multi-parameter domains. Hajiahyai et al. [Haj+04] and Devanur and Hartline [DH09] study the auction problem with limited supply in the bidding model under stochastic
4.5. Related Work

arrival of the agents. Babaioff et al. [Bab+12] extends these results to the posted-price model, by using insights from MAB problems. Our mathematical analysis builds on the results of Babaioff et al. [Bab+12]. However, in contrast, we consider dynamic pricing for procurement (reverse) auctions under a budget constraint. The results from Kleinberg and Leighton [KL03] and Babaioff et al. [Bab+12] are not applicable to this setting. In fact, a straightforward application of the mechanisms of Kleinberg and Leighton [KL03] and Babaioff et al. [Bab+12] in our setting would simply offer the highest price, as that maximizes the utility of the action (acceptance of the price by the user), though quickly exhaust the budget (and incur large regret).

4.5.3 Online Procurement Auctions

Mechanisms for procurement auctions have been extensively studied. Earlier work [AT02; KKT05; Car+08] concerns the frugality of mechanisms with the goal of procuring a feasible solution to a complex problem while minimizing the budget spent. In contrast, we are interested in studying truthful budget feasible mechanisms initiated recently by Singer [Sin10; Sin11]. Recent research addresses various models of budget constraints including the online knapsack secretary problem [Bab+07] and the weighted secretary problem [Bab+09]. However, these are not directly applicable to truthful procurement mechanisms. Singer and Mittal [SM11] and Badanidiyuru, Kleinberg, and Singer [BKS12] study a problem that is perhaps most similar to ours: they develop mechanisms for budgeted procurement in the stochastic setting for the bidding and posted-price model respectively, and prove constant multiplicative bounds. In contrast, our mechanisms use the regret minimization framework, and we prove additive bounds on the regret. We note that mechanisms of constant multiplicative bounds could have arbitrarily poor performance in terms of additive regret. In our experimental evaluation in the next chapter (cf. Chapter 5), Singer and Mittal [SM11] and Badanidiyuru, Kleinberg, and Singer [BKS12] are also used as benchmarks for our experiments and our mechanism BP-UCB shows a substantial improvement over the state of the art mechanism of Badanidiyuru, Kleinberg, and Singer [BKS12].
4.6 Summary

In this chapter, we designed mechanisms for online budgeted procurement using a regret minimization approach. We started with mechanism BP-DGREEDY for the bidding model and then extended it to our main mechanism BP-UCB for the posted-price model. These are the first provable no-regret mechanisms for online budgeted procurement.

There are some natural extensions for future work. Here, we considered a simple additive utility function for the requester. It would be useful to extend our approach to more complex utility functions. Additionally, we assumed a homogeneous pool of users, although it would be more practical to design mechanisms which can take into account skills and different utility values of the users. We used the knowledge of known bounded support and furthermore discretized the price space. Results from continuous arm bandits in [Kle04; AOS07; BZ09b; Sri+10] can be applied here by making more realistic assumptions about cost distributions. This would enable learning the support as part of the mechanism itself and remove the regret from discretization. In our work, we assumed a finite yet large pool of available users. A perhaps more natural approach is to use time-discounted rewards where a mechanism’s goal would be a timely completion of the task.
In the previous chapter, we developed the posted-price mechanism BP-UCB and provided the no-regret guarantees of our mechanism. In this chapter, we apply our mechanism to an important application of procurement of workers in microwork based crowdsourcing marketplaces like Mechanical Turk [Mtu] and CrowdFlower [Cro]. Our proposed model of posted-price offers naturally fit in microwork based marketplaces where the amount of monetary payments is usually small, and it is impractical to elicit it from workers via bids. Furthermore, the task requester in such platforms usually has a limited budget and time constraints on the task completion (which in turn leads to a limited pool size of workers the mechanism interacts with).

We carry out extensive experiments to compare the performance of BP-UCB with optimal benchmarks, as well as the state-of-the-art mechanism of Badanidiyuru, Kleinberg, and Singer [BKS12]. To the best of our knowledge, this is the first empirical study of posted-price mechanisms in procurement auctions using real-world data. Furthermore, we compare the performance of our posted-price mechanism against the mechanisms for bidding model, allowing us to quantify the loss we incur by limiting the information we elicit from the workers. Apart from experimenting with simulated workers’ cost distributions, we perform experiments using data gathered from a study
on the *Mechanical Turk* platform to demonstrate the effectiveness of our approach on real world inputs.

**Organization of this Chapter**

This chapter is organized as follows. In Section 5.1, we provide details of the experimental set, describing benchmarks, performance metrics, etc. Section 5.2 presents the experimental results using simulated cost distributions. In Section 5.3, we provide details of the survey study performed on the *Mechanical Turk* platform and Section 5.4 presents experimental results using the data collected in this survey study. Section 5.5 discusses the related work on understanding the role of incentives in crowdsourcing platforms and Section 5.6 presents our conclusions.

**Relevant Publication**

Results presented in this chapter are published in Singla and Krause [SK13b].

## 5.1 Experimental Setup

In this section, we describe the experimental setup, including our benchmarks, metrics, and the parameter choices made in the experiments.

### 5.1.1 Benchmarks and Baselines

We compare our posted-price mechanism BP-UCB against various benchmarks and state-of-the-art mechanisms, listed below. Furthermore, we also measure the performance of our mechanism BP-DGREEDY in the bidding model. BP-DGREEDY serves as a natural upper bound for our mechanism BP-UCB. This comparison of BP-UCB and BP-DGREEDY also allows us to quantify the loss our posted-price mechanism incurs because of limited information (*i.e.*, binary feedback) instead of eliciting private costs from the workers. In the plots, we denote BP-DGREEDY as bp-gd and BP-UCB as bp-ucb.
5.1. Experimental Setup

**Offline/Untruthful Mechanisms**

OPT-Var and OPT-Fix are offline, untruthful mechanisms with full information of workers’ true costs as introduced in the model described in the previous chapter (cf., Section 4.2 in Chapter 4). OPT-Var would sort the workers by their increasing costs and offer the price same as that of worker’s costs until budget exhausts. OPT-Fix is similar to OPT-Var except that the mechanism commits to a fixed optimal price (denoted as $p^*$ in Equation (4.1), cf., Chapter 4) to be offered to all workers. In the plots, we denote OPT-Var as opt-var and OPT-Fix as opt-fix.

**Mean Price Offers**

Mean is a simple offline mechanism that operates under the bidding model. It offers a fixed price computed as the mean value of the workers’ bids. This mechanism serves as a rule of thumb to determine fixed prices for tasks, as possibly used by inexperienced requesters. In the plots, we denote Mean as mean.

**State-of-the-Art Mechanism for Bidding Model**

We consider the state-of-the-art mechanism, BS’11, for the bidding model based on sampling bids from Singer and Mittal [SM11]. This mechanism assumes that workers’ arrival order is stochastic i.i.d. In the plots, we denote BS’11 as bs’11.

**State-of-the-Art Mechanism for Posted-Price Model**

We consider the state-of-the-art online posted-price mechanism, PP’12, from Badanidiyuru, Kleinberg, and Singer [BKS12] designed for the stochastic setting. We found that the recommended parameters used for proving theoretical guarantees did not work in practice. We therefore manually tune parameters to optimize the performance of this benchmark. Specifically, we ignore the parameter $z$ (cf., [BKS12]) in determining the price of the highest arm as $\frac{B}{z}$ and instead use $c_{\text{max}}$ as a bound. Also, we use $a = 5$ instead of 4000 (cf., [BKS12]), which would need an extremely large pool of workers for execution. In the plots, we denote PP’12 as pp’12.
5.1.2 Metrics and Types of Experiments

The primary metric we track is the utility of the mechanism as we vary the budget $B$, setting $N = \frac{B}{c_{\text{min}}}$. We also compute the average regret of the mechanism w.r.t. increasing budget to verify its no-regret property. To study the effect of the worker pool size, we also look into varying $N$ for a fixed budget. To gain insight into execution of the mechanisms, we measure their rate of convergence by determining the unique price to which the mechanism converges in the end and measuring the number of times this price has been offered so far with increasing timesteps. Lastly, we evaluate the utility over time to understand the dynamics of how quickly the budget is exhausted.

5.1.3 Parameter Choices

We use $c_{\text{min}} = 0.01$ and $c_{\text{max}} = 1$ based on the payment bounds typically seen on the Mechanical Turk platform. The price discretization factor $\alpha$ is set to 0.2. We note that setting $\alpha$ to $O\left(\frac{\ln(B/c_{\text{min}})}{B}\right)$ guarantees asymptotic bounds of $O\left(\ln(B/c_{\text{min}})\right)$, however smaller values of $\alpha$ would increase the number of arms, leading to slower convergence.

5.1.4 Cost Distributions

We consider cost distributions based on simulations as well as gathered from an actual survey study. We consider various simulated distributions for analyzing our algorithms, including uniform, normal, exponential and more complex ones including a mixture of two uniform or two Gaussian distributions. Also, we consider various settings to simulate the arrival of workers, including ordering by ascending bids to simulate adversarial arrival. To simulate a more realistic non-stochastic setting, we consider groups of two distributions arriving one after another, ordered by their increasing means.

5.2 Experimental Results on Simulated Distributions

We now present and discuss the findings from our experiments on simulated distributions.
5.2. Experimental Results on Simulated Distributions

Figure 5.1: Acquired utility for simulated distributions when varying budget. (a) and (b) show results for *i.i.d.* settings for bidding and posted-price model, respectively. In (b), BP-UCB outperforms PP’12 by over 150% increase in utility for the stochastic settings. (c) considers workers arriving in order of ascending bids. (d) considers two groups of workers with bids uniformly distributed in \([0.1, 0.5]\) and \([0.5, 0.9]\) arriving one after another.

### 5.2.1 Acquired Utility

Figure 5.1 shows results for costs uniformly distributed in the range \([0.1, 0.9]\), though the results are qualitatively similar for other distributions and ranges. We consider the online setting with stochastic arrival of workers and also assess the robustness of the mechanisms when these assumptions are violated. In Figure 5.1(a), we can see that the mechanism BP-DGREEDY performs very close to OPT-FIX and slightly outperforms the
Chapter 5. Applications to Procurement in Crowdsourcing Marketplaces

state-of-the-art mechanism BS’11 for the bidding model. Somewhat surprisingly, as we can see in Figure 5.1(b), our mechanism BP-UCB for the posted-price model performs as good as BP-DGreedy even though it operates under limited feedback. It clearly outperforms PP’12 by an over 150% increase in utility for all the budgets considered. Mean is much lower compared to both of our mechanisms, suggesting that rules of thumb prices may not be optimal.

We also simulate arrival of the workers in order of ascending bids, violating the stochastic i.i.d. assumptions. In Figure 5.1(c), we see that all the mechanisms perform quite poorly in this somewhat unrealistic case. Figure 5.1(d) shows results for a perhaps more meaningful non-stochastic setting where two groups with bids uniformly distributed in $[0.0, 0.5)$ and $[0.5, 0.9]$ respectively arrive after another. Therefore, a natural question is how robust the algorithms are w.r.t. more realistic cost distributions, as we will analyze in the next two sections using data from a survey study on the Mechanical Turk platform.

5.2.2 Getting Deeper Insights

We perform a few more experiments to get deeper insights into the execution of the mechanism, including the convergence behavior and offers made over time.

Effect of Varying $N$

Apart from varying the budget, it is interesting to compare the impact of workers’ pool size on the mechanisms for a fixed budget. Note that the availability of more workers (larger $N$) shifts the optimal solution towards lower prices. Figure 5.2(a) shows the impact of varying $N$. As one would expect, our mechanisms BP-DGreedy and BP-UCB as well as BS’11 show an increase in utility exhibiting diminishing returns. Interestingly, PP’12 shows a decrease in utility as the number of workers increases.

Average Regret and Convergence

Figure 5.2(b) shows the average regret of the mechanisms with increasing budget. Note that the average regret of BP-UCB decreases at a much faster rate compared to that of PP’12. Next, we look at the rate of convergence of the mechanisms in Figure
5.2. Experimental Results on Simulated Distributions

Figure 5.2: Uniform distribution in [0.1, 0.9], stochastic settings. In (b), no-regret properties of BP-UCB can be seen as the average regret diminishes with increase in budget. (c) shows better convergence rate of BP-UCB compared to PP’12. (d) shows that BP-UCB makes low offers in beginning, in contrast to PP’12 which quickly exhausts the budget.

5.2(c), by computing the proportion of times the unique price, to which the mechanism converges in the end, has been offered so far with increasing timesteps. BS’11 rapidly converges to the unique price, favorably compared to BP-DGREEDY. We can see an initial phase of exploration for BP-UCB followed by exploitation as the mechanism converges. However, PP’12 stabilizes at 50% convergence as the Markov model used by the mechanism flips back and forth between the equilibrium prices.
Utility with Timesteps

Lastly, we study how the mechanisms accrue utility over time (Figure 5.2(d)). BP-UCB offers very low prices in the initial phase of exploration, followed by convergence to a unique price, after which the utility increases almost linearly. In contrast, PP’12 quickly exhausts the budget in the beginning by offering high prices, leading to overall reduced utility.

5.3 Survey Study on the Mechanical Turk Platform

We now describe details of the survey study performed on the Mechanical Turk platform. The goal of this survey study is twofold. Firstly, we want to get the workers’ cost distribution for a realistic scenario which fits our procurement auction task. Secondly, we want to understand whether the assumption of stochastic costs holds true in real world inputs.

5.3.1 Setup of the Survey Study

We posted a Human Intelligent Task (HIT) on the Mechanical Turk platform in the form of a survey, where workers were told about an option to participate in a hypothetical advertisement system for a social networking site. In this hypothetical system, they can opt to use the top of their homepage for banner ads and obtain some monthly payment from the publishers. Workers were asked to bid on the monthly payment they would like to receive, in addition to providing information like years of being active on social networks and time spent there, approximate geographical location, the number of friends, and optional comments. They were asked to provide this information for different times including July 2012, July 2011, and July 2010.

A total of 1200 workers participated in our HIT, which was online for one week, restricted to workers with more than 90% approval rate. Workers were paid a fixed amount for participation in the HIT. We did not restrict the workers to any geographical region. Additionally, we made a bonus payment to selected 20 individuals based on their insightful comments about factors affecting their payment choice.
5.3. Survey Study on the Mechanical Turk Platform

Figure 5.3: Survey study on the Mechanical Turk platform about an option to participate in a hypothetical advertisement system for a social networking site. (a) Distribution of workers’ bids, (b) correlation with usage time, and (c) correlation with the number of friends online.

5.3.2 Statistics of the Data

The workers represented more than 20 different countries with 44.5% from USA and 44.0% from India. In total 72.25% of the workers agreed to participate in the hypothetical online advertisement system, and we analyze the statistics from these workers below. Workers from USA have a lower acceptance rate of participation (59.3%) compared to workers from India (86.1%), which shows interesting dependence on geographical factors in determining the pricing model of the workers. Table 5.1 shows the mean and median values of various features. The data shows an increase in social
activity (in terms of friend count and service usage) as well as the bids reported by workers for their service. Figure 5.3 shows the distribution of bids collected as well as the correlation with usage time and the number of friends. The data is skewed towards lower bids and is discretized because of the tendency of workers to bid at rounded numbers. In total 75.8% of the workers provided subjective feedback in the comments section about the pricing factors. Common themes reported by the workers for the pricing factors were the usage time, friend count, and nature of the ads. The statistics related to friend size matches closely with those of publicly available numbers supporting the quality of the data obtained from the workers.

### 5.4 Experimental Results on Real Workers’ Distribution

Figure 5.4 shows results for cost distributions corresponding to the above-mentioned survey study on the Mechanical Turk platform (denoted as MTurk in the plots). We considered bids ranging in $[10, 100]$, scaled down by 100, although the results are qualitatively similar for other ranges. Note that scaling down the costs is equivalent to scaling up the budget. We sampled with replacement from the bids to generate the entire pool of workers. We considered various online settings to simulate the arrival order of workers: the actual order in which workers arrived on the platform for completing the HIT (i.e. the survey); ordered by their usage time; ordered by the number of friends, and by the year of joining the social network. Here, we discuss the results for the arguably most natural orderings.
Figure 5.4: Utility for the workers’ distribution for bids in [10, 100] collected from the Mechanical Turk platform (MTurk), varying budget. In (b), BP-UCB outperforms PP’12 by over 180% increase in utility. Also, BP-UCB and BP-DGREEDY are robust against all the online settings above.

5.4.1 Acquired Utility

For the bidding model, we can see in Figure 5.4(a) that both OPT-Fix and BS’11 coincide exactly with OPT-Var, in the case of workers arriving according to the actual ordering in which they arrived on the platform for completing the HIT. We attribute this to the highly skewed nature of bids at low prices, as the optimal strategy for all these three mechanisms is to offer a single fixed price corresponding to the lowest bid. For the posted-price model in Figure 5.4(b), BP-UCB clearly outperforms PP’12, increasing the utility by over 180% for all the budgets considered.
5.4.2 Robustness in Different Online Settings

Figure 5.4(c) shows results where workers are ordered by the year in which they joined and Figure 5.4(d) shows the results where workers are ordered by their increasing bids. Interestingly, BP-UCB and BP-DGreedy continue to perform well in both the settings, whereas BS’11 degrades in Figure 5.4(d) and PP’12 performs poorly in both.

5.5 Related Work

In this section, we review some relevant literature on understanding the role of incentives in crowdsourcing platforms.

5.5.1 Incentives in Crowdsourcing Platforms

There has been a growing interest in understanding the right incentives for workers in online labor markets and crowdsourcing platforms like Mechanical Turk and CrowdFlower. Ipeirotis [Ipe10] provides a detailed study of the Mechanical Turk platform, a popular microwork based crowdsourcing platform. They shed light on some basic questions such as who are the workers on the platform, how should tasks be priced, and what is the completion time of the tasks. Faradani, Hartmann, and Ipeirotis [FHI11] study the interaction between pricing policy and the task completion time on crowdsourcing platforms. They devise an algorithm for pricing the tasks on these platforms based on models of workers arrival. Horton and Chilton [HC10] present a model of workers and introduce methods to estimate workers’ appropriate wages. The hagglebot of Horton and Zeckhauser [HZ10] negotiates payment rates for an image-labeling task with workers on the Mechanical Turk platform.

Shaw, Horton, and Chen [SHC11] conduct a series of behavioral experiments in an online labor market to measure the effectiveness of different social and financial incentive schemes. Their results on the Mechanical Turk platform show that financial incentives combined with some sort of peer prediction (whereby the workers are asked to also predict the response of other workers) perform best in their setting. A recent work by Ho et al. [Ho+15] studies the impact of performance based bonus on the quality of the work in crowdsourcing platforms through a series of behavioral experiments on the Mechanical Turk platform. Mao et al. [Mao+13] perform behavioral experiments to
compare the relative performance of volunteering vs. paid crowdsourcing, in order to understand the impact of financial incentives on the work quality. Mason and Watts [MW10] study the relationship between the workers’ performance and the offered financial incentives via online experiments performed on the Mechanical Turk platform.

5.6 Summary

In this chapter, we applied the pricing mechanism BP-UCB to the application of procuring workers in microwork based crowdsourcing platforms using monetary incentives. Apart from theoretical guarantees proven in the previous chapter, we show that our approach is empirically efficient compared to optimal benchmarks, and dramatically outperform the state-of-the-art posted-price mechanism. Our experiments on the Mechanical Turk platform further supports the practical applicability of our mechanism on crowdsourcing platforms. We believe that our results provide an important step towards developing practical, yet theoretically well-founded techniques for increasing the efficiency of crowdsourcing.

Our experiments on the Mechanical Turk platform suggest that real world inputs may violate stochastic assumptions. While our mechanisms are robust against our study’s cost distribution, one can force all the currently available mechanisms to perform poorly by carefully designing (unrealistic) cost distributions. It would be of interest to develop mechanisms that are more robust and extend to the oblivious adversary model. Existing crowdsourcing platforms support only fixed price mechanisms and limited capabilities to design pricing policies. Our experiments show that simple mechanisms like Mean perform quite poorly, although inexperienced requesters may be tempted to use them as rule of thumb. It would be interesting to build applications and conduct studies where we can actually run our mechanisms in real-time on crowdsourcing platforms.
In this chapter, we consider bike sharing systems based on shared mobility, and tackle the imbalance problem, i.e., the unavailability of bikes or parking docks at stations. We present a novel crowdsourcing scheme that incentivizes the users in the bike repositioning process by providing them with alternate choices to pick or return bikes in exchange for monetary incentives. Our pricing mechanism BP-UCB is the key learning component in the crowdsourcing scheme that we develop. We begin by first describing the bike sharing systems.

A Bike Sharing System (henceforth BSS) is a new concept of public transportation offering citizens a flexible and green alternative as well as complementing the slow and crowded transportation in urban areas. This mobility trend had an exponential growth over the last years, and, as of June 2014, over 700 cities actively operate automated bike sharing systems deploying an estimate of 800,000 bikes worldwide [MD14]. BSSs increase the user’s commuting flexibility by allowing her to pick up or drop off a bike at any station and let her decide the duration of the trip without any prior planning or reservation. Indeed, this flexibility is the key factor for the success of BSSs.
Imbalance Problem

This flexibility also poses a number of new challenges for the BSS operators. The demand is often unpredictable, asymmetric, and fluctuating throughout the day. Other factors such as altitude differences, weather conditions, or events in the city can cause irregular or asymmetric rental demands and flow of bikes. As the number of available bikes and parking spots at any bike station in the system is limited, satisfying the forthcoming demand with such limited resources is a major challenge and recurrent problem for BSS operators. A BSS user would experience an unsatisfactory quality of service when attempting to rent or drop off a bike at an empty or full station respectively. Thus, fleets of vehicles (or trucks) are deployed to redistribute bikes among stations to balance the system out [BMB11]. Meeting the demand for bikes and free docks requires an extensive logistic work and poses major operational costs to the operators.

Incentivizing Users

Apart from the large operational costs, deploying trucks for the redistribution goes against the green concept of BSS. The key research question that we seek to answer in this chapter is:

*Can we engage the users of such systems and provide them incentives to contribute towards rebalancing the system?*

Termed as crowdphysics by Sadilek, Krumm, and Horvitz [SKH13], this class of crowdsourcing systems requires users to sequence or synchronize physical actions in time and space, and can lead to building self-sustainable systems. The goal of building such a system presents several challenges including: (i) the system’s large scale in terms of the user base as well as the number of stations, (ii) the fluctuating demand over time that needs to be taken into account while determining problematic stations, (iii) the unknown personal costs of the users and their strategic behavior potentially aiming at maximizing their profit, (iv) the budget constraints of how much can be spent on incentives, and (v) a user-friendly interface to make such a system appealing (e.g., through a smartphone app). Building such a self-sustainable and crowd-powered bike sharing system is the key idea we explore in this chapter. While we focus on BSSs,
similar problems arise in other domains (e.g., car sharing or rerouting users on overbooked flights) and our methodology is applicable to these systems as well.

**Organization of this Chapter**

This chapter is organized as follows. In Section 6.1, we give a high-level overview of our approach. Section 6.2 provides a formal model of the bike sharing systems and formalizes the problem addressed in this chapter. In Section 6.3, we describe our incentive scheme and the pricing mechanism based on BP-UCB. Section 6.4 describes the experimental setup and the results based on simulations are presented in Section 6.5. We report the results from deployment in *Mainz, Germany* in Section 6.6. Section 6.7 discusses the related work and Section 6.8 presents our conclusions.

**Relevant Publication**

Results presented in this chapter are published in Singla et al. [Sin+15a].

### 6.1 Overview of Our Approach

In this chapter, we address this key research question mentioned above and present a crowdsourcing scheme that incentivizes users in the bike repositioning process. Our scheme provides users with alternate choices to pick or return bikes in exchange for monetary incentives. We design the complete architecture of the incentives system which employs pricing policies based on BP-UCB, our posted-price mechanism developed in Chapter 4 of this dissertation. We extensively evaluate our approach on a historic dataset from *Boston’s Hubway* bike sharing system, made publicly available by the *Boston Metropolitan Area Planning Council* [Hub].

Finally, in collaboration with a bike sharing company *ElectricFeel* [Ele], we deployed the proposed system through a smartphone app among users of a large-scale bike sharing system operated by a public transport company *MVGmeinRad* [Mvg] in the city of *Mainz, Germany*. In Section 6.6 we provide results from this experimental deployment and report on our experience deploying our system. To our knowledge, this is the first dynamic incentives system for bikes redistribution ever deployed in a real-world bike sharing system.
6.1.1 Architecture of Our Incentives System

In this chapter, we design the complete architecture of an operating incentives system for engaging users in the bike-repositioning effort, see Figure 6.1. In Figure 6.1, the components \textit{C1 Rental Stations} and \textit{C2 BSS Infrastructure} correspond to the rental stations, the hardware and software infrastructure of the public bike sharing system. These components involve tracking all the user activities such as picking or dropping the bikes, managing user accounts, and the payment system.

The main components that we designed and built include \textit{C3 User Model}, \textit{C4 Incentives Deployment Schema (IDS)}, and \textit{C5 Smartphone App}. The IDS is the central component in the overall design that handles the user’s requests through \textit{C5 Smartphone App}. The request includes input such as the user’s target station and type \textit{(i.e.,} bike pick up or drop off\textit{)}. The IDS communicates with the BSS Infrastructure to evaluate the current and predicted status of the stations, and then decides whether to offer incentives to the user by requesting a change in her pickup or drop-off location. We consider the users as strategic agents who may untruthfully report information about their personal cost and location to maximize their profit. We use a dynamic variant of our posted-price mechanism BP-UCB to offer incentives to the users—an important part of the IDS in order to maximize the efficiency under given budget constraints. We model the full rental process of the user and her reaction to incentives in the component \textit{C3 User Model}.

Figure 6.1: Overview of the system, our contributions are shown in the dotted box.
6.2 The Model

We now formalize the model of bike sharing systems and the imbalance problem addressed in this chapter.

6.2.1 Bike Sharing Systems

There are $m$ stations in a city denoted by the set $S = \{s_1, s_2, \ldots, s_m\}$. We denote the number of bikes available at time $t$ at station $s$ as $v(s, t)$. Along the lines of Pfrommer et al. [Pfr+14], we split every day into 12 slices $h \in \mathcal{H}$ of two hours each, where a time of the day $t$ is mapped to the corresponding time-slice using $h(t)$. The demand of rentals from station $s_i$ to station $s_j$ within time-slice $h$ is modeled as a discrete random variable $z$ which represents a count of the number of trips with probability density function $\tau(z|s_i, s_j, h)$. We denote $\hat{z}(h)$ to represent the estimate of the total number of trips within time-slice $h$. We consider that the BSS infrastructure has a proprietary demand forecaster that can predict the future demand of incoming and outgoing traffic at the stations and is made available to us through their APIs, cf., Figure 6.1. See the work by Côme and Oukhellou [CO12], Kaltenbrunner et al. [Kal+10], Borgnat et al. [Bor+11], Froehlich, Neumann, and Oliver [FNO09], and Han, Côme, and Oukhellou [HCO14] for possible approaches.

6.2.2 Quality of Service

The goal of a bike repositioning policy is to prevent customers’ dissatisfaction about not finding an available bike or a parking slot. Waserhole, Jost, and Brauner [WJB13], Chemla, Meunier, and Pradeau [CMP13], and Fricker and Gast [FG12] have proposed metrics aiming at maximizing the number of trips or minimizing the number of problematic stations. However, these metrics fail to capture the dissatisfaction experienced by BSS users. We adopt the measure of service level proposed in Pfrommer et al. [Pfr+14]:

$$\text{Service level} = \frac{\text{(potential customers)} - \text{(no-service events)}}{\text{(potential customers)}}$$

where the number of no-service events corresponds to the customers who could not rent a bike at an empty station or failed to return their bike at a full station.
6.2.3 Truck-Based Repositioning Policies

The BSS operator allocates a fixed daily budget $B$ that can be used by the truck-based policy. The BSS company where we deployed our incentives system uses its proprietary repositioning policy. See the work by Chemla, Meunier, and Pradeau [CMP13] and Raviv and Kolka [RK13] for possible approaches.

6.2.4 User Model

Consider a user (or customer) $u$ at location $l_x^u$ interested in an action $x \in \{\text{pick, return}\}$, i.e., pick up or drop off a bike. We assume that, by default, the user would consider the nearest station to her location to perform the action, given by $s_x^u \leftarrow \arg\min_{s \in S} d(l_x^u, s)$, where $d$ measures the distance between two locations. The incentives system encourages users to shift their pickup or drop-off stations to contribute balancing the BSS. For simplicity, we assume that the user $u$ is willing to walk to another station up to a maximum distance $\gamma_u^u$, and, within this radius, her cost for the additional effort is constant given by $c_u^u$. She is not willing to walk for distance more than $\gamma_u^u$ despite of any offer by our incentive system. We validate these assumptions through a survey study with real BSS customers. In our model, the costs of the users are $i.i.d.$ sampled from an underlying unknown distribution $f$. We let $F : [c_{\min}, c_{\max}] \to [0,1]$ denote the cumulative distribution function (CDF) of costs or cost curve, unknown to the system.

Suppose a user $u$ at location $l_x^u$, going to station $s_x^u$, is offered an alternate station $s_x^*$ for a payment of $p_u^u$. We model her reaction to this incentive as an indicator function:

$$1_{\text{accept}}(l_x^u, s_x^*, p_u^u) = \begin{cases} 0 & \text{if } \gamma_u^u < d(l_x^u, s_x^*) \text{ or } p_u^u < c_u^u \\ 1 & \text{otherwise} \end{cases}$$

which takes value 0 or 1 when the user rejects or accepts the offer, respectively. We remark that the information about user cost $c_u^u$ and her true location $l_x^u$ is private to the user and may vary among users. We consider the users as strategic agents who may untruthfully report their private information about the location $l_x^u$ or private cost $c_u^u$ to maximize their profit. For simplicity of the mechanism design, we use $\hat{\gamma}$ to represent the maximum distance for the whole population, and we obtain its estimate through a user survey.
6.2.5 Incentivizing Users for Repositioning

We study an incentives system as an alternative to truck-based redistribution in BSSs. We propose a dynamic incentives system that computes incentives each time a user makes a new request. The BSS operator regularly allocates a budget $B(h)$ for time batch $h$, and the goal of the incentives system is to exploit the budget efficiently. Therefore, if a user $u$ makes a request at time $t$ for action type $x$, a dynamic algorithm is responsible to select the alternate station $s^*_x$ and compute the value of the incentive $p^u$ based on the following inputs: (i) reported location of user $l^u_x$, (ii) current status of the system: $v(s,t)$ $\forall s \in S$, (iii) near-future demand $\tau(z|s_i,s_j,h)$ $\forall s_i,s_j \in S$, (iv) budget available, and (v) information about historical interactions with users.

Our goal is to design a system that is (i) budget feasible, i.e., operates under strict budget constraints, (ii) efficient in terms of improvement in quality of service for a given budget, (iii) incentive-compatible (truthful), i.e., it is always in best interest of users to reveal true information, and (iv) can deal with unknown user costs and learn pricing policies over time.

6.3 The Incentives System

We now present the various components of our incentives system and discuss its properties.

6.3.1 Incentives Deployment Schema

Procedure presented in the Algorithm 6.1 illustrates the Incentives Deployment Schema (IDS) that handles the users’ requests forwarded by the deployed smartphone app (cf., Figure 6.1). For each request, the IDS gets as input the user identifier $u$, the action type $x \in \{\text{pick, return}\}$, and the user’s reported location $l^u_x$. Based on the user input and the current status of the system, IDS computes the offer that consists of the alternative target station $s^*_x$ and the offered price $p^u$. Initially, the IDS ensures that the user does not have any pending incentive, i.e., an incentive offer that she recently accepted but did not accomplish yet. Then it creates a set of candidate stations that are within the maximum walking distance $\hat{\gamma}$ from $l^u_x$ and are problematic, i.e., full or empty. The problematic status of the stations is based on input from the APIs of the
Algorithm 6.1: Incentives Deployment Schema (IDS)

1. **Input:** user $u$; location $l^u_x$; $x \in \{\text{pick, return}\}$
2. **Output:** Offer: station $s^*_x$; price $p^u$;
3. **Parameters:** radius $\hat{\gamma}$;

```
begin
  if \text{HASINCENTIVESPENDING}(u) then
    return null;
  end
  candidate stations $C \leftarrow S$;
  default target $s^u_x \leftarrow \arg \min_{s \in S} d(l^u_x, s)$;
  foreach $s \in C$ do
    $s.status \leftarrow \text{PROBLEMATIC}(s)$;
    if $d(l^u_x, s) > \hat{\gamma}$ then
      $C = C \setminus \{s\}$;
    end
    if $x = \text{return}$ & $s.status \neq \text{EMPTY}$ then
      $C = C \setminus \{s\}$;
    end
    if $x = \text{pick}$ & $s.status \neq \text{FULL}$ then
      $C = C \setminus \{s\}$;
    end
  end
  if $s^u_x \in C \text{ OR } C = \emptyset$ then
    return NULL;
  end
  $s^*_x \leftarrow \arg \min_{s \in C} d(l^u_x, s)$;
  $p^u \leftarrow \text{PRICINGMECHANISM}()$;
  \text{▷ DBP-UCB, a dynamic variant of BP-UCB from Chapter 4;}
  Output: $s^*_x, p^u$;
end
```

BSS infrastructure that provides the current status $v(s, t)$ as well as the forecast of future traffic. The IDS also takes into account the pending incentives associated with
6.3. The Incentives System

Each station while updating the status that additionally accounts for future traffic.

Once the IDS has filtered the candidate stations, it selects as target station $x^*_u$ the one closest to reported location $l^u_x$. Then, it sends a request to the pricing mechanism to obtain the price $p^u$ to be offered. This offer is displayed to the user on the smartphone app. If the user accepts, the IDS registers this incentive offer as pending until she accomplishes the assigned task or it expires after a prefixed amount of time.

6.3.2 Pricing Mechanism: Dynamic BP-UCB

Next, we present our pricing mechanism DBP-UCB invoked by the IDS. It is a dynamic variant of the posted-price mechanism BP-UCB developed in Chapter 4 of this dissertation.

The main challenges in designing an optimal pricing mechanism for our problem setting in this chapter involve limited and dynamically changing budget $B$ as well as pool size of users to make offers to $N$ and the unknown cost curve $F$. We note that, in the model we studied in Chapter 4 while designing BP-UCB, the budget $B$ and the pool size $N$ are fixed in advance.

For static settings where the budget $B$ and the number of users $N$ is known and fixed in advance, the mechanism BP-UCB can directly be used by the IDS. This scenario does not correspond to the incentives system we are interested in as, in this context, the pricing mechanism has to operate continuously throughout time, is allocated new budget in time batches, and the number of users coming to the system is dynamic. In this light, we adapt BP-UCB to design our mechanism DBP-UCB (Dynamic BP-UCB), illustrated in Algorithm 6.2. The key idea we use is that the learning process of the mechanism BP-UCB can be decoupled from the constraints on $B$ and $N$. While the parameters $B$ and $N$ change dynamically at the onset of each time batch, it carries over the estimates of $F(p)$ across batches, ensuring DBP-UCB maintains the convergence properties of BP-UCB.

At a high level, the mechanism DBP-UCB operates as follows. At the onset of each new time batch $h$, the mechanism is provided with an additional budget $B(h)$ by the BSS operator. Furthermore, the number of participants $N$ for a batch is approximated by the expected number of trips $\hat{z}(h)$ taking place in the corresponding batch $h$ estimated by the BSS forecaster. In particular, the expected number of rentals is multiplied by
two as the user may interact with the incentives system both to pick and return a bike. The mechanism sequentially interacts with users in discrete steps denoted by \( n \). Let \( B^n \) be the budget at iteration \( n \), \( N^n_i \) be the number of times \( p_i \) price has been offered so far, and \( F^n_i \) be the current estimate of the cost curve for prices \( \forall i \in [0, \ldots K] \). The mechanism also maintains upper confidence bounds on \( F^n_i \) denoted as \( \tilde{F}^n_i \), representing the optimistic estimates of \( F \) at iteration \( n \), and used to compute the optimal price \( p^n \) to offer. The mechanism receives the binary feedback \( y^n \) which is then used to update the estimates for price \( p^n \).

### 6.3.3 Truthfulness of the Incentives System

As the system users may act strategically, it is important to analyze the truthfulness of the IDS to understand the real-world performance of the proposed system. Each user has a private cost \( c^u \) and location \( l^u_x \) and we now discuss the truthfulness of our system w.r.t. these two dimensions of the user’s private information.

The truthfulness of DBP-UCB follows from the fact that it is based on the posted-price model: In such models, it is in the user’s best interest to truthfully accept the offer whenever the offered price is higher than or equals the private cost \( c^u \), thus making it dominant strategy incentive-compatible (truthful), cf., Chapter 4 for more discussion.

Let us now analyze the truthfulness of IDS w.r.t the location information. One crucial aspect here is that no incentives are offered when the user’s default station \( s^u_x \) is already among the candidate stations. This notably improves the efficiency (by over 50% in our simulations) of the IDS by avoiding to pay out incentives for events that the user would have taken anyways. We discuss the implications of this on the truthfulness of IDS below. If the system has no way to infer \( s^u_x \), the user may indeed act strategically by misreporting to obtain an incentive that would not have been offered otherwise. In the real deployment of our system, instead of allowing the user to declare her current location \( l^u_x \), it is directly retrieved from the deployed smartphone app which provides localization features. Based on this location information, the user’s default pickup station can directly be inferred. This technical solution does not directly apply to the bike drop-off scenario where users explicitly declare their target location. However, the drop-off location could potentially be inferred based on user’s historical trips, making the system incentive-compatible in real-world deployments.
Algorithm 6.2: Pricing Mechanism DBP-UCB

1. **Input**: start time $t_0$; prices $\{p_0, \ldots, p_i, \ldots, p_K\}$;
2. **Output**: Price $p^n$ at iteration $n$;
3. **Initialize**:
   - $n = 0; h^0 = h(t_0)$;  
   - $B = B(h^n); B^n = B; N = 2 \cdot \hat{z}(h^n)$;
   - $N^i_n = 0; F^i_n = 0; \forall i \in [0, \ldots, K]$  
4. **foreach** request by user at time $t$ do
   - if $h^n \neq h(t)$ then
     - $h^n = h(t)$;  
     - $B^n = B^n + B(h^n); B = B^n$;
     - $N = 2 \cdot \hat{z}(h^n)$;
8. $F^n_i = F^n_i + \sqrt{\frac{2 \ln(n)}{N^n_i}}$;
9. $i^n = \arg\max_i \min \left\{ F^n_i, \frac{B}{N^i_n} \right\}$ s.t. $p_i \leq B^n$;
10. **Output**: Offer price $p^n = p_{i^n}$ at iteration $n$;
11. **Feedback**: Observe acceptance decision $y^n$;
12. **Update Variables**:
   - $B^{n+1} = B^n - p^n \cdot y^n; F^{i^n+1} = F^{i^n} + \left( \frac{y^n - F^n_{i^n}}{N^{i^n+1}_n} \right)$;
   - $N^{i^n+1}_n = N^{i^n}_n + 1; h^{n+1} = h^n; n = n + 1$;
13. Wait new user request;
end

6.4 Experimental Setup

In this section, we carry out extensive experiments to understand the practical performance of our system. Next, we describe our datasets and the BSS simulator.
What is the maximum additional distance you are willing to walk?
Choose one of the options:
- Unwilling to walk
- More than 2,000 m
- 1,500 m
- 1,000 m
- 750 m
- 500 m
- 250 m
- More than 2,000 m

Figure 6.2: Survey study with customers of a real-world BSS in the city of Mainz, Germany where we deployed our incentives system.

6.4.1 Historical BSS Dataset from Boston’s Hubway

For our simulations, we use a historical dataset from Boston’s Hubway, made publicly available by the Boston Metropolitan Area Planning Council [Hub]. The Hubway dataset contains data collected between 28th July, 2011 and 1st October, 2012 with rich information about 95 stations, 694 bicycles, 552,030 rentals, and snapshots of the status of the BSS at regular intervals.

6.4.2 Survey Study Among BSS Customers in Mainz, Germany

We did a survey study among the customers of a real-world BSS in the city of Mainz, Germany where we also deployed our system. Our goal is to validate the realism of our model, as well as to obtain realistic statistics about the distribution of users’
personal costs $c^u$ and of their maximum walking range $\gamma^u$. The participants were asked generic questions about their rental behaviors (e.g., purpose and duration of the trips) and then we introduced them to the proposed incentives system asking various questions to elicit their preferences about private cost and walking distance. Figure 6.2(a) illustrates a survey question, and the distributions obtained from the survey for $\gamma^u$ and $c^u$ are shown in Figures 6.2(b) and 6.2(c) respectively. Note that the survey also contained an additional choice of unwilling to walk or unwilling to participate at any cost that accounted for roughly 20% of the response for both questions.

6.4.3 BSS Simulator

We built a complete simulator of a real BSS based on Hubway’s historical data as well as the survey data from customers. Each simulation starts by taking a snapshot of the status of the BSS (i.e., the number of bikes at each station retrieved from the historical dataset) and then runs for a total of 30 days. The BSS simulator generates users’ trip events by sampling from the distribution learned from the historical data. Lastly, the simulator associates each rental to a user with cost $c^u$ and $\gamma^u$ sampled from the distributions obtained from the survey. For the Trucks, we use the myopic greedy policy as defined by Chemla, Meunier, and Pradeau [CMP13]. Using the idea of dynamic (during rush hours) and static (during off hours) repositioning from Raviv and Kolka [RK13], our policies operate in time from 12:00 p.m. and 3:00 p.m., and then from mid-night to morning. We assume that the trucks entail a fixed cost of 50 € per hour to the system and the policy allocates the number of hours based on the total budget.

6.5 Experimental Results on BSS Simulator

We now discuss the findings from our experiments performed on the BSS simulator.

6.5.1 Varying Budget

We compare the performance of different policies by varying the allocated budget. We envision that the operators would deploy the incentives system in parallel to existing Trucks repositioning rather than as standalone policy. To take this into consideration,
we compare the policies when deployed in addition to an already running Trucks policy allocated with 150 €, (i.e., equivalent of 3 hours of trucks that would run between 12:00 p.m. and 3:00 p.m). Apart from comparing to Trucks, we also compare our proposed system with IDS running different baseline pricing mechanisms instead of DBP-UCB: (i) Minimum mechanism selects the minimum price available $c_{\text{min}}$; (ii) Mean mechanism offers the mean price obtained from the users’ cost distribution; and (iii) the benchmark OPT-Fix, i.e., the optimal fixed price mechanism computed with full knowledge of the underlying cost curve $F$ (cf., Section 5.1 in Chapter 5). Our results in Figure 6.3(a) indicate that the proposed IDS with DBP-UCB pricing mechanism compares favorably w.r.t. Trucks, as well as other baseline pricing mechanisms. The performance of DBP-UCB almost matches OPT-Fix, as it converges very fast to the optimal price in less than a day of simulation time. Note that allocating more budget $B$ to Trucks is simply equivalent to running policy with a total of $B + 150$ € budget.
6.5.2 Budget Tradeoff Between Incentives and TRUCKS

This experiment aims to understand how to trade off budget investment between TRUCKS and the incentives system. Figure 6.3(b) illustrates that the best service levels are achieved when the two repositioning services are run in complement to each other, and shows the high potential of adding our incentives system to the existing BSS infrastructure. In fact, these two policies have complementary effects. While the trucks tackle the imbalances at a macro level by moving the bicycles from a part of the city to the other, the effect of the incentives system is rather dynamic and at a micro level, incorporating the traffic flow and fluctuating demands.

6.5.3 Varying User Participation

The previous experiments assume that all the BSS customers participate in the incentives system. However, this assumption may not hold in reality, especially in the testing phase of the system. With the same setup as used in Figure 6.3(a), we vary the participation rate of users between 0% and 100%, with budget fixed at 300 €. Figure 6.3(c) illustrates that the proposed system can surpass standalone TRUCKS policy with 20% participation.

6.6 Deployment in Mainz, Germany

We integrated and deployed our incentives system on a real-world BSS in the city of Mainz, Germany. The deployment is done in a beta-test phase for a period of 30 days, in collaboration with a company ElectricFeel [Ele] (based in the city of Zurich, Switzerland) and a public transport company MVGmeinRad [Mvg] (operating the BSS in the city of Mainz, Germany).

We designed a smartphone app as an interface between the users and the incentives system (cf., Figure 6.1). The system itself was integrated with the BSS infrastructure through their APIs. The participants were offered monetary incentives for the bike pickup scenario, with payments directly transferred to the customer accounts. Given the small pool of participants in the testing phase limiting the process of learning, we adopted a simple pricing mechanism based on OPT-Fix. Using the approximate users’ cost distribution, denoted as \( \bar{F} \), obtained through the survey study, we computed the
6.6.1 Participation and Reaction to Incentives

Figure 6.4(a) represents the histogram of the number of incentives obtained by each participant. The result shows that most participants collected five or less incentives each while few particularly active users collected more than 50 incentives each.

In aggregation, the acceptance rate of the incentive offers over all participants was about 60%. Here, the rejection of an offer corresponds to the case of a user rejecting an incentive to pick up a bike at an offered target station and starting a rental somewhere else. Furthermore, as we recorded the information about user’s location when an offer
is made, we can compute the distance that the users were required to walk for a given offer. Figure 6.4(b) shows the probability of an offer being accepted as decreasing with respect to the distance to be walked and matches closely with the survey data in Figure 6.2(b).

6.6.2 Temporal and Spatial Distribution of Accepted Offers

Figure 6.4(c) shows that incentives were not collected equally throughout the day, but most incentives were collected during the afternoon and at late morning. In Figure 6.4(d), the diameter of each station is proportional to the number of incentives collected by users at that station. The map shows that the majority of incentives were paid out at stations in the city center.

6.7 Related Work

In this section, we provide an overview of the related work for truck-based and crowdsourcing-based repositioning policies in bike sharing systems.

6.7.1 Truck-Based Repositioning in BSS

The problem of asymmetric flow, imbalance, and repositioning of the vehicles has been studied in generic shared mobility systems that could involve cars or bikes [CF12; Kek+09; Cle+13; NMH11; ACA12]. Regarding BSS’s balancing problem, a substantial literature [RH+13; Pfr+14; RK13; CMW13; SHH13; GX11] introduced bike repositioning policies based on trucks. The user dissatisfaction function introduced by Raviv and Kolka [RK13] measures the performance of a BSS station and provides the quality of a repositioning. Nair et al. [Nai+13] provides a detailed quantitative analysis of the repositioning strategies for Vélib’, the BSS in Paris.

6.7.2 Crowd-Based Repositioning in BSS

Recent literature has investigated crowd-based BSS balancing policies [FG12; WJB13; Pfr+14; CMP13; GoD11] to study the potential of influencing BSS users by offering
Chapter 6. Applications to Balancing Bike Sharing Systems

Incentives that could improve the balancing of the system. In Paris, Vélib’ (cf., [V14]) runs a static incentive scheme that offers users 15 extra free minutes each time they return a bike to an elevated station. However, in contrast, our proposed system is dynamic and can efficiently take into account the current state of the system to decide where and what incentives to offer. Pfrommer et al. [Pfr+14] also introduced a dynamic pricing mechanism using model-based predictive control principles. Our pricing mechanism, in contrast, is based on an efficient and provably near-optimal online learning framework. Furthermore, we build an operating end-to-end system integrated with a real-world BSS for deployment, and we investigate the incentive compatibility of our approach.

6.8 Summary

In this chapter, we presented the architecture of an incentives system to engage BSS users in the bike repositioning process. We worked in collaboration with a bike sharing company, designed and built a working system integrated with the BSS infrastructure. A key component of this incentives system is the dynamic pricing mechanism based on BP-UCB, demonstrating the application of online learning based mechanisms developed in this dissertation in real-world applications. Furthermore, we investigated the incentive compatibility of our system w.r.t. user’s private location and cost. We evaluated the proposed system through extensive simulations using historical and user survey data, and deployed it through a smartphone app for a period of 30 days in the city of Mainz, Germany.

In this chapter, we focused on shared mobility systems based on bike sharing—similar problems also arise in other domains, e.g., car sharing or rerouting users on overbooked flights, and our ideas can also be applied to these systems.
Part III

Incentives for Learning
Overview of Part III

Part II of this dissertation focused on developing machine learning techniques to learn about the users’ preferences for offering optimized incentives. Our focus so far has been on learning about the incentives. A different facet in the interplay of learning and incentives is the design of incentives for data collection by incentivizing users to solving complex sensing and computation tasks. This ability to collect data from users, in turn, is an important aspect governing the performance of large-scale machine learning and AI systems by providing access to training data at scale. In Part III of this dissertation, we explore the aspect of Incentives for Learning; some of the key research questions in this direction of research include:

*How can we design privacy-aware incentive-compatible mechanisms for learning and optimal information gathering? I.e., how should we recruit users to solve complex sensing and computation tasks?*

One specific problem that we focus on in the subsequent chapters concerns community sensing, a new paradigm for creating efficient and cost-effective sensing applications by harnessing the data of large populations of sensors. This work is partly done in the context of the OpenSense [Ope] project (a Swiss nationwide project) with the goal of creating a community-based solution for air quality sensing and understanding the impacts of air pollution exposure on the health of citizens. In Chapter 8, we design
privacy-aware mechanisms that can valuate and negotiate access to the sensing data of the participants in an incentive-compatible manner. Our proposed mechanism—SeqTGREEDY—is based on the key insight that privacy tradeoffs can be cast as an adaptive submodular optimization problem for a large class of sensing applications. In Chapter 9, we perform a case study of air quality monitoring and collect data from a survey. In this survey, we asked users to report on their willingness as well as bids to participate in such community sensing applications. We perform extensive experiments using the survey data and demonstrate the effectiveness of our approach. In the context of the OpenSense project, we record air pollution measurements using a bike equipped with CO2 gas sensors from the NODE+ platform [Nod] and we report our findings in Chapter 9.

7.1 Relevant Research Problems and Our Contributions

In the rest of this chapter, we provide a summary of our work that tackles interrelated research problems relevant to the aspect of Incentives for Learning. These results are not covered in the rest of this dissertation, primarily to keep the content of this dissertation concise.

In this dissertation, our privacy-aware mechanisms for information gathering are presented primarily in the context of community sensing applications. Similar challenges, at the interplay of incentives and privacy, also arise in other applications, for instance, web search or e-commerce applications and online social networks. In our work in Singla et al. [Sin+14c] and Singla et al. [Sin+15b], we study the design of privacy-aware information gathering methodologies in the context of web search and social networks, respectively.

7.1.1 Privacy and Incentives for Sharing Data on the Web [Sin+14c]

Online services such as web search and e-commerce applications typically rely on the collection of data about users, including details of their activities on the web. Such personal data is used to maximize revenues via targeting of advertisements and longer engagements of users, and to enhance the quality of service via personalization of content. Permissions are typically obtained via broad consent agreements that request
user permission to share their data through system dialogs or via complex Terms of Service. Such notices are typically difficult to understand and are often ignored [Tec12]. In other cases, a plethora of requests for information, such as attempts to gain access to users’ locations, may be shown in system dialogs at run time or installation time.

One of the key challenges is to provide understandable approaches to enhancing privacy while enabling rich, personalized online services. In our work Singla et al. [Sin+14c], we introduce a new approach to privacy that we refer to as stochastic privacy. Stochastic privacy centers on providing a guarantee to users about the likelihood that their data will be accessed and used by a service provider. We refer to this measure as the assessed or communicated privacy risk, which may be increased in return for increases in the quality of service or other incentives. We demonstrate the methodology with a case study and evaluation of the procedures applied to web search personalization.

7.1.2 Privacy-Aware Information Gathering on Social Networks [Sin+15b]

The proliferation of online social networks in the last decade has opened up numerous opportunities of how people exchange information with each other. Recently, there has been an increasing interest in building applications that can leverage this social connectivity. For instance, Bernstein et al. [Ber+10] introduced the concept of friendsourcing, a form of crowdsourcing aimed at collecting accurate information available from social networks. Richardson and White [RW11] developed IM-an-Expert, a synchronous social Q&A system, where a user can pose a question via instant messaging that is then routed to an expert by the centralized system. However, implementing such a centralized system having access to the entire network is unrealistic in many real-world applications (such as Facebook or LinkedIn) because of concerns of privacy and disrupting the users. In fact, the privacy constraint that restricts a node’s visibility to friends is one of the key characteristics of these online social networks.

Motivated by the real-world applications of information seeking in networks, such as surveying or task routing in social networks and team formation in collaborative networks, the key question is: How should we gather information in a network, where each node’s visibility is limited to its local neighborhood? In our work in Singla et al. [Sin+15b], we present a formal model of information gathering in networks with lo-
cal visibility constraints. We develop a novel algorithm for this problem that actively explores the accessible local neighborhood to increase the visibility of the unseen network, while at the same time exploiting the value of the information available in this neighborhood. We analyze the performance of our algorithm and provide theoretical bounds on its performance dependent on structural properties of the underlying network. We evaluate our approach on data collected from a real-world application of task routing in a social Q&A system deployed within a large enterprise, to show the practical applicability of our methodology.
Incentives for Privacy Tradeoffs via Adaptive Submodularity

In this chapter, we develop privacy-aware mechanisms for information gathering from strategic agents. For the ease of presentation, we will formalize our problem and develop the mechanisms in the context of community sensing applications. We note that our approach and mechanisms in this chapter are general, and of independent interest, both theoretically as well as for many potential applications (e.g., viral marketing) dealing with information acquisition from strategic agents under uncertainty. In the next chapter, we will apply our results to a case study of community-based air quality sensing.

Community sensing, fusing information from populations of privately-held sensors, presents a great opportunity to create efficient and cost-effective sensing applications. For example, the accelerometer data from smartphone users could be used for earthquake detection and fine-grained analysis of seismic events. Velocity data from GPS devices (in smartphones or automobiles) could be used to provide real-time traffic maps or detect accidents.

However, accessing this stream of private sensor data raises reasonable concerns about the privacy of individual users [Lie07; Wun+07; OGH05]. For example, mobility patterns and the house or office locations of a user could possibly be inferred from their
GPS tracks [Kru07]. Irrespective of the models of privacy we consider [Swe02; Dwo06; Mac+06], the key concern is about identifiability as users become members of increasingly smaller groups of people sharing the same characteristics inferred from data. Beyond concerns about sharing sensitive information, there are general anxieties among users about sharing data from their private devices. These concerns limit the practical applicability of deploying such applications. In this light, the key research questions that need to be addressed include:

How should systems valuate and negotiate access to private information, for example in return for monetary incentives? How should they optimally choose the participants from a large population of strategic users with privacy concerns, and compensate them for information shared?

In this chapter, we work towards addressing these questions and propose a principled approach to negotiating access to certain private information in an incentive-compatible manner.

Organization of this Chapter

This chapter is organized as follows. In Section 8.1, we give a high-level overview of our approach. Section 8.2 formalizes the problem addressed in this chapter. In Section 8.3, we first review existing mechanisms that fall short of either privacy-preservation, adaptivity, or truthfulness. In Section 8.4, we develop our mechanism $SEQGReEDY$ and then prove the desirable properties of the mechanism in Section 8.5. Section 8.6 discusses the related work and Section 8.7 presents our conclusions.

Relevant Publication

Results presented in this chapter are published in Singla and Krause [SK13a].

8.1 Overview of Our Approach

We begin by providing a high-level overview of our approach.
8.1. Overview of Our Approach

8.1.1 Incentives to Participants for Privacy Tradeoff

Olson, Grudin, and Horvitz [OGH05] show that people’s willingness to share information depends greatly on the type of information being shared, with whom the information is shared, and how it is going to be used. They are willing to share certain private information if compensated in terms of their utility gain [KH08].

In our approach, we are exploring the design of intelligent systems that empower users to consciously share certain private information in return of, e.g., monetary or other form of incentives. We model the users as strategic agents who are willing to negotiate access to certain private information, aiming to maximize the monetary incentives they receive in return. Empowering users to opt into such negotiations is the key idea that we explore in this chapter.

8.1.2 Obfuscation Protocol and Interaction with Users

In our approach, the key idea towards protecting users’ privacy is that they communicate only their obfuscated locations until they are (if ever) recruited by the system. We note that such an obfuscation protocol is easy to implement in practice, for example, via smartphone app whereby users declare their obfuscated locations for which they could sense the data. However, this leads to technical challenges in developing algo-
Figure 8.1 illustrates the protocol via which our system interacts with the users. As a basis for selection, the community sensing system receives obfuscated estimates of the private attributes. For concreteness, we focus on sensor location as private information, but our approach generalizes to other attributes. The users also declare a bid or cost as the desired monetary incentive for participation and hence privacy tradeoff. After receiving the bids, the mechanism sequentially selects a participant, commits to make her the payment, receives the actual private information, selects the next participant and so on. At the end, all selected participants are provided the agreed payment.

8.1.3 Our Results

Our goal is to design policies for selecting (and compensating) the participants, which provide near-optimal utility for the sensing application under strict budget constraints. We model the participants as strategic agents who aim to maximize their profit, by possibly misreporting their private costs. As a consequence, we require the mechanism to be truthful. In order to capture a large class of sensing applications, we only require the utility function to satisfy submodularity, a natural diminishing returns condition [NWF78; KG07]. To design our mechanism, we first reduce the sequential negotiation of the privacy tradeoff to the problem of adaptive submodular maximization [ANS08; GK11]. Then, we extend recent results on truthful budget feasible mechanisms for submodular functions (cf., [Sin10; CGL11; Sin12]) to the adaptive setting.

Our main technical contribution is a novel mechanism, SeqTGreedy, for budgeted recruitment of strategic participants, which achieves near-optimal utility for a large class of sensing applications. The mechanism is general and of independent interest, suitable also for other applications, e.g., viral marketing.

8.2 Problem Statement

We now formalize the problem addressed in this paper.
8.2. Problem Statement

8.2.1 Sensing Phenomena

We focus on community sensing applications with the goal to monitor some spatial phenomenon, such as air quality or traffic. We discretize the environment as a finite set of locations $\mathcal{V}$, where each $v \in \mathcal{V}$ could, e.g., denote a zip code or more fine grained street addresses, depending on the application. We quantify the utility $f(\mathcal{A})$ of obtaining measurements from a set of locations $\mathcal{A}$ using a set function $f : 2^\mathcal{V} \to \mathbb{R}$. Formally, we only require that $f$ is nonnegative, monotone (i.e., whenever $\mathcal{A} \subseteq \mathcal{A}' \subseteq \mathcal{V}$ it holds that $f(\mathcal{A}) \leq f(\mathcal{A}')$) and submodular. Submodularity is an intuitive notion of diminishing returns, stating that, for any sets $\mathcal{A} \subseteq \mathcal{A}' \subseteq \mathcal{V}$, and any fixed location $a \notin \mathcal{A}'$ it holds that $f(\mathcal{A} \cup \{a\}) - f(\mathcal{A}) \geq f(\mathcal{A}' \cup \{a\}) - f(\mathcal{A}')$. As a simple, concrete example, we may derive some nonnegative value $d_a$ for observing each location $a \in \mathcal{A}$, and may define $f(\mathcal{A}) = \sum_{a \in \mathcal{A}} d_a$. More generally, sensing at location $a \in \mathcal{V}$ may actually cover a subset $\mathcal{S}_a$ of nearby locations, and $f(\mathcal{A}) = \sum\{d_j : j \in \bigcup_{a \in \mathcal{A}} \mathcal{S}_a\}$. These conditions are rather general, satisfied by many sensing utility functions and $f$ can capture much more complex notions, such as reduction of predictive uncertainty in a probabilistic model [KG07].

8.2.2 Sensing Profile of Users

We consider a community $\mathcal{W}$ of $|\mathcal{W}| = N$ users, owning some sensing device such as a smartphone. Each user can make observations at a set of locations depending on her geolocation or mobility as well as the type of device used. We model this through a collection of sensing profiles $\mathcal{O} \subseteq 2^\mathcal{V}$ whereby we associate each user $w \in \mathcal{W}$ with a profile $y_w \in \mathcal{O}$, specifying the set of locations covered by her. This set $y_w$ could be a singleton $y_w = \{a\}$ for some $a \in \mathcal{V}$, modeling the location of the user at a particular point in time, or could model an entire trajectory visiting multiple locations in $\mathcal{V}$. We denote a given set of users $S \subseteq \mathcal{W}$ jointly with their sensing profiles as $y_S \subseteq \mathcal{W} \times \mathcal{O}$. The goal is to select set of users $S$ (also called participants) so as to maximize the utility of the sensing application given by $g(y_S) = f(\mathcal{A})$ where $\mathcal{A} = \bigcup_{s \in S} y_s$. We assume that each user’s maximal contribution to the utility is bounded by a constant $f_{\text{max}}$. 

85
Figure 8.2: The sensing region is uniformly discretized into a set of locations $\mathcal{V}$ indicated by the dots. (a) illustrates a population of users, along with their sensing profiles in (b). The set of users selected by the system in absence of privacy are shown in (c). However, to protect privacy, users only share an obfuscated location with the system in (d) and a collection of sensing profiles ($\{y_1^w, y_2^w, y_3^w\}$ for user $w$) in (e). The privacy profile of user $w$, given by $Y_w$, is the uniform distribution over these sensing profiles, given by $P(Y_w = y_i^w) = \frac{1}{3}$. (f) shows the selection of the participants in presence of uncertainty introduced by privacy profiles. The actual sensing profile is only revealed to the system after a user has been selected.

### 8.2.3 Privacy Profile of Users via Obfuscation

In order to protect privacy, we consider the setting where the exact sensing profiles $y_w$ of the users (containing, e.g., tracks of locations visited) are not known to the sensing system. Instead, $y_w$ is only shared after obfuscation with a random perturbation intended to reduce the risk of identifiability [Swe02; Dwo06]. The system’s highly uncertain belief about the sensing profile of user $w$ can therefore be represented as a (set-valued) random variable (also called *privacy profile*) $Y_w$ with $y_w$ being its real-
ization. For example, suppose \( y_w = \{a\} \) for some location \( a \) (i.e., the user’s private location is \( a \in \mathcal{V} \)). In this case, the user may share with the system a collection of locations \( a_1, \ldots, a_m \) containing \( a \) (but not revealing which one it is), w.l.o.g. \( a = a_1 \). In this case the distribution shared \( P(Y_w = \{a_i\}) = \frac{1}{m} \) is simply the uniform distribution over the candidate locations. Figure 8.2 illustrates the notions of sensing and privacy profiles for a user.

We use \( Y_W = [Y_1, \ldots, Y_N] \) to refer to the collection of all (independent) variables associated with population \( W \) and assume that \( Y_W \) is distributed according to a factorial joint distribution \( P(Y_W) = \prod_w P(Y_w) \). The sensing profile \( y_w \) (and the actual sensor data obtained from sensing at locations \( y_w \)) is revealed to the application only after it commits to provide the desired incentives to the user \( w \). Then, the goal is to select a set of users \( S \) to maximize \( \mathbb{E}_{Y_W}[g(y_S)] \), i.e., the expected utility, where the expectation is taken over the realizations of \( Y_W \) w.r.t. \( P(Y_W) \).

### 8.2.4 Incentive Structure for Privacy Tradeoff

We assume that users are willing to share certain non-sensitive private information in return for monetary incentives. Each user \( w \) has a private cost \( c_w \in \mathbb{R}_{\geq 0} \) that she experiences for her privacy tradeoff. Instead of revealing \( c_w \), she only reveals a bid \( b_w \in \mathbb{R}_{\geq 0} \). We are interested in truthful mechanisms, where it is a dominant strategy for a user to report \( b_w = c_w \), i.e., users cannot increase their profit (in expectation) by lying about their true cost. We assume that costs have known bounded support, i.e., \( c_w \in [c_{\min}, c_{\max}] \).

### 8.2.5 Optimization Problem

Given a strict budget constraint \( B \), the goal of the sensing application is to design a mechanism \( M \), which implements an allocation policy to select participants \( S \) and a payment scheme to make truthful payments \( \theta_s \) to each of the participants, with the goal of maximizing the expected utility. Instead of committing to a fixed set of participants \( S \) in advance (non-adaptive policy), we are interested in mechanisms that implement an adaptive policy taking into account the observations made so far (revealed sensing profiles of participants already selected) when choosing the next user. Formally, the goal of the mechanism is to adaptively select participants \( S^* \) along with the payments

---

87
\( \theta_{S^*} \), such that

\[
S^* = \arg \max_{S \subseteq W} E_{Y_W}[g(y_S)] \text{ subject to } \sum_{s \in S} \theta_s \leq B. \tag{8.1}
\]

Here, the set of participants \( S \) selected and the payments \( \theta_S \) may depend on the realization of \( Y_W \) as well. We formally introduce adaptive policies in subsequent sections.

## 8.3 Existing Mechanisms

We first review existing mechanisms that fall short of either privacy-preservation, adaptivity, or truthfulness. In the next section, we then build on these and present our main contribution: a privacy-respecting, truthful, and adaptive mechanism.

### 8.3.1 Non-Private Mechanisms

Consider first an unrealistic setting, where the system has full information about the users’ exact sensing profiles and their true costs. In such a setting, Problem 8.1 reduces to that of budgeted maximization of a monotone non-negative submodular function with non-uniform costs, studied by Sviridenko [Svi04]. A simple algorithm combining partial enumeration with greedy selection guarantees a utility of at least \( (1 - 1/e) \) \((= 0.63)\) times that obtained by optimal selection \( \text{Opt} \). This result is tight under reasonable complexity assumptions [Fei98]. We denote this setting and mechanism as Greedy. Note that each participant is paid their true cost in this untruthful setting. Now, consider the non-private setting with unknown true costs. The problem then requires designing a truthful budget feasible mechanism for monotone submodular set functions, as done by Singer [Sin10], Chen, Gravin, and Lu [CGL11], and Singer [Sin12]. In this setting, a constant factor \( 1/7.91 \) \((= 0.13)\) approximation compared to \( \text{Opt} \) can be achieved, using a mechanism that we will refer to as \( \text{TGreedy} \). \( \text{TGreedy} \) executes a greedy allocation on a reduced budget with carefully chosen stopping criteria (for ensuring budget feasibility), in order to select a set of participants and then computes the truthful payments to be made to them.
8.3. Existing Mechanisms

<table>
<thead>
<tr>
<th></th>
<th>Untruthful</th>
<th>Truthful</th>
</tr>
</thead>
<tbody>
<tr>
<td>Privacy off</td>
<td>Greedy</td>
<td>TGreedy</td>
</tr>
<tr>
<td>Privacy on (Non-Adaptive)</td>
<td>ConstGreedy</td>
<td>ConstTGreedy</td>
</tr>
<tr>
<td>Privacy on (Adaptive)</td>
<td>SeqGreedy</td>
<td>SeqTGreedy</td>
</tr>
</tbody>
</table>

Table 8.1: Different information settings and existing mechanisms that fall short of either privacy-preservation, adaptivity, or truthfulness. Our main mechanism SeqTGreedy satisfies all the desirable properties.

8.3.2 Non-Adaptive Mechanisms with Privacy

In our case, where privacy is preserved through random obfuscation, one must deal with the stochasticity caused by the uncertainty about users’ sensing profiles. Here, the objective

$$G(S) \equiv \mathbb{E}_{Y_W}[g(Y_S)] = \sum_{y_W} P(Y_W = y_W)f\left(\bigcup_{s \in S} y_s\right)$$

in (8.1) can be seen as an expectation over multiple submodular set functions, one for each realisation of the privacy profile variables $Y_W$. However, as submodularity is preserved under expectations, the set function $G(S)$ is submodular as well. One can therefore still apply the mechanisms Greedy and TGreedy in order to obtain near-optimal non-adaptive solutions (i.e., the set of participants is fixed in advance) to Problem (8.1). We denote these non-adaptive (constant) mechanisms applied to our privacy-preserving setting as ConstGreedy and ConstTGreedy.

8.3.3 Untruthful, Adaptive Mechanisms with Privacy

Instead of non-adaptively committing to the set $S$ of participants a priori, one may wish to obtain increased utility through adaptive (active/sequential) selection, i.e., by taking into account the observations from the users selected so far when choosing the next user. Without assumptions, computing such an optimal policy for Problem (8.1) is intractable. Fortunately, as long as the sensing quality function $f$ is monotone and submodular, Problem (8.1) satisfies a natural condition called adaptive submodularity [GK11]. This condition generalizes the classical notion of submodularity to problems with sequential decision making or active selection as faced here.
Chapter 8. Incentives for Privacy Tradeoffs via Adaptive Submodularity

Adaptive Submodularity

For the condition of adaptive submodularity to hold in our setting, the key requirement is that the expected benefit of any fixed user $w \in W$ given a set of observations (i.e., set of users and observed sensing profiles) can never increase as we make more observations. Formally, consider the conditional expected marginal gain of adding a user $w \in W \setminus S$ to an existing set of observations $y_S \subseteq W \times O$:

$$
\Delta_g(w|y_S) = \mathbb{E}_{Y_w}[g(y_S \cup \{(w,y_w)\}) - g(y_S)|y_S]
= \sum_{y \in O} P(Y_w = y|y_S) \cdot [g(y_S \cup \{(w,y)\}) - g(y_S)].
$$

Function $g$ with distribution $P(Y_W)$ is adaptive submodular, if

$$
\Delta_g(w|y_S) \geq \Delta_g(w|y_{S'}) \quad \text{whenever } y_S \subseteq y_{S'}.
$$

Thus, the gain of a user $w$, in expectation over its unknown privacy profile, can never increase as we select and obtain data from more participants.

**Proposition 8.1.** Suppose $f$ is monotone and submodular. Then the objective $g$ and distribution $P$ used in Problem 8.1 are adaptive submodular.

Above Proposition follows from Theorem 6.1 of Golovin and Krause [GK11], assuming distribution $P$ is factorial (i.e., the random obfuscation is independent between users). Given this problem structure, for the simpler, untruthful setting (i.e., known true costs), we can thus use the sequential greedy policy for stochastic submodular maximization studied by Golovin and Krause [GK11]. This approach is denoted by SeqGreedy and obtains a utility of at least $(1 - 1/e) (= 0.63)$ times that of optimal sequential policy SeqOpt.

Table 8.1 summarizes the settings and mechanisms considered so far. They all fall short of at least one of the desired characteristics of privacy-preservation, truthfulness, or adaptivity. In the next section, we present our main contribution—SeqTGreedy—an adaptive mechanism for the realistic setting of privacy-sensitive and strategic agents.

### 8.4 Our Main Mechanism: SeqTGreedy

We now describe our mechanism, SeqTGreedy, given by $M = (\pi_M, \theta_M)$, with allocation policy $\pi_M$ and payment scheme $\theta_M$. $M$ first obtains the bids $B_W$ and privacy
8.4. Our Main Mechanism: SeqTGreedy

profiles $P(Y_W)$ from all users, runs the allocation policy $\pi_M$ to adaptively select participants $S$ and makes observations $y_S$ during selection. At the end, it computes payments $\theta_S$ using scheme $\theta_M$. The allocation policy $\pi_M$ can be thought of as a decision tree. Formally, a policy $\pi : 2^{W \times O} \rightarrow W$ is a partial mapping from observations $y_S$ made so far to the next user $w \in W \setminus S$ to be recruited, denoted by $\pi(y_S) = w$. We seek policies that are provably competitive with the optimal (intractable) sequential policy SeqOpt. $\theta_M$ computes payments which are truthful in expectation (a user cannot increase her total expected profit by lying about her true cost, for a fixed set of bids of other users) and individually rational ($\theta_s \geq b_s$). For budget feasibility, the allocation policy needs to ensure that the budget $B$ is sufficient to make the payments $\theta_S$ to all selected participants. Next, we describe in detail the allocation policy and payment scheme of SeqTGreedy with these desirable properties.

8.4.1 Allocation Policy

Algorithm 8.1 presents the allocation policy of SeqTGreedy. The main ingredient of the policy is to greedily pick the next user that maximizes the expected marginal gain $\Delta_S(w | y_S)$ per unit cost. The policy uses additional stopping criteria to enforce budget feasibility, similar to TGreedy [CGL11]. Firstly, it runs on a reduced budget $B/\alpha$. Secondly, it uses a proportional share rule ensuring that the expected marginal gain per unit cost for the next potential participant is at least equal to or greater than the expected utility of the new set of participants divided by the budget. We shall prove below that $\alpha = 2$ achieves the desired properties.

8.4.2 Payment Characterization

The payment scheme is based on the characterization of threshold payments used by TGreedy [Sin10]. However, a difficulty arises from the fact that the computation of payments for a participant depends also on the unallocated users, whose sensing profiles are not known to the mechanism. Let $S$ denote the set of participants allocated by $\pi_M$ along with making observations $y_S$. Let us consider the set of all possible realizations of $Y_W = Y_W \subseteq W \times O$ consistent with $y_S$, i.e., $y_S \subseteq y_W$. We denote this set by $Z_{W,S} = [y^1, y^2 \ldots y^r \ldots y^Z]$, where $Z = |Z_{W,S}|$. We first discuss how to compute the payment for each one of these possible realizations $y^r \in Z_{W,S}$, denoted by $\theta^d(y^r)$.
Algorithm 8.1: Allocation Policy of SEQTGREEDY

1. **Input**: budget $B$; users $\mathcal{W}$; privacy profiles $Y_\mathcal{W}$; bids $B_\mathcal{W}$; reduced budget factor $\alpha$;
2. **Initialize**:
   - **Outputs**: participants $S \leftarrow \emptyset$; observations $y_S \leftarrow \emptyset$; marginals $\Delta_S \leftarrow \emptyset$;
   - **Variables**: remaining users $\mathcal{W}' \leftarrow \mathcal{W}$;
3. **begin**
   4. while $\mathcal{W}' \neq \emptyset$ do
   5.     $w^* \leftarrow \arg\max_{w \in \mathcal{W}'} \frac{\Delta_S(w|y_S)}{b_w}$;
   6.     $\Delta_{w^*} \leftarrow \Delta_S(w^*|y_S)$;
   7.     if $B_S + b_{w^*} \leq B$ then
   8.         if $b_{w^*} \leq \frac{B}{\alpha} \cdot \frac{\Delta_{w^*}}{\left(\sum_{s \in S} \Delta_S + \Delta_{w^*}\right)}$ then
   9.             $S \leftarrow S \cup \{w^*\}$; $\Delta_S \leftarrow \Delta_S \cup \{\Delta_{w^*}\}$;
   10.            Observe $y_{w^*}$; $y_S \leftarrow y_S \cup \{(w^*, y_{w^*})\}$;
   11.            $\mathcal{W}' \leftarrow \mathcal{W}' \setminus \{w^*\}$;
   12.         else
   13.             $\mathcal{W}' \leftarrow \emptyset$;
   14.         end
   15.     else
   16.         $\mathcal{W}' \leftarrow \mathcal{W}' \setminus \{w^*\}$;
   17.     end
   18. end
4. **Output**: $S$; $y_S$; $\Delta_S$

(Where $d$ indicates here an association with the deterministic setting of knowing the exact sensing profiles of all users $w \in \mathcal{W}$). These payments for specific realizations are then combined together to compute the final payment to each participant.
8.4. Our Main Mechanism: \textsc{SeqGreedy}

**Payment \( \theta_s^d \) for a Given \( y_W \)**

Consider the case where the variables \( Y_W \) are in state \( y_W \in \mathbb{Z}_{W,S} \) and let \( S \) be the set of participants allocated by the policy. We use the well-known characterization of Myerson [Mye81] of truthful payments in single-parameter domains. It states that a mechanism is truthful if (i) the allocation rule is monotone (i.e., an already allocated user cannot be unallocated by lowering her bid, for a fixed set of bids of others) and (ii) allocated users are paid threshold payments (i.e., the highest bid they can declare before being removed from the allocated set). Monotonicity follows naturally from the greedy allocation policy, which sorts users based on expected marginal gain per unit cost. To compute threshold payments, we need to consider a maximum of all the possible bids that a user can declare and still get allocated. We next explain how this can be done.

Let us renumber the users \( S = \{1, \ldots, i, \ldots, k\} \) in the order of their allocation and let us analyze the payment for participant \( s = i \). Consider running the policy on an alternate set \( \mathcal{W}' = \mathcal{W} \setminus \{i\} \) and let \( S' = \{1, \ldots, j, \ldots, k'\} \) be the allocated set (users renumbered again based on order of allocation when running the policy on \( \mathcal{W}' \)). \( \Delta_S \) and \( \Delta_{S'} \) are the marginal contributions of the participants in the above two runs of the policy. We define \( \Delta_{i(j)} \) to be the marginal contribution of \( i \) (from \( S \)) if it has to replace the position of \( j \) (in set \( S' \)). Now, consider the bid that \( i \) can declare to replace \( j \) in \( S' \) by making a marginal contribution per cost higher than \( j \), given by

\[
b_{i(j)} = \frac{\Delta_{i(j)} \cdot b_j}{\Delta'_j}.
\]

Additionally, the bid that \( i \) can declare must satisfy the proportional share rule, denoted by

\[
\rho_{i(j)} = \frac{B}{\alpha} \cdot \frac{\Delta_{i(j)}}{\left(\sum_{s' \in [j-1]} \Delta'_{s'} + \Delta_{i(j)}\right)}.
\]

By taking the minimum of these two values, we get \( \theta^d_{i(j)} = \min(b_{i(j)}, \rho_{i(j)}) \) as the bid that \( i \) can declare to replace \( j \) in \( S' \). The threshold payment for participant \( s = i \) is given by \( \theta^d_i = \max_{j \in [k'+1]} \theta^d_{i(j)} \).
Computing the Final Payment $\theta_s$

For each $y^r \in Z_{W,S}$, compute $\theta^d_{t,r} = \theta^d_t(y^r)$. The final payment made to participant $s$ is given by $\theta_s = \sum_{y^r \in Z_{W,S}} P(Y_W = y^r | y_S) \cdot \theta^d_{s,r}$. Note that the set $Z_{W,S}$ could be exponentially large, and hence computing the exact $\theta_s$ may be intractable. However, one can use sampling to get estimates of $\theta_s$ in polynomial time (using Hoeffding’s inequality to bound sample complexity) and thus implement an approximately truthful payment scheme to any desired accuracy. Further, note that the approximation guarantees of $\mathcal{M}$ do not require computation of the payments at all, and only require execution of the allocation policy, which runs in polynomial time.

8.5 Analysis of SEQGREEDY

We now analyze the mechanism and prove its desirable properties. The proofs of all theorems are presented in the Appendix B. We only sketch them here.

8.5.1 Truthfulness of the Mechanism

**Theorem 8.1.** SEQGREEDY is truthful in expectation, i.e., no user can increase her profit in expectation by lying about her true cost, for a fixed set of bids of other users.

Firstly, truthfulness of payments $\theta^d_{s,r}$ is proved for a considered realization $y^r$. This is done by showing the monotonicity property of the greedy allocation policy and proving the threshold nature of the payment $\theta^d_{s,r}$. Truthfulness of the actual payment $\theta_s$ follows from the fact that it is a linear combination of individually truthful payments $\theta^d_{s,r}$.

8.5.2 Individually Rationality

**Theorem 8.2.** Payments made by SEQGREEDY are individually rational, i.e., $\theta_s \geq b_s$.

This is proved by showing a lower bound of $b_s$ on each of the payments $\theta^d_{s,r}$ used to compute the final payment $\theta_s$. 

8.5.3 Budget Feasibility

**Theorem 8.3.** For $\alpha = 2$, \textsc{SeqTGreedy} is budget feasible, i.e., $\theta_S \leq B$. Moreover, an application specific tighter bound on $\alpha$ can be computed to better utilize the budget.

We first show that when full budget $B$ is used by mechanism, the maximum raise in bid $b'_s$ that a participant $s$ can make, keeping the bids of other users to be the same, to still get selected by mechanism is upper-bounded by

$$\alpha \cdot B \cdot \frac{\Delta_s}{(\sum_{s' \in S} \Delta_{s'})}.$$

By adapting the proof of Chen, Gravin, and Lu [CGL11], we prove that $\alpha$ is bounded by 2. Surprisingly, this payment bound on $\alpha$ holds irrespectively of the payment scheme used by the mechanism. Hence, when the budget is reduced by $\alpha = 2$, this results in an upper bound on the payments made to any participant by $B \cdot \Delta_s/(\sum_{s' \in S} \Delta_{s'})$. Summing over these payments ensures budget feasibility. Moreover, by adapting a proof from Singer [Sin10], we show that a tighter bound on $\alpha$ can be computed based on the characterization of threshold payments used by \textsc{SeqTGreedy}. Intuitively, the proof is based on the fact that a raise in the bid that a participant can make depends on how much utility the application would lose if she refused to participate.

8.5.4 Guarantees on the Utility

**Theorem 8.4.** For $\alpha = 2$, \textsc{SeqTGreedy} achieves a utility of at least $\left(\frac{e^{-1}}{3e} - \gamma\right)$ times that obtained by the optimal policy \textsc{SeqOpt} with full knowledge of the true costs. Hereby, $\gamma$ is the ratio of the participants’ largest marginal contribution $f_{\max}$ and the expected utility achieved by \textsc{SeqOpt}.

We show that, because of the diminishing returns property of the utility function, the stopping criteria used by the mechanism based on proportional share and using only an $\alpha$ proportion of the budget still allows the allocation of sufficiently many participants to achieve a competitive amount of utility. As a concrete example, if each participant can contribute at most 1% to the optimal utility (i.e., $\gamma = 0.01$), Theorem 8.4 guarantees a constant approximation factor of 0.20.
8.6 Related Work

In this section, we review some existing work in the literature concerning privacy and incentives in the context of location-based services, sensing, and information gathering from users.

8.6.1 Privacy, Incentives, and Sensing

The proliferation of smartphones and internet-enabled handheld devices has led to growing interests in applications related to the personalization of location-based services and targeted advertisements. This, in turn, has led to new challenges and research questions regarding privacy and incentives. Himmel et al. [Him+05] propose to provide users with rewards such as free minutes to motivate them to accept mobile advertisements. Hui et al. [Hui+11] develop MobiAd, a system for targeted mobile advertisements, by utilizing the rich set of information available on the phone and suggesting the service providers to give discounts to the users, in order to incentivize them to use the system. Liu, Krishnamachari, and Annavaram [LKA08] propose a game-theoretic model of privacy for social networking-based mobile applications and presents a tit-for-tat mechanism by which users take decisions about their exposed location obfuscation for increasing personal or social utility.

Chorppath and Alpcan [CA12] study a privacy game in mobile commerce, where users choose the degree of granularity at which to report their location and the service providers offer them monetary incentives under budget constraints. The best users’ response and the optimal strategy for the company are derived by analyzing the Nash equilibrium of the underlying privacy game. This is very different from our setting as we focus on algorithmic aspects of the mechanism in choosing the best set of users for participation in community sensing.

Li and Faltings [LF12] and Faltings, Jurca, and Li [FJL12] study the problem of incentivizing users in community sensing to report accurate measurements and place sensors in the most useful locations. While developing incentive-compatible mechanisms, they do not consider the privacy aspect.

Carrascal et al. [Car+13] study how users value their personally identifiable information (PII) while browsing. The experiments demonstrate that users have different
valuations, depending on the type and information content of private data. Higher valuations are chosen for offline PII, such as age and address, compared to browsing history. This work is complementary and supports the assertion that users indeed associate monetary valuations to certain private data.

8.7 Summary

There is much potential in intelligent systems that incentivize and empower their users to consciously share certain private information. In this chapter, we presented a principled approach for negotiating access to such private information in community sensing. By using insights from mechanism design and adaptive submodular optimization, we designed the first adaptive, truthful, and budget feasible mechanism guaranteed to recruit a near-optimal subset of participants.

Privacy tradeoff is a personal choice and sensitive issue. In realistic deployments of the proposed approach, the choice of participation ultimately lies with the users. We believe that this integrated approach connecting privacy, utility, and incentives provides an important step towards developing practical, yet theoretically well-founded techniques for community sensing.
In the previous chapter, we developed our privacy-preserving mechanism, SeqTGreedy, for information gathering from privacy-sensitive and strategic agents. We proved several desirable properties of our mechanism including budget feasibility, truthfulness, and approximation guarantees on the acquired utility.

In this chapter, we carry out extensive experiments to understand the practical performance of our mechanism on a case study of community-based air quality monitoring. This case study is motivated by and is done partly in the context of OpenSense [Abe+10; Ope]—a Swiss nationwide project for developing community-based solution for air quality sensing. As part of this case study, we surveyed users on a crowdsourcing platform to understand their willingness to participate in community sensing applications. We also elicited users’ bids for participation which we use in our experiments to test our approach on real-world inputs.

Then, we study the feasibility of a sensing modality with a bike equipped with CO2 gas sensors from the NODE+ platform [Nod]. Furthermore, we report on the field experiments we did to test and calibrate NODE+ CO2 gas sensors at EMPA [Emp]. We learn a spatial model of CO2 concentration from the collected data, which we use in our experiments to define a realistic utility function of the sensing phenomena.
Chapter 9. Applications to Community-Based Air Quality Sensing

Organization of this Chapter

This chapter is organized as follows. In Section 9.1, we provide details of the survey study and then analyze the obtained responses. In Section 9.2, we report our findings on sensor measurements recorded from NODE+ CO2 gas sensors. In Section 9.3, we describe the experimental setup including benchmarks, performance metrics, etc. Section 9.4 presents the experimental evaluation of our mechanisms using the data that we collected. Section 9.5 discusses the related work on applications of community sensing and Section 9.6 presents our conclusions.

Relevant Publication

The results presented in this chapter are partly based on Singla and Krause [SK13a].

9.1 Survey Study

In this section, we describe the survey that we did to understand the preferences of the users to participate in community sensing applications motivated by settings of the OpenSense project. This survey data is then used to simulate the cost distributions of the users in our experiments.

9.1.1 Setup of the Survey Study

We posted a Human Intelligence Task (HIT) on the Mechanical Turk platform in the form of a survey, where workers were told about an option to participate in a community sensing application. Our HIT clearly stated the purpose as purely academic, requesting workers to provide correct and honest information. The HIT presented the application scenario and asked workers about their willingness (yes/no) to participate in such applications. A total of 650 workers participated in our HIT, restricted to workers from the USA with more than 90% approval rate and were paid a fixed amount each. Workers were asked to express their sensitivity (on a scale of \([1 \rightarrow 100]\)), as well as the payment bids (in the range of \([1 \rightarrow 500]\) $) they desire to receive about exposing their location at the granularity of home address, zip, city, state, or country respectively.
9.2. The OpenSense Project: Data Collection and Spatial Modeling

In the OpenSense project, trams and buses of the public transport system in the city of Zurich and Lausanne are equipped with sensors for measuring air pollution, Li et al. [Li+12] provide details of this setup and the data collection process. However, there are some limitations of this approach, specifically in the context of building community sensing applications. First of all, these are expensive sensors, deployed only in limited capacity and thus provide limited spatial coverage. Ignoring the cost aspects, these sensors are usually bulky which cannot be carried out by people easily (e.g., while walking or riding a bike). Furthermore, setting up these devices require

Figure 9.1: (a) Bids ($) and sensitivity ([1–100]) for different levels of privacy tradeoff. (b) Distribution of bids for sharing location at a granularity level of zip codes.

9.1.2 Statistics of the Data

75% (487 out of 650) responded positively, showing their willingness to participate in the application—we used the data of these 487 workers for our experiments. Figure 9.1(a) shows the mean bids and expressed sensitivity for different levels of obfuscation. Figure 9.1(b) shows the distribution of bids for exposing zip-level location information: In our experiments, we will use this distribution for simulating the private costs of the users.
domain expertise, limiting the possibility to easily incorporate them into the daily lives of people (e.g., installing at home or on top of a personal vehicle).

While working in the context of the OpenSense project, we study the feasibility of a sensing modality whereby bikes of users can be equipped with light-weight, low-cost sensors. In this section, we report our findings based on using sensors from the NODE+ platform [Nod]. We then build a spatial model of CO2 concentration based on the CO2 measurements we collected in the city of Zurich. This spatial model forms the basis of our experiments in defining a realistic utility function.

9.2.1 Calibrating NODE+ CO2 Gas Sensors

In our study, we opt for sensors from the NODE+ wireless sensing platform [Nod], shown in Figure 9.2(a). NODE+ platform is based on a modular approach whereby the two extension sockets on both ends can be attached with different combinations of sensor modules—for our study, we use the CO2 gas sensor module. Furthermore, the platform features low energy Bluetooth connectivity with a connected smartphone, thereby allowing one to develop applications for real-time transfer of sensor measurements from NODE+ sensors to a backend server via user’s smartphone.

As our first step, in order to study the quality of the NODE+ CO2 gas sensor measurements, we performed calibration experiments at EMPA [Emp]. Figure 9.2(b) shows the setup used for performing the calibration experiments. We then compare the measurements of NODE+ sensors against the sensing device at EMPA, which serves as a baseline for the calibration. As illustrated in Figure 9.2(c), the NODE+ CO2 gas sensor has high-quality measurements showing a good sensitivity and high correlation with the baseline sensor measurements.

9.2.2 Data Collection from NODE+ Sensors Mounted on a Bike

Next, we collect data from NODE+ CO2 gas sensors in real-world settings—we study the feasibility of a sensing modality with bikes of the users equipped with NODE+ sensors. We mounted two NODE+ CO2 gas sensors on a bike, as shown in Figure 9.3(a). Figure 9.3(b) shows the screenshot of NODE+ app running on the smartphone of the user (i.e., rider) displaying the measurements in real-time. In this study, we collected data of the CO2 measurements in the city of Zurich over several days.
9.2. The OpenSense Project: Data Collection and Spatial Modeling

Figure 9.2: (a) NODE+ wireless sensor platform with expansion sockets on both ends to attach different sensor modules. (b) shows the set up of calibration experiments at EMPA: NODE+ sensors are installed at the vent which feeds the air to the sensing device employed by EMPA. (c) shows that the NODE+ CO2 gas sensor has high-quality measurements.

Based on this collected data, we learn a spatial model of CO2 concentration by fitting a Gaussian variogram model to the empirical variogram of the data (cf. [CD09]). An important parameter of this model is the range and our fitted model has a range of 236 meters. Intuitively, it illustrates that based on our collected data via NODE+ sensors, the CO2 concentration of a given spatial point ceases to have any significant correlations with other points beyond this distance. These spatial models are useful in guiding the system to define realistic utility functions of the sensing phenomena and
deciding the density of the sensing network needed to achieve a desired utility. In the next section, we define a realistic utility function of the sensing phenomena, guided by this spatial model. Furthermore, our study serves as a first step towards building applications to perform data collection at scale, which can then be used to learn richer spatial-temporal models of air quality.

9.3 Experimental Setup

In this section, we describe the experimental setup of our case study, including our benchmarks, metrics, and the parameter choices made in the experiments.

9.3.1 Sensing Locations using the OpenStreetMap Data

In order to define the set of sensing locations $V$, we begin by considering an area of 2000 meters $\times$ 2000 meters in the center of New York City, shown in Figure 9.4(a). Then, we generate a set of discrete sensing locations in this area via using the publicly available data from OpenStreetMap\(^1\). More specifically, we use the nodes in the

\(^1\)http://wiki.openstreetmap.org/wiki/Node
OpenStreetMap data, which uniquely represents the point of intersections of roads and pathways in a city. Each node is defined by its latitude, longitude, and a unique identifier. There are over 2 billion nodes mapped worldwide, and a total of 786 nodes are present in the considered area of Figure 9.4(a). Out of these 786 nodes, we randomly subsample to select 300 nodes, representing the set of sensing locations $V$ used in our experiments (shown in red dots in Figure 9.4(a)).

9.3.2 Population of Users using the Strava Data

Our next step is to obtain a distribution over locations in the set $V$ to simulate a realistic population of the $N$ users across the city. We obtained a publicly accessible dataset from Strava—the social network of athletes\(^2\). In particular, this dataset contains a detailed activity of the bicycle rides taken by a random sample of about 4500 athletes/users in the New York City in the time period between 13\textsuperscript{th} July, 2015 and 19\textsuperscript{th} July, 2015. The heat map corresponding to these rides is shown in Figure 9.4(b). From this data, we assign a weight to each of the location $v \in V$ representing the count of the number of rides/activities that involved location $v$—let us denote this weight as $\text{rides}_v$. These weights, after appropriate normalization, are then used to define the probability distribution over the locations $V$. The population of $N$ users in our experiments is then sampled from this probability distribution.

9.3.3 Utility Function Based on the Spatial Model of CO2

The utility function, \textit{i.e.}, our objective $f$, is guided by the spatial model of CO2 concentration learnt in the previous section. The Gaussian variogram model learnt from the collected data has a range of 236 meters, illustrating the fact that correlations among the CO2 concentrations of two points cease to exist beyond this distance. Guided by this spatial model, we define a simple utility function based on the disk coverage model (cf., [HM85; Bai+06]), \textit{i.e.}, each sensor can perfectly observe everything within a fixed radius and nothing else. We used a fixed sensing radius of 150 meters associated with all the users—this represents a distance beyond which the value of Gaussian variogram model function falls below a value of 0.30. In order to encourage spatial coverage, we choose our objective $f$ such that one unit utility is obtained for every sensing location

\(^2\text{http://metro.strava.com/}\)
Chapter 9. Applications to Community-Based Air Quality Sensing

Figure 9.4: (a) Red dots represent a set of 300 sensing locations \( \mathcal{V} \) in the center of New York City obtained using the OpenStreetMap data. (b) Heat map of rides on Strava in the New York City used to get probability distribution of users across different locations.

observed by the selected participants. Our experimental evaluation can be extended to incorporate more complex objectives and we discuss this aspect in the future directions in Section 9.6.

9.3.4 Simulating User Profiles

We consider a population of size \( N = 1000 \), distributed across 300 locations (cf., Figure 9.4(a)) according to the probability distribution obtained via Strava data in Section 9.3.2. To generate private costs of the users, we used the distribution of bids reported for sharing location at a granularity level of zip codes (cf., Figure 9.1(b)). We set \( c_{\text{min}} = 0.10 \) and \( c_{\text{max}} = 0.90 \) by scaling the bids in this range. For a given location \( v \) of a user, the sensing radius of the user is set to 150 meters as discussed above in Section 9.3.3. This gives a maximum possible utility obtained from any user to be \( f_{\text{max}} = 28 \), representing the maximal number of observal locations within a sensing radius of 150 meters from any location \( v \in \mathcal{V} \).

Given a user’s location \( v \), the sensing profile of the user is uniquely specified. To create privacy profiles, we use obfuscated user locations by considering an obfuscation within a fixed radius centered around the user’s location—let us denote this obfuscation radius as \( \gamma \). More concretely, for a given user’s location \( v \) and obfuscation radius
9.3. Experimental Setup

Let \( \gamma \), let \( \mathcal{V}(v, \gamma) \) represents the set of locations which are within \( \gamma \) distance from \( v \). The main idea of the obfuscation protocol is to obfuscate the user’s true location \( v \) among the locations in the set \( \mathcal{V}(v, \gamma) \) where each location in \( \mathcal{V}(v, \gamma) \) is weighted according to the weights \( \text{rides}_v \) defined in Section 9.3.2. As a first step, the obfuscation protocol selects a random location \( v' \) from \( \mathcal{V}(v, \gamma) \) using the probability distribution defined via weights \( \text{rides}_v \), after appropriate normalization and limiting the support of distribution to locations in \( \mathcal{V}(v, \gamma) \). Then, the obfuscation protocol shares the obfuscated location \( v' \) along with a distribution over locations \( \mathcal{V}(v', \gamma) \) to the mechanism. Note that the user’s actual location \( v \) belongs to the set \( \mathcal{V}(v', \gamma) \). The distribution over the locations \( \mathcal{V}(v', \gamma) \)—defined via weights \( \text{rides}_x \forall x \in \mathcal{V}(v', \gamma) \)—denotes the user’s privacy profile.

9.3.5 Benchmarks and Baselines

We compare our mechanism \textsc{SeqGreedy} developed in this dissertation against the following benchmarks and baseline mechanisms.

- **\textsc{SeqGreedy}** (unrealistically) assumes access to the true costs of the users, thus measuring the loss incurred by \textsc{SeqGreedy} for enforcing truthfulness and serving as an upper bound benchmark on untruthful mechanisms.

- **\textsc{Random}** allocates users randomly until the budget is exhausted and pays each participant its true cost. This represents a lower bound benchmark on untruthful mechanisms.

- **\textsc{ConstGreedy}** is the non-adaptive variant of \textsc{SeqGreedy} and the state-of-the-art truthful mechanism.

- **\textsc{Greedy}** (unrealistically) assumes access to the exact sensing profiles of the users and hence provides insights in measuring the loss incurred due to privacy protection.

9.3.6 Metrics and Types of Experiments

The primary metric we measure is the utility acquired by the application. We also measure budget required to achieve a specified utility. To this end, we conduct experiments by varying the given budget and then varying the specified utility, for a fixed
Figure 9.5: (a) and (b) compares SEQGREEDY using $\alpha = 2$ w.r.t. a variant using an optimized value of $\alpha$.

obfuscation level. To further understand the impact of random obfuscation, we then vary the level of obfuscation and measure: (i) $\%$ Gain from adaptivity (SEQGREEDY vs. CONSTGREEDY), (ii) $\%$ Loss from truthfulness (SEQGREEDY vs. SEQGREEDY), and (iii) $\%$ Loss from privacy (SEQGREEDY vs. TGREEDY).

9.4 Experimental Results

We now discuss the findings from our experiments.

9.4.1 Computing Tighter Bounds on Payment

Based on Theorem 8.3 in the previous chapter, we compute tighter bounds on the payment and optimized the budget reduction factor $\alpha$ used by our mechanism in an application specific manner.

In community sensing applications with a large number of users and bounded maximal contribution from each user, $\alpha$ is close to 1, resulting in a utilization of almost the entire budget. In particular, we set optimized $\alpha = 1.1$ for our experiments. Figure 9.5 demonstrates the benefit of using tighter payment bounds compared to a mechanism simply using $\alpha = 2$. Henceforth, in the results, we use the optimized $\alpha = 1.1$ for all the truthful mechanisms.
9.4. Experimental Results

Figure 9.6: For a fixed obfuscation level of 400 meters radius, (a) varies the budget given and (b) varies the desired utility.

9.4.2 Varying the Given Budget and Specified Utility

For a fixed obfuscation level of 400 meters radius, Figures 9.6(a) and 9.6(b) show the effect of varying the given budget and desired utility respectively. Figure 9.6(a) illustrates the bounded approximation of our mechanism SEQ_GREEDY w.r.t. SEQ_GREEDY and about 5% improvement over CONST_GREEDY in terms of acquired utility. Figure 9.6(b) shows that the budget required to achieve a specified utility by our mechanism is larger w.r.t. SEQ_GREEDY and we achieve about 20% reduction in required budget by using the adaptive mechanism.

9.4.3 Varying Obfuscation Levels

Next, we perform experiments by varying the obfuscation levels, for a given budget or specified utility.

Utility Acquired at Different Obfuscation Levels

In Figures 9.7(a) and 9.7(b), the acquired utility is measured for a given budget of 10$ by varying the obfuscation level. We can see that adaptivity helps acquire about 5% higher utility. The loss from truthfulness is bounded (by 30%), agreeing with our
approximation guarantees. As expected, the loss from the lack of private information grows with increasing obfuscation.

**Budget Required at Different Obfuscation Levels**

In Figures 9.7(c) and 9.7(d), the required budget is computed for a desired utility value of 150 by varying the obfuscation level. We can see an adaptivity gain of up to a total of 30% reduction in required budget. As the privacy level increases, the loss from the lack of private information increases and dominates the adaptivity gain. However, for
a small obfuscation level of 250 meters (cf., Figure 9.7(d)), the adaptivity gain helps to partially recover the incurred loss from privacy in terms of the budget requirement.

9.5 Related Work

In this section, we review relevant work on community sensing applications.

9.5.1 Applications of Community Sensing

Several case studies have demonstrated the principal feasibility and usefulness of community sensing. A number of research and commercial prototypes are build, often relying on special campaigns to recruit volunteers [ZXM10] or on contracts with service providers to obtain anonymized data [Wun+07]. The SenseWeb system [Kan+07] has been developed as an infrastructure for sharing sensing data to enable various applications. Methods have been developed to estimate traffic [YNL07; MM08; Kra+08], perform forecasts about future traffic situations [Hor+05], or predict a driver’s trajectory [KH06]. Cell tower signals obtained from the service providers are leveraged for travel time estimation on roadways [Wun+07]. Additionally, captured images and video clips from smartphones have been used to link places with various categories [Cho+12]. Clayton et al. [Cla+12] describes the design of a Community Seismic Network to detect and monitor earthquakes using a dense network of low-cost sensors hosted by volunteers from the community. Aberer et al. [Abe+10] envisions a community driven sensing infrastructure for monitoring air quality in the context of OpenSense project. Ganti, Ye, and Lei [GYL11] provides an overview of mobile crowdsensing and review the existing applications. Furthermore, they discuss the current and future challenges in the context of an application of crowdsensing with users carrying personal air quality monitoring devices in combination with smartphones.

9.6 Summary

In this dissertation, we have developed an integrated approach to community sensing where users are offered monetary incentives in return for their participation and sharing of private information. In this chapter, we performed extensive experiments and
demonstrated the efficiency of our approach in a case study of community-based air quality monitoring.

There are some natural extensions for future work. For the experimental evaluation, we considered a fairly simple utility function for the sensing phenomena. A good next step is to incorporate more complex objectives in the experimental evaluation, e.g., reduction in predictive variance in a statistical model such as Gaussian Processes (GPs). It would also be useful to collect more data from the NODE+ sensors to build high-resolution spatial-temporal GP models for predicting air pollution.

Further, we would like to design a system (e.g., a smartphone app) for deploying our end-to-end approach in a real-world air quality sensing application. It would also be interesting to apply our mechanisms to other application domains that involve uncertainty, sequential decision-making, and strategic interactions, e.g., viral marketing.
Part IV

Learning as an Incentive
So far, we have developed new machine learning techniques to offer optimized incentives (cf., Part II) and explored the role of incentives for data collection which in turn helps to improve the performance of ML/AI systems (cf., Part III). The research work in Part IV of this dissertation tightly couples the interplay of learning and incentives. Our work is based on the idea that the participants in many crowd-powered systems have strong incentives to learn to increase their knowledge, for instance, volunteers in citizen science projects. Furthermore, learning acts as an incentive to improve the personal performance accuracy leading to higher monetary payments or better ranking in the leaderboard. This leads us to explore—Learning as an Incentive—the third facet of this interplay. The key question in this direction of research is:

How can we use learning itself as an incentive, by teaching participants in citizen science projects and crowdsourcing platforms to be more effective?

The specific problem that we study is: How should we present training examples to learners to teach them classification rules? This is a natural problem when training participants for labeling tasks in citizen science projects or crowdsourcing platforms.

In Chapter 11, we propose a natural stochastic model of the human learners, modeling them as switching among hypotheses via a probabilistic model based on the observed feedback. Our model generalizes existing models of teaching in order to increase robustness. We then develop a novel teaching algorithm, STRICT, that shows a sequence
of training examples to the learners in order to steer them towards the true hypothesis. Our solution greedily maximizes a submodular surrogate objective function in order to select examples to show to the learners. We prove that our teaching algorithm is competitive with the optimal teaching policy. Then, in Chapter 12, we carry out extensive experiments to compare the performance of our proposed algorithm with natural baselines. For experiments, we consider three different tasks of classifying animal species, motivated by the applications to biodiversity monitoring via citizen science such as the eBird project.

10.1 Relevant Research Problems and Our Contributions

In the rest of this chapter, we provide a summary of our work that tackles interrelated research problems relevant to the aspect of Learning as an Incentive. These results are not covered in the rest of this dissertation, primarily to keep the content of this dissertation concise.

10.1.1 Personalized Teaching Polices [Sin+13]

The work presented in this dissertation develops non-personalized teaching polices, i.e., we select the best representative set of the training examples to be shown to a user (or cohort of users) to improve their expected classification error. A natural extension of this work is to consider interactive, personalized models of teaching whereby the teaching examples shown to a particular user are adapted based on received responses. In our preliminary work in Singla et al. [Sin+13], we propose a natural Bayesian model of the learner: each learner is modeled as unique learning entity with an initial skill, competence, and dynamics. Our model is interactive, allowing for inferences about the learner’s current hypothesis by requesting labels, and incorporating noisy responses. We further formalize assumptions about how the learner transits between hypotheses based on observed feedback. We then propose a teaching algorithm that tracks the learner’s progress and adaptively chooses which examples to teach. An interesting research question to tackle is to analyze the complexity results for such interactive models of teaching.
Near-Optimally Teaching the Crowd to Classify

Citizen science projects such as eBird [Ebi; Sul+09] or GalaxyZoo [Gal; Lin+08] are powered by volunteers around the world to help conduct scientific research. For instance, over 50 millions of annotations have been provided by the volunteers on GalaxyZoo for classifying the telescopic images to identify the galaxies. Similarly, crowdsourcing services like Mechanical Turk [Mtu] platform are becoming vital for outsourcing information processing to large groups of workers: popular tasks include image annotation, transcription of audio and rating the relevance of web pages. Machine learning and AI systems can hugely benefit from the use of these services as large-scale annotated data is often of crucial importance [Sno+08; SF08; Lin+08].

Data collected from such services however is often noisy, primarily due to lack of expertise among the participants, due to spamming, or carelessness among the participants [SF08]. As the accuracy of the annotated data is crucial, the problem of tackling noise from crowdsourcing services has received considerable attention. Most of the work so far has focused on methods for combining labels from many annotators [Wel+10; Gom+11; Dal+13] or in designing control measures by estimating the participants’ reliabilities through gold standard questions [Sno+08]. In this chapter, we take a fundamentally different approach in solving this problem and explore an orthogonal
direction of research tackling the following question:

*Can we teach participants in crowdsourcing services in order to improve their accuracy?*

That is, instead of designing models and methods for determining participants’ reliability, can we develop intelligent systems that teach participants to be more effective? While we focus on crowdsourcing services in this chapter, similar challenges arise in other areas of data-driven education. As a running example, in this chapter, we focus on the task of image labeling via crowdsourcing. In particular, we consider the task of classifying animal species, an important component in several citizen science projects such as the *eBird* project with applications to biodiversity monitoring.

**Organization of this Chapter**

This chapter is organized as follows. In Section 11.1, we give a high-level overview of our approach. Section 11.2 describes our learning domain and teaching protocol. In Section 11.3, we introduce our model of the learner, and then develop the teaching algorithm in Section 11.4. In Section 11.5, we prove the strong theoretical approximation guarantees on the convergence of our algorithm. Section 11.6 discusses the related work and Section 11.7 presents our conclusions.

**Relevant Publication**

Results presented in this chapter are published in Singla et al. [Sin+14b].

**11.1 Overview of Our Approach**

We start with a high-level overview of our approach. Suppose we wish to teach the crowd to label a large set of images (*e.g.*, distinguishing butterflies from moths). How can this be done without already having access to the labels, or a set of informative features, for all the images (in which case crowdsourcing would be useless)? We suppose we have ground truth labels only for a small *teaching set* of examples. Our premise is that if we can teach a participant to classify this teaching set well, she can generalize to new images. In our approach, we first elicit—on the teaching set—a set of
11.1. Overview of Our Approach

Figure 11.1: Illustration of the process of teaching the crowd. Given a large set of images, the teacher (randomly) picks a small teaching set. For this set, expert labels, as well as candidate features and hypotheses used by the crowd are elicited (cf., Chapter 12). The teacher then uses this information to teach the rest of the crowd to label the rest of the data, for which no features or labels are available. The teacher sequentially provides an unlabeled example from the teaching set to the participant, who attempts an answer. Upon receipt of the correct label, the participant may update her hypothesis before the next example is shown.

candidate features as well as a collection of hypotheses (e.g., linear classifiers) that the crowd may be using. We will describe the concrete procedures of doing this elicitation, specific to the different classification tasks, in the next chapter (cf., Chapter 12). Having access to this information, we use a teaching algorithm to select training examples and steer the learner towards the target hypothesis.

Classical work on teaching classifiers (reviewed in Section 11.6), assumes that learners are noise-free: Hypotheses are immediately eliminated from consideration upon observation of an inconsistent training example. As we demonstrate in our real-world experiments on human learners (cf., Chapter 12) such approaches can be brittle. In contrast, we propose a noise-tolerant stochastic model of the learners, capturing our assumptions on how they incorporate training examples. We then (cf., Section 11.4) propose STRICT (Submodular Teaching for cRowdsourcIng ClassificaTion), a novel teaching algorithm that selects a sequence of training examples to the learners in order to steer them towards the true hypothesis. We theoretically analyze our approach, proving strong approximation guarantees and teaching complexity results.
Chapter 11. Near-Optimally Teaching the Crowd to Classify

11.2 Teaching Process

We now describe our learning domain and teaching protocol. As a running example, we consider the task of teaching to classify images, e.g., to distinguish butterflies from moths (see Figure 11.1). In Section 11.6, we review existing approaches to teaching classifiers.

11.2.1 The Learning Domain

Let $\mathcal{X}$ denote a set of examples (e.g., images), called the teaching set. We use $(x, y)$ to denote a labeled example where $x \in \mathcal{X}$ and $y \in \{-1, 1\}$. We denote by $\mathcal{H}$ a finite class of hypotheses. Each element of $\mathcal{H}$ is a function $h : \mathcal{X} \mapsto \mathbb{R}$. The label assigned to $x$ by hypothesis $h$ is $\text{sgn}(h(x))$. The magnitude $|h(x)|$ indicates the confidence hypothesis $h$ has in the label of $x$. For now, let us assume that $\mathcal{X}$ and $\mathcal{H}$ are known to both the teacher and the learner. For instance, in our image classification example, each image may be given by a feature vector $x$, and each hypothesis $h(x) = w_h^T x$ could be a linear function. In the next chapter (cf., Chapter 12), we discuss the concrete hypothesis spaces used in our crowdsourcing tasks, and how we can elicit them from the crowd.

11.2.2 The Teaching Protocol

The teacher has access to the labels $y(x)$ of all the examples $x$ in $\mathcal{X}$. We consider the realizable setting where $\mathcal{H}$ contains a hypothesis $h^*$ (known to the teacher, but not the learner) for which $\text{sgn}(h^*(x)) = y(x)$ for all $x \in \mathcal{X}$. The goal of the teacher is to teach the correct hypothesis $h^*$ to the learner. The basic assumption behind our approach is that if we can teach the participants to classify $\mathcal{X}$ correctly, then they will be able to generalize to new examples drawn from the same distribution as $\mathcal{X}$ (for which we neither have ground truth labels nor features). We will verify this assumption experimentally in the next chapter (cf., Chapter 12).

11.3 Model of the Learner

We now introduce our model of the learner, by formalizing our assumptions about how she adapts her hypothesis based on the training examples she receives from the
teacher. Generally, we assume that the learner is not aware that she is being taught. We assume that she carries out a random walk in the hypothesis space $H$: She starts at some hypothesis, stays there as long as the training examples received are consistent with it, and randomly jumps to an alternative hypothesis upon an observed inconsistency. Hereby, preference will be given to hypotheses that better agree with the received training.

More formally, we model the learner via a stochastic process, in particular a (non-stationary) Markov chain. Before the first example, the learner randomly chooses a hypothesis $h_1$, drawn from a prior distribution $P_0$. Then, in every round $t$ there are two possibilities: If the example $(x_t, y_t)$ received agrees with the label implied by the learner’s current hypothesis (i.e., $\text{sgn}(h_t(x_t)) = y_t$), she sticks to it: $h_{t+1} = h_t$. On the other hand, if the label $y_t$ disagrees with the learner’s prediction $\text{sgn}(h_t(x_t))$, she draws a new hypothesis $h_{t+1}$ based on a distribution $P_t$ constructed in a way that reduces the probability of hypotheses that disagreed with the true labels in the previous steps:

$$P_t(h) = \frac{1}{Z_t} P_0(h) \prod_{s=1}^{t} P(y_s | h, x_s)$$

with normalization factor

$$Z_t = \sum_{h \in H} P_0(h) \prod_{s=1}^{t} P(y_s | h, x_s).$$

In Equation (11.1), for some $\alpha > 0$, the term

$$P(y_s | h, x_s) = \frac{1}{1 + \exp(-\alpha h(x_s) y_s)}$$

models a likelihood function, encoding the confidence that hypothesis $h$ places in example $x_s$. Thus, if the example $(x_s, y_s)$ is strongly inconsistent with $h$ (i.e., $y_s \cdot h(x_s)$ takes a large negative value and consequently $P(y_s | h, x_s)$ is very small), then the learner will be very unlikely to jump to hypothesis $h$. The scaling parameter $\alpha$ allows to control the effect of observing inconsistent examples. The limit $\alpha \to \infty$ results in a behavior where inconsistent hypotheses are completely removed from consideration. This case precisely coincides with the noise-free learner models classically considered in the literature [GK92].
It can be shown (cf., Appendix C.2), that the marginal probability that the learner implements some hypothesis \( h \) in step \( t \) is equal to \( P_t(h) \), even when the true label and the predicted label agreed in the previous step. Though counter-intuitive, this is an important property this is used throughout the analysis of our teaching algorithm.

### 11.4 Teaching Algorithm: STRICT

Given the learner’s prior over the hypotheses \( P_0(h) \), how should the teacher choose examples to help the learner narrow down her belief to accurate hypotheses? By carefully showing examples, the teacher can control the learner’s progress by steering her posterior towards \( h^* \).

With a slight abuse of notation, if the teacher showed the set of examples \( A = \{x_1, \ldots, x_t\} \) we denote the posterior distribution by \( P_t(\cdot) \) and \( P(\cdot|A) \) interchangeably. We use the latter notation when we want to emphasize that the examples shown are the elements of \( A \). With the new notation, we can write the learner’s posterior after showing \( A \) as

\[
P(h|A) = \frac{1}{Z(A)} P_0(h) \prod_{y(x) \neq \text{sgn}(h(x))} P(y(x)|h, x).
\]

The ultimate goal of the teacher is to steer the learner towards a distribution with which she makes few mistakes. The expected error-rate of the learner after seeing examples \( A = \{x_1, \ldots, x_t\} \) together with their labels \( y_i = \text{sgn}(h^*(x_i)) \) can be expressed as

\[
\mathbb{E}[\text{err}_L | A] = \sum_{h \in H} P(h|A) \text{err}(h, h^*), \text{ where}
\]

\[
\text{err}(h, h^*) = \frac{|\{x \in X : \text{sgn}(h(x)) \neq \text{sgn}(h^*(x))\}|}{|X|}
\]

is the fraction of examples \( x \) from the teaching set \( X \) on which \( h \) and \( h^* \) disagree about the label. We use the notation \( \mathbb{E}[\text{err}_L] = \mathbb{E}[\text{err}_L | \{\}] \) as shorthand to refer to the learner’s error before receiving training.

Given an allowed tolerance \( \epsilon \) for the learner’s error, a natural objective for the teacher is to find the smallest set of examples \( A^* \) achieving this error, \( i.e.\):

\[
A^*_\epsilon = \arg \min_{A \subseteq X} |A| \text{ s.t. } \mathbb{E}[\text{err}_L | A] \leq \epsilon.
\]
We will use the notation $\text{OPT}(\epsilon) = |A^*_\epsilon|$ to refer to the size of the optimal solution achieving error $\epsilon$. Unfortunately, Problem (11.2) is a difficult combinatorial optimization problem. The following proposition, proved in the Appendix C.1, establishes hardness via a reduction from set cover.

**Proposition 11.1.** Problem (11.2) is NP-hard.

Given this hardness, in the following, we introduce an efficient approximation algorithm for Problem (11.2).

The first observation is that, in order to solve Problem (11.2), we can look at the objective function

$$R(A) = \mathbb{E}[\text{err}_L] - \mathbb{E}[\text{err}_L | A]$$

$$= \sum_{h \in H} (P_0(h) - P(h | A)) \text{err}(h, h^*),$$

quantifying the expected reduction in error upon teaching $A$. Solving Problem (11.2) is equivalent to finding the smallest set $A$ achieving error reduction $\mathbb{E}[\text{err}_L] - \epsilon$. Thus, if we could, for each $k$, find a set $A$ of size $k$ maximizing $R(A)$, we could solve Problem (11.2), contradicting the hardness.

The key idea is to replace the objective $R(A)$ with the following surrogate function:

$$F(A) = \sum_{h \in H} (Q(h) - Q(h | A)) \text{err}(h, h^*),$$

where $Q(h | A) = P_0(h) \prod_{x \in A} P(y(x) | h, x)_{y(x) \neq \text{sgn}(h(x))}$

is the unnormalized posterior of the learner. As shown in the Appendix C.3, this surrogate objective function satisfies submodularity, a natural diminishing returns condition. Submodular functions can be effectively optimized using a greedy algorithm, which, at every iteration, adds the example that maximally increases the surrogate function $F$

[1987]. We will show that maximizing $F(A)$ gives us good results in terms of the original, normalized objective function $R(A)$, that is, the expected error reduction of the learner. In fact, we show that running the algorithm until $F(A) \geq \mathbb{E}[\text{err}_L] - P_0(h^*)\epsilon$ is sufficient to produce a feasible solution to Problem (11.2), providing a natural stopping condition. We call the greedy algorithm for $F(A)$ as STRICT (Submodular Teaching for cRowdsourctng ClassificaTion), and describe it in Policy 11.1.
Chapter 11. Near-Optimally Teaching the Crowd to Classify

Policy 11.1: Teaching Policy STRICT

1. **Input:** examples $\mathcal{X}$, hypotheses $\mathcal{H}$, prior $P_0$, error $\epsilon$
2. **Output:** teaching set $A$
3. **Initialize:** $A \leftarrow \emptyset$
4. while $F(A) < \mathbb{E}[\text{err}_L] - P_0(h^*) \epsilon$ do
   5. $x \leftarrow \arg \max_{x \in \mathcal{X}} (F(A \cup \{x\}))$
   6. $A \leftarrow A \cup \{x\}$
end

Note that in the limit $\alpha \to \infty$, $F(A)$ quantifies the prior mass of all hypotheses $h$ (weighted by $\text{err}(h, h^*)$) that are inconsistent with the examples $A$. Thus, in this case, $F(A)$ is simply a weighted coverage function, consistent with classical work in noise-free teaching [GK92].

11.5 Performance Analysis

We now prove the strong theoretical approximation guarantees on the convergence of our algorithm to a desired error rate.

11.5.1 Approximation Guarantees

The following theorem ensures that if we choose the examples in a greedy manner to maximize our surrogate objective function $F(A)$, as done by Policy 11.1, we are close to being optimal in some sense.

**Theorem 11.1.** Fix $\epsilon > 0$. The STRICT Policy 11.1 terminates after at most

$$\text{OPT} \left( P_0(h^*) \epsilon / 2 \right) \log \frac{1}{P_0(h^*) \epsilon}$$

steps with a set $A$ such that $\mathbb{E}[\text{err}_L \mid A] \leq \epsilon$.

Thus, informally, Policy 11.1 uses a near-minimal number of examples when compared to any policy achieving $O(\epsilon)$ error (viewing $P_0(h^*)$ as a constant).
11.5. Performance Analysis

The main idea behind the proof of this theorem is that we first observe that \( F(A) \) is submodular and thus the greedy algorithm gives a set reasonably close to \( F \)'s optimum. Then we analyze the connection between maximizing \( F(A) \) and minimizing the expected error of the learner, \( \mathbb{E}[\text{err}_L \mid A] \). A detailed proof can be found in the Appendix C.3.

Note that maximizing \( F(A) \) is not only sufficient but also necessary to achieve \( \epsilon \) precision. Indeed, it is immediate that \( P(h \mid A) \geq Q(h \mid A) \), which in turn leads to

\[
\mathbb{E}[\text{err}_L \mid A] = \sum_{h \in \mathcal{H}} P(h \mid A) \text{err}(h, h^*) \geq \sum_{h \in \mathcal{H}} Q(h \mid A) \text{err}(h, h^*) = \mathbb{E}[\text{err}_L] - F(A).
\]

Thus, if \( \mathbb{E}[\text{err}_L] - F(A) > \epsilon \), then the expected posterior error \( \mathbb{E}[\text{err}_L \mid A] \) of the learner is also greater than \( \epsilon \).

11.5.2 Teaching Complexity for Linear Separators

Theorem 11.1 shows that greedily optimizing \( F(A) \) leads to low error with a number of examples not far away from the optimal. Now we show that, under some additional assumptions, the optimal number of examples is not too large.

We consider the important case where the set of hypotheses \( \mathcal{H} = \{h_1, h_2, \ldots, h_n\} \) consists of linear separators \( h(x) = w_h^T x + b_h \) for some weight vector \( w_h \in \mathbb{R}^d \) and offset \( b_h \in \mathbb{R} \). The label predicted by \( h \) for example \( x \) is \( \text{sgn}(w_h^T x + b_h) \).

We introduce an additional assumption, namely \( \lambda \)-richness of \((\mathcal{X}, \mathcal{H})\). First notice that \( \mathcal{H} \) partitions \( \mathcal{X} \) into polytopes (intersections of half-spaces), where within one polytope, all examples are labeled the same by every hypothesis, that is, within a polytope \( P \), for every \( x, x' \in P \subseteq \mathbb{R}^d \) and \( h \in \mathcal{H} \), \( \text{sgn}(h(x)) = \text{sgn}(h(x')) \). We say that \( \mathcal{X} \) is \( \lambda \)-rich if any \( P \) contains at least \( \lambda \) examples. In other words, if the teacher needs to show (up to) \( \lambda \) distinct examples to the learner from the same polytope in order to reduce her error below some level, this can be done.

**Theorem 11.2.** Fix \( \epsilon > 0 \). Suppose that the hypotheses are hyperplanes in \( \mathbb{R}^d \) and that \((\mathcal{X}, \mathcal{H})\) is \((8 \log^2 \frac{2}{\epsilon})\)-rich. Then the STRICT policy achieves learner error less than \( \epsilon \) after at most \( m = 8 \log^2 \frac{2}{\epsilon} \) teaching examples.
Chapter 11. Near-Optimally Teaching the Crowd to Classify

The proof of this theorem is in the Appendix C.4. In a nutshell, the proof works by establishing the existence (via the probabilistic method) of a teaching policy for which the number of examples needed can be bounded—hence also bounding the optimal policy—and then using Theorem 11.1.

11.6 Related Work

In the following, we review existing approaches to teaching classifiers.

11.6.1 Non-Interactive vs. Interactive Teaching

In their seminal work, [GK92] consider the non-interactive model: The teacher reveals a sequence of labeled examples, and the learner discards any inconsistent hypotheses (i.e., for which \( h(x) \neq y \) for any example \((x, y)\) shown). In these non-interactive models, teacher’s sequence of examples is completely independent of the learner’s responses. For a given hypothesis class, the Teaching dimension is the smallest number of examples required to ensure that all inconsistent hypotheses are eliminated. Goldman and Kearns [GK92] relate this notion of complexity to other notions in learning theory such as the VC dimension. Alternative notions of teaching complexity are introduced [Zil+11; DSZ10], which capture rational learners that reason about the teachers intent.

More recent work [BZ09a; Zil+11; DSZ10; DL11] consider models of interactive teaching, where the teacher, after showing each example, obtains feedback about the hypothesis that the learner is currently entertaining. Such feedback can be used to select future teaching examples in a more informed way. Balbach and Zeugmann [BZ09a] consider a setting where, after the teacher shows each labeled example \((x, y)\), the learner reveals his current hypothesis \( h \). In case \( h(x) \neq y \), the learner adopts an alternative hypothesis \( h' \). [BZ09a] prove that under some assumptions on the learner’s transition model, the teaching complexity (i.e., number of labels required) can be reduced. One example is the lazy learner model [BZ09a], in which hypotheses are endowed by a metric, and the learner is assumed to deterministically move to the closest consistent hypothesis.

While these interactive teaching models are theoretically intriguing, in this chapter we focused on non-interactive models, which are typically easier to deploy in practice.
11.6.2 Noise-Free vs. Noise-Tolerant Teaching

In existing methods, a broad separation can be made about assumptions that learners use to process training examples. *Noise-free* models assume learners immediately discard hypotheses inconsistent with observed examples (cf., [GK92] and Section 11.3). As our real-world experiments on human learners in Chapter 12 show, such models can be brittle in practice. In contrast, *noise-tolerant* models make less strict assumptions on how learners treat inconsistent hypotheses. In our work, we focus on the *noise-tolerant* setting, which fits naturally for many crowdsourcing services with human participants.

In contrast to the noise-free setting, the practically extremely important *noise-tolerant* setting is theoretically much less understood. In a recent work, Zhu [Zhu13] investigates the optimization problem of generating a set of teaching examples that trades off between the expected future error of the learner and the effort (i.e., number of examples) taken by the teacher, in the special case when the prior of the learner falls into the exponential family, and the learner performs Bayesian inference. Their algorithmic approach does not apply to the problem addressed in this chapter. Further, the approach is based on heuristically rounding the solution of a convex program, with no bounds on the integrality gap.

11.7 Summary

Our work in this chapter is inspired by the problem of tackling noise in the data collected from crowdsourcing services such as citizen science projects. We took a fundamentally different approach in solving this problem by exploring the design of intelligent systems which can effectively teach the participants. Our key idea is that the participants have strong incentives to learn to increase their knowledge or performance accuracy, and we explored a novel direction of research of teaching the participants by demonstration. We proposed a noise-tolerant stochastic model of the participants’ learning process in crowdsourcing classification tasks: Our model generalizes existing models of teaching in order to increase robustness. We then developed a novel teaching algorithm STRICT that exploits this model to teach the participants efficiently. We proved strong theoretical approximation guarantees on the convergence to a desired error rate.
Applications to Biodiversity Monitoring via Citizen Science

In the previous chapter, we proposed a mathematical model of the participants learning to annotate and classify data, e.g., images in a citizen science project. Then, we developed our teaching algorithm—STRICT—for selecting examples to teach classification rules to them. A natural next step is to validate the assumptions of our model and the effectiveness of our teaching algorithm in real-world settings. In this chapter, we present our experimental results, consisting of simulations and actual annotation tasks performed on a crowdsourcing platform.

In our experiments, we consider three different image classification tasks: (i) classification of synthetic insect images into two hypothetical species Vespula and Weevil (denoted as VW); (ii) distinguishing Butterflies and Moths on real images (denoted as BM); and (iii) identification of birds belonging to an endangered species of Woodpeckers from real images (denoted as WP). Our teaching process requires a known feature space for image dataset \( \mathcal{X} \) (i.e., the teaching set of images) and a hypothesis class \( \mathcal{H} \). While \( \mathcal{X} \) and \( \mathcal{H} \) can be controlled by design for the synthetic images, we illustrate different ways on how to automatically obtain a crowd-based embedding of the data for real images. We will discuss in detail the experimental setup of obtaining \( \mathcal{X} \) with its feature space and \( \mathcal{H} \) for the three different data sets used in our classification tasks.
Chapter 12. Applications to Biodiversity Monitoring via Citizen Science

Our primary performance metric is the test error (avg. classification error of the learners), of simulated or real participants on a hold-out test data set. We compare STRICT against two baseline teachers: RANDOM (picking uniformly random examples), and SETCOVER (the classical noise-free teaching model introduced in Chapter 11).

Organization of this Chapter

This chapter is organized as follows. In Sections 12.1, 12.2 and 12.3, we provide details of the three different tasks, as well as ways of obtaining the teaching set \( \mathcal{X} \) and a hypothesis class \( \mathcal{H} \). Section 12.4 presents the experimental results on simulated learners, and the results on real participants from a crowdsourcing platform are discussed in Section 12.5. Section 12.6 discusses the related work on approaches to crowd teaching. Section 12.7 presents our conclusions.

Relevant Publication

Results presented in this chapter are published in Singla et al. [Sin+14b].

12.1 Teaching Task: Identifying Vespula vs. Weevil

We first generate a classification problem using synthetic images \( \mathcal{X} \) in order to allow controlled experimentation. As a crucial advantage, in this setting the hypothesis class \( \mathcal{H} \) is known by design, and the task difficulty can be controlled. Furthermore, this setting ensures that participants have no prior knowledge of the image categories.

Dataset \( \mathcal{X} \) and Feature Space

We generated synthetic images of insects belonging to two hypothetical species: Weevil and Vespula. The task is to classify whether a given image contains a Vespula or not. The images were generated by varying body size and color as well as head size and color. Figure 12.1(a) shows sample images of the two species. A given image \( x_i \) can be distinguished based on the following two-dimensional feature vector \( x_i = [x_{i,1} = f_1, x_{i,2} = f_2] \) – (i) \( f_1 \): the head/body size ratio and (ii) \( f_2 \): head/body color contrast. Figure 12.1(b) shows the embedding of this data set in a two-dimensional
12.1. Teaching Task: Identifying Vespula vs. Weevil

Figure 12.1: Task of classification of synthetic insect images into two hypothetical species *Vespula* and *Weevil* (denoted as VW). (a) shows sample images of the dataset $\mathcal{X}$. (b) shows the 2-D embedding of synthetic images for the features: head/body size proportion ($f_1$) and head/body color contrast ($f_2$), normalized around origin. It shows four of the hypotheses in $\mathcal{H}$, with the target hypothesis $h^*$ in red.

space based on these two features. Figure 12.1(a) illustrates that Weevils have short heads with color similar to their body, whereas Vespula are distinguished by their big and contrasting heads. A total of 80 images per species were generated by sampling the features $f_1$ and $f_2$ from two bivariate Gaussian distributions: $(\mu = [0.10, 0.13], \Sigma = [0.12, 0; 0, 0.12])$ for Vespula and $(\mu = [-0.10, -0.13], \Sigma = [0.12, 0; 0, 0.12])$ for Weevil. A separate test set of 20 images per species were generated as well, for evaluating learning performance.

**Hypothesis Class $\mathcal{H}$**

As we know the exact feature space of $\mathcal{X}$, we can use any parametrized class of functions $\mathcal{H}$ on $\mathcal{X}$. In our experiments, we use a class of linear functions for $\mathcal{H}$, and further restrict $\mathcal{H}$ to eight clusters of hypotheses, centered at the origin and rotated by $\pi/4$ from each other. Specifically, we sampled the parameters of the linear hypotheses from the following multivariate Gaussian distribution: $(\mu_i = [\pi/4 \cdot i, 0], \Sigma_i = [2, 0; 0, 0.005])$, where $i$ varies from 0 to 7. Each hypothesis captures a different set of cues about the
Chapter 12. Applications to Biodiversity Monitoring via Citizen Science

features that participants could reasonably have: (i) ignoring a feature, (ii) using it as a positive signal for Vespula, and (iii) using it as a negative signal for Vespula. Amongst the generated hypotheses, we picked the target hypothesis $h^*$ as the one with minimal error on teaching set $\mathcal{X}$. In order to ensure realizability, we then removed any data points $x \in \mathcal{X}$ where $\text{sgn}(h^*(x)) \neq y(x)$. Figure 12.1(b) shows a subset of four of these hypotheses, with the target hypothesis $h^*$ represented in red. The prior distribution $P_0$ is chosen as uniform.

12.2 Teaching Task: Identifying Butterflies vs. Moths

Dataset Images $\mathcal{X}$

As our second dataset, we used a collection of 200 real images of four species of butterflies and moths from publicly available images on ImageNet [Ima]: (i) Peacock Butterfly, (ii) Ringlet Butterfly, (iii) Caterpillar Moth, and (iv) Tiger Moth, as shown in Figure 12.2(a). The task is to classify whether a given image contains a butterfly or not. While Peacock Butterfly and Caterpillar Moth are clearly distinguishable as butterflies and moths, Tiger Moth and Ringlet Butterfly are often considered hard to classify correctly. We used 160 of these images (40 per sub-species) as teaching set $\mathcal{X}$ and the remaining 40 (10 per sub-species) for testing.

Crowd-Embedding of $\mathcal{X}$

A Euclidean embedding of $\mathcal{X}$ for this image set is not readily available. Human-perceptible features for such real images may be difficult to compute. In fact, this challenge is one major motivation for using crowdsourcing in image annotation. However, several techniques do exist that allow estimating such an embedding from a small set of images and a limited number of crowd labels. In particular, we used the approach of Welinder et al. [Wel+10] as a preprocessing step. Welinder et al. [Wel+10] propose a generative Bayesian model for the annotation process of the images by the annotators (e.g., workers from Mechanical Turk platform or volunteers in citizen science projects) and then use an inference algorithm to jointly estimate a low-dimensional embedding of the data, as well as a collection of linear hypotheses—one for each annotator—that best explain their provided labels. We requested binary labels (of whether the image
12.2. Teaching Task: Identifying Butterflies vs. Moths

Figure 12.2: Task of distinguishing butterflies and moths on real images (denoted as BM). (a) shows sample images of the dataset $\mathcal{X}$. (b) shows the 2-D embedding of images of this data set, and the hypotheses for a small set of participants, as obtained using the approach of Welinder et al. [Wel+10].

contains a butterfly) for our teaching set $\mathcal{X}$, $|\mathcal{X}| = 160$, from a set of 60 participants from the Mechanical Turk platform. By using the software CUBAM [CUB], implementing the approach of Welinder et al. [Wel+10], we inferred a 2-D embedding of the data, as well as linear hypotheses corresponding to each of the 60 participants who provided the labels. Figure 12.2(b) shows this embedding of the data, as well as a small subset of participants’ hypotheses as colored lines.

**Hypothesis Class $\mathcal{H}$**

The 60 hypotheses obtained through the crowd-embedding provide a prior distribution over linear hypotheses that the participants in the crowd may have been using. Note that these hypotheses capture various idiosyncrasies (termed schools of thought by Welinder et al. [Wel+10]) in the participants’ annotation behavior, for instance, some participants were more likely to classify certain moths as butterflies and vice versa. To create our hypothesis class $\mathcal{H}$, we randomly sampled 15 hypotheses from these. Additionally, we fitted a linear classifier that best separates the classes and used it as target hypothesis $h^*$, shown in red in Figure 12.2(b). The few examples in $\mathcal{X}$ that disagreed
with \( h^* \) were removed from our teaching set, to ensure realizability.

**Teaching the Rest of the Crowd**

The teacher then uses this embedding and hypotheses in order to teach the rest of the crowd. We emphasize that the embedding is *not* required for test images. Furthermore, neither the participants nor the system used any information about sub-species in the images.

### 12.3 Teaching Task: Identifying Endangered Woodpecker Bird Species

**Dataset Images \( \mathcal{X} \)**

Our third classification task is inspired from the *eBird* citizen science project \([Ebi; Sul+09]\) and the goal of this task is to identify birds belonging to an endangered species of woodpeckers. We used a collection of 150 real images belonging to three species of woodpeckers from a publicly available dataset \([Wah+11]\), with one endangered species: (i) *Red-cockaded* woodpecker and other two species belonging to the least-concerned category: (ii) *Red-bellied woodpecker* and (iii) *Downy woodpecker*. Figure 12.3(a) shows sample images of the three species. On this dataset, the task is to classify whether a given image contains a Red-cockaded woodpecker or not. We used 80 of these images (40 per Red-cockaded, and 20 each per the other two species of the least-concerned categories) for teaching (*i.e.*, dataset \( \mathcal{X} \)). We also created a testing set of 20 images (10 for Red-cockaded, and 5 each for the other two species).

**Crowd-embedding of \( \mathcal{X} \)**

We need to infer an embedding and hypothesis space of the teaching set for our teaching process. While an approach similar to the one used for the BM task is applicable here as well, we considered an alternate option of using metadata associated with these images, elicited from the crowd, as further explained below.
12.3. Teaching Task: Identifying Endangered Woodpecker Bird Species

![Endangered Woodpeckers (WP)](image)

<table>
<thead>
<tr>
<th>Features</th>
<th>% feature presence in species</th>
<th>h^*</th>
</tr>
</thead>
<tbody>
<tr>
<td>has_breast_pattern: solid</td>
<td>22</td>
<td>-1</td>
</tr>
<tr>
<td>has_upper_tail_color: white</td>
<td>55</td>
<td>0</td>
</tr>
<tr>
<td>has_bill_length: same_as_head</td>
<td>78</td>
<td>0</td>
</tr>
<tr>
<td>has_bill_length: shorter_than_head</td>
<td>23</td>
<td>0</td>
</tr>
<tr>
<td>has_forehead_color: black</td>
<td>91</td>
<td>0</td>
</tr>
<tr>
<td>has_forehead_color: red</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>has_nape_color: black</td>
<td>85</td>
<td>0</td>
</tr>
<tr>
<td>has_nape_color: red</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>has_back_pattern: spotted</td>
<td>75</td>
<td>0</td>
</tr>
<tr>
<td>has_back_pattern: striped</td>
<td>18</td>
<td>0</td>
</tr>
<tr>
<td>has_belly_pattern: solid</td>
<td>25</td>
<td>-1</td>
</tr>
<tr>
<td>has_crown_color: black</td>
<td>98</td>
<td>0</td>
</tr>
<tr>
<td>has_crown_color: red</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 12.3: Task of identification of birds belonging to an *endangered species of woodpeckers* from real images (denoted as WP). (a) shows sample images of the dataset X. (b) shows the 13 features used for representation of woodpecker images and the w^h^ vector of the target hypothesis. It also lists the average number of times a particular feature is present in the images of a given species.

Each image in this dataset (cf., [Wah+11]) is annotated with 312 binary attributes, for example,

\[\text{has\_forehead\_color:black, or has\_bill\_length:same\_as\_head,}\]

through workers on the *Mechanical Turk* platform. The features can take values \{+1, -1, 0\} indicating the presence or absence of an attribute, or uncertainty (when the annotator is not sure or the answer cannot be inferred from the image given). Hence, this gives us an embedding of the data in \(\mathbb{R}^{312}\). To further reduce the dimensionality of the feature space, we pruned the features which are not informative enough for the woodpecker species. We considered all the species of woodpeckers present in the dataset (total of 6), simply computed the average number of times a given species is associated positively with a feature, and then looked for features with maximal variance among the various species. By applying a simple cutoff of 60 on the variance, we picked the top \(d = 13\) features as shown in Figure 12.3(b), also listing the average number of times the feature is associated positively with the three species.
Hypothesis Class $\mathcal{H}$

We considered a simple set of linear hypotheses $h(x) = w^T x$ for $w \in \{+1, 0, -1\}^d$, which place a weight of $\{+1, 0, -1\}$ on any given feature and passing through the origin. The intuition behind these simple hypotheses is to capture the cues that participants could possibly use or learn for different features: ignoring a feature ($0$), using it as a positive signal ($+1$), and using it as a negative signal ($-1$). Another set of simple hypotheses that we explored are conjunctions and disjunctions of these features that can be created by setting the appropriate offset factor $b_h$ [Ant+92]. Assuming that participants focus only on a small set of features, we considered sparse hypotheses with non-zero weight on only a small set of features. To obtain the target hypothesis, we enumerated all possible hypotheses that have non-zero weight for at most three features. We then picked as $h^*$ the hypothesis with minimal error on $X$ (shown in Figure 12.3(b)). Again, we pruned the few examples in $X$ which disagreed with $h^*$ to ensure realizability. As hypothesis class $\mathcal{H}$, we considered all hypotheses with a non-zero weight for at most two features along with the target $h^*$, resulting in a hypothesis class of size 339.

Teaching the Rest of the Crowd

Given this embedding and hypothesis class, the teacher then uses the same approach as in the two previous datasets to teach the rest of the crowd. Importantly, this embedding is not required for test images.

12.4 Experimental Results on Simulated Learners

We start with simulated learners as participants and report results only on the VW dataset here for brevity. The simulations allow us to control the problem (parameters of the learner, the size of the hypothesis space, etc.), and hence gain more insight into the teaching process. Additionally, we can observe how robust our teaching algorithm is against misspecified parameters.
12.4. Experimental Results on Simulated Learners

Figure 12.4: (a) compares the algorithms’ teaching performance in terms of simulated learners’ test error (VW task). (b) shows the robustness of STRICT w.r.t. unknown $\alpha$ parameters of the learners. Thus, a noise-tolerant teacher (i.e., $\alpha < \infty$) performs much better than noise-free teaching, even with misspecified $\alpha$. (c) shows how the difficulty of STRICT’s examples naturally increase during teaching.

12.4.1 Test Error

We simulated 100 learners with varying $\alpha$ parameters chosen randomly from the set $\{2, 3, 4\}$ and different initial hypotheses of the learners, sampled from $\mathcal{H}$. We varied the experimental setting by changing the size of the hypothesis space and the $\alpha$ value used by STRICT. Figure 12.4(a) reports results with $\alpha = 2$ for STRICT and the size of hypothesis class as 96 (i.e., 12 hypotheses per each of the eight clusters, described in Section 12.1 for the VW dataset).
12.4.2 Robustness against Learner’s Mismatched Parameters

In real-world annotation tasks, the learner’s $\alpha$ parameter is not known. In this experiment, we vary the $\alpha$ values used by the teaching algorithm STRICT against three learners with values of $\alpha = 1, 2$ and 3. Figure 12.4(b) shows that a conservative teacher using $\alpha$ bounded in the range 1 to 5 performs as good as the one knowing the true $\alpha$ value.

12.4.3 On the Difficulty Level of Teaching

Figure 12.4(c) shows the difficulty of examples picked by different algorithms during the process of teaching, where difficulty is measured in terms of expected uncertainty (entropy) that a learner would face for the shown example, assuming that the expectation is taken w.r.t. the learner’s current posterior distribution over the hypotheses. SetCover starts with difficult examples assuming that the learner is perfect. STRICT starts with easy examples, followed by more difficult ones, as also illustrated in the experiments in Figure 12.5. Recent results of Basu and Christensen [BC13] show that such curriculum-based learning (where the difficulty level of teaching increases with time) indeed is a useful teaching mechanism. Note that our teaching process inherently incorporates this behavior, without requiring explicit heuristic choices. Also, the transition of SetCover to easier examples is just an artifact as SetCover randomly starts selecting examples once it (incorrectly) infers that the learner has adopted the target hypothesis. The difficulty can be easily seen when comparing the examples picked by SetCover and STRICT in Figure 12.5.

12.5 Experimental Results on Human Learners

Next, we measure the performance of our algorithms on human learners. We recruited workers from the Mechanical Turk platform to participate in our experiments by posting the tasks on the platform. Participants were split into different control groups, depending on the algorithm and the length of teaching used (each control group corresponds to a point in the plots of Figure 12.6). Figure 11.1 from Chapter 11 provides a high-level overview of how the teaching algorithm interacted with the participant. Teaching is followed by a phase of testing examples without providing feedback, for
12.5. Experimental Results on Human Learners

Figure 12.5: The order of examples picked (top 5) by the teaching algorithms. For the VW and BM tasks, we embed the picked examples in the 2-D feature space in Figures 12.1(b) and 12.2(b), respectively.

which we report the classification error. We have the following number of participants per task: (i) 780 participated (i.e., 60 participants per control group) in the VW task, (ii) 300 participated in the BM task, and (iii) 520 participated in the VW task. The length of the teaching phase was varied as shown in Figure 12.6. The test phase was set to 10 examples for the VW and BM tasks, and 16 examples for the WP task. The participants were given a fixed payment for participation and completion; additionally, a bonus payment was reserved for the top 10% performing participants within each control group.

12.5.1 Generating the Teaching Sequence

We generate sequences of teaching examples for STRICT, as well as RANDOM and SetCover. We used the feature spaces $\mathcal{X}$ and hypothesis spaces $\mathcal{H}$ as explained in...
Section 12.3. We chose $\alpha = 2$ for our algorithm STRICT. To better understand the execution of the algorithms, we illustrate the examples picked by our algorithm as part of teaching, shown in Figure 12.5. We further show these examples in the 2-D embedding for the VW and BM datasets in Figures 12.1(b) and 12.2(b), respectively.
12.5.2 Does Teaching Help?

Considering the participant’s test set classification performance in Figure 12.6, we can consistently see an accuracy improvement as participants classify unseen images. This aligns with the results from simulated learners and shows that teaching is indeed helpful in practice. Furthermore, the improvement is monotonic w.r.t. the length of teaching phase used by STRICT. In order to understand the significance of these results, we carried out Welch’s t-test (t-test of unpaired samples, of possibly unequal variance) [Wel47] comparing the participants who received teaching by STRICT to the control group of participants without any teaching. The hypothesis that STRICT significantly improves the classification accuracy has two-tailed p-values of $p < 0.001$ for VW and WP tasks, and $p = 0.01$ for the BM task.

12.5.3 Does STRICT Outperform Baselines?

Figure 12.6 demonstrates that our algorithm STRICT outperforms both RANDOM and SetCover teaching qualitatively in all studies. We check the significance by performing a paired t-test, by computing the average performance of the participants in a given control group and pairing the control groups with the same length of teaching for a given task. For the VW task, STRICT is significantly better than SetCover and RANDOM (at $p = 0.05$ and $p = 0.05$). For WP, STRICT is significantly better than SetCover ($p = 0.002$) whereas comparing with RANDOM, the p-value is $p = 0.07$.

12.6 Related Work

In the following, we review relevant empirical work on teaching the participants in crowdsourcing platforms.

12.6.1 Empirical Work on Crowd Teaching

Basu and Christensen [BC13] study a similar problem of teaching participants to classify images. The authors empirically investigate a variety of heuristic teaching policies on a set of human subjects for a synthetically generated data set. One of their empirical
findings is that the curriculum-based learning, where the difficulty level of teaching increases with time, indeed is a useful and competitive teaching mechanism. Interesting, as shown by our experimental results in Figure 12.4(c), our teaching process inherently incorporates this behavior, without requiring explicit heuristic choices. Lindsey et al. [Lin+13] propose a method for evaluating and optimizing over parametrized policies (e.g., with different orderings of positive and negative examples). The main goal of their work is to approximate the performance function over the whole policy space without experimenting with every possible policy.

Some contemporary work has studied a similar problem of teaching the crowd. Gadiraju, Fetahu, and Kawase [GFK15] compared implicit training (providing feedback to the participants when they provide erroneous responses) and explicit training (where participants are required to go through a training phase before they attempt to work) against the default set up of no training. Their findings based on experiments with human subjects on real-world crowdsourcing platform support that of ours, showing a reduction in the test error. A recent work by Johns, Mac Aodha, and Brostow [JMAB15] takes a step further by designing an interactive teaching policy whereby the responses of the participants are taken into account to personalize the picked examples. (cf., Section 11.6 for a discussion on non-interactive vs. interactive teaching). Doroudi et al. [Dor+16] study the problem of training crowd workers for complex crowdsourcing tasks by comparing different training techniques in the context of web search related tasks. Furthermore, they study this problem in the absence of domain expertise, whereby the ground truth data is not available to the teacher. To this end, they explore the feasibility of workers validating the work of their peers, showing this to be an effective methodology in practice.

However, we note that none of these approaches offer theoretical performance guarantees of the kind provided by our algorithmic approach.

12.7 Summary

Our extensive experiments on simulated learners as well as on three real annotation tasks on the Mechanical Turk platform demonstrated the effectiveness of our teaching approach. Our experimental results showed that our algorithm STRICT is robust against the misspecified parameters of the learners (which are not known in real-world
settings). Furthermore, our experiments illustrated that STRICT naturally incorporates the curriculum-based learning behavior, without requiring explicit heuristic choices as is often done in the literature. More generally, our approach goes beyond solving the problem of teaching participants in crowdsourcing services. With the recent growth of online education and tutoring systems (e.g., Coursera [Coul]), algorithms such as STRICT can be envisioned to aid in supporting data-driven online education [Wel+12; DGW13].
Part V

Summary and Outlook
This dissertation set out to investigate the power of combining learning and incentives in improving the effectiveness of crowd-powered systems. We explored research questions encompassing three facets of the interplay between learning and incentives, namely: (i) Learning about Incentives, (ii) Incentives for Learning, and (iii) Learning as an Incentive.

13.1 Summary

Below, we briefly summarize the key results presented in this dissertation.

13.1.1 Learning about Incentives

In Part II, we explored the role of machine learning techniques in learning and designing optimized incentives. Using the approach of regret minimization in online learning, we developed an online pricing mechanism, BP-UCB, for offering monetary incentives when operating under strict constraints on the allocated budget. We proved no-regret guarantees of the mechanism and other desirable properties including budget feasibility and truthfulness. Our mechanism BP-UCB is based on the posted-price model and is easy to deploy in real-world settings instead of, for instance, the com-
Chapter 13. Conclusions

monly studied mechanisms based on the bidding model. Furthermore, our experimental evaluation on real data from the Mechanical Turk platform showed that BP-UCB outperforms the state of the art mechanisms by over 180% increase in utility.

Our pricing mechanism BP-UCB is the basis of the dynamic pricing based incentives system that we developed in this dissertation to tackle the imbalance problem in bike sharing systems (BSSs). Our experiments on a BSS simulator illustrated that, with just over 20% of BSS users starting to participate in our incentives system, the proposed system already outperforms (i.e., provides a higher service level than) the existing approaches to redistribution based only on trucks. The service level metric continuous to improve as the participation level goes up. We deployed our incentives system in the city of Mainz, Germany. To our knowledge, this is the first dynamic incentives system for bike redistribution ever deployed in a real-world bike sharing system.

13.1.2 Incentives for Learning

In Part III, we studied the role of designing incentives in crowd-powered systems for data collection—for instance, to obtain labeled data for improving the performance of machine learning systems or incentivizing users to solve complex sensing tasks. We designed a novel privacy-aware truthful mechanism, SeqTGreedy, for information acquisition from strategic agents, suitable for a large class of sensing applications. Our approach is based on a key insight that privacy tradeoffs could be formulated as an adaptive submodular optimization problem. We proved strong theoretical guarantees of our mechanism including near-optimal utility, budget feasibility, and truthfulness. Our mechanism is general and of independent interest.

We evaluated our mechanism in a case study of a community sensing application for air quality monitoring, motivated by the OpenSense project. As part of this case study, we performed a survey to understand the feasibility of such applications regarding the willingness of users to participate—out of 650 users who took the survey, 75% responded positively. In this survey, we also collected users’ bids to participate in such applications, and we used this data for our experimental evaluation. In our experiments, in order to achieve a desired level of utility, our approach achieved up to 30% reduction in cost compared to the state of the art mechanism. Furthermore, our experimental results provided us insights into the loss in utility our mechanism incurs for enforcing the properties of privacy and truthfulness.
13.2 Concluding Remarks

13.1.3 Learning as an Incentive

Part IV explored the role of learning as an incentive, motivated by applications to educating volunteers in citizen science projects like *eBird*. We studied the problem of selecting training examples for teaching classification rules to participants in order to improve their labeling accuracy. We introduced a new *noise-tolerant* stochastic model of human learners, generalizing the existing *noise-free* models. We then developed our teaching algorithm, STRICT, that selects a sequence of training examples to steer the learners towards the true hypothesis. We proved that our strategy is competitive with the optimal teaching policy. Moreover, for the special case of linear separators, we proved that an exponential reduction in error probability could be achieved.

Motivated by the applications to biodiversity monitoring via citizen science, we performed experiments on three real image annotation tasks of classifying animal species. Our experimental results on simulated learners as well as on participants from a crowdsourcing platform demonstrated the effectiveness of our teaching approach. Our results showed that STRICT naturally incorporates the curriculum-based learning behavior (where the difficulty level of teaching increases with time), without requiring explicit heuristic choices as are often made in the literature. Our experiments on human subjects demonstrated that the existing approaches based on *noise-free* models of learners are brittle in practice, performing poorly compared to our proposed approach.

13.2 Concluding Remarks

The novel techniques presented in this dissertation explore several fundamental challenges that are prevalent in the ongoing technological revolution, including privacy concerns, learning/modeling user preferences, and robustness of machine learning techniques against the strategic behavior of users. The techniques we developed are both theoretically well-founded and practically applicable, with the potential to dramatically increase the effectiveness of crowd-powered systems in industry and science. A central theme of our research is empowering users so that they are actively engaged in contributing to the system—our results point to a future of self-sustainable and intelligent crowd-powered systems.
Future Research Directions

Our results presented in this dissertation inspire several exciting directions for future research; we briefly outline a few of them below.

14.1 Incentivizing Exploration for Maximizing Welfare

One of the key characteristics of the sharing economy and community-based online services such as Airbnb is the review-based reputation system. The users consuming these services often favor purchasing/experiencing the highly-rated items/options. This creates a vicious circle, as under-reviewed items are often neglected and thus unlikely to be evaluated. For instance, our recent study in Hirnschall et al. [Hir+17] shows that about 40% of the apartments on Airbnb have only two reviews or less. Similar unbalanced distribution of reviews has been observed in other online marketplaces, for instances, reviews for books on Barnes & Noble or Amazon [CM06]. This vicious circle creates multiple challenges for the “company” (e.g., Airbnb), for the “sellers” providing services (e.g., hosts on Airbnb), and for the “consumers” (e.g., guests on Airbnb). For instance, new “sellers” face significant barriers to entering an existing online marketplace [Res+00], a problem that also encourages the creation of fraudulent customer reviews [JL07].
Chapter 14. Future Research Directions

The fundamental research question is how to incentivize self-interested users to explore new items to maximize the overall welfare, who otherwise prefer to undertake myopic decisions. There has been some recent interest in tackling this question: new mathematical models formalizing this problem have been proposed based on Bayesian multi-armed bandits [Fra+14]; a few practical solutions have also been proposed, for instance, Xue et al. [Xue+16] tackled the problem of data sparsity in citizen science projects by using similar ideas. In our recent work [STK16a; Hir+17], we introduced a novel framework of encoding users’ preferences across different choices via hemimetrics and studied this in the context of applications for gathering more valuable reviews on Yelp and Airbnb.

These approaches can be seen as an initial step in tackling this challenge of incentivizing exploration. We need new mathematical models of the user preferences that quantify the tradeoffs they make when switching choices in these applications. Furthermore, these models and techniques crucially depend on the type of incentives that can be offered, ranging from monetary incentives (e.g., based on discount coupons) and social incentives (such as badges) to merely randomized A/B-testing style exploration experiments. In order to gain deeper insight into the problem, we believe that a good next step is to conduct behavioral experiments and survey studies in such marketplaces to better understand user preferences. With the exponentially growing impact of the sharing economy paradigm, we believe that this line of research poses a great potential for a real-world impact.

14.2 Leveraging Online Connectivity and Social Incentives

The proliferation of online social networks in the last decade has opened up numerous ways for people to interact and exchange information with each other. Social connectivity and visible commonality of friends give users a sense of trust when they interact with others in a network, both online and in real life. We believe that there is a huge potential in leveraging network connectivity, specifically the trust and social incentives that arise from it.

Let us illustrate this idea with one concrete example in the context of the sharing economy and community-based online services. Our recent survey study [Hir+17] on
the rental marketplace Airbnb suggests that trust and safety are major factors in users not picking under-reviewed apartments in such online marketplaces. On the other hand, the world-scale studies of connectivity on Facebook network by Backstrom et al. [Bac+12] have shown that an average number of intermediate nodes/friends between two random individuals is less than 4. Our premise is that if a user searching for an apartment on services like Airbnb can see such a chain of connectivity to the host of an under-reviewed apartment, it would alleviate her concerns of trust and safety.

We envision crowd-powered systems built with online social networks like Facebook as the backbone, i.e., users in these systems are associated with their social profile and implicitly connected with others via this network. For instance, imagine adding this social backbone to crowdsourcing services such as Mechanical Turk, Upwork, and citizen science projects. It can help in automatically inferring the users’ skills/expertise from their social profile, which in turn can lead to better matching of tasks. Furthermore, this opens up exciting new opportunities for building teams to tackle complex heterogeneous tasks by leveraging social connections—the DARPA Red Balloon Challenge [Dar] is the perfect illustration of this fact.

Recently, there has been an increasing interest in building applications that can leverage this social connectivity and exploit social incentives to get users to help their peers/friends. For instance, Bernstein et al. [Ber+10] introduced the concept of friendsourcing, a form of crowdsourcing aimed at collecting accurate information available from social networks. Zhang et al. [Zha+12] considered the problem of task routing in social networks for prediction tasks, designing routing-based scoring rules and studying their truthfulness/efficiency in the equilibrium. Our recent work [Sin+15b] suggests that an increasing number of people resort to online social networks to post their queries—from asking simple informational questions to seeking more specific expertise (e.g., a part-time tutor to learn a new language). The current design of online social networks like Facebook does not provide the required technology to initiate these kinds of queries or reroute them in the network. People often resort to posting queries on their Timelines or special-purpose Groups and find themselves unable to get the desired information. This calls for new design principles whereby we can empower users to leverage the underlying networks to search for the information or expertise they desire. Furthermore, there is a need to build mathematical models of social incentives and efficient algorithms for information gathering in networks. A fundamental challenge that needs to be tackled is ensuring user privacy and minimizing the dis-
ruptions they encounter while we seek to build crowd-powered systems on this social backbone.

14.3 Learning with and from People

Consider the well-studied machine learning algorithms and optimization techniques, such as active learning by querying labels from experts or information gathering by selecting a set of sensors under budget constraints. These algorithmic techniques are typically designed to interact with non-strategic components such as programmable sensors or domain experts. As people are becoming an integral part of the AI and machine learning systems, this raises new issues and technical challenges including concerns about privacy and fairness, as well as the robustness of the deployed algorithms which typically do not account for the intricacies of human behavior (e.g., strategic users acting in their self-interest). Below, we list a few interrelated directions of research.

The Role of Behavioral Modeling in Data Science

Human-generated data is one of the fastest growing datasets in this era of big data—this growth provides rich opportunities for mining actionable intelligence from this data, forming the basis of data science. We believe that a key to the success of data science projects is to model and better understand the source of this data itself, i.e., people and their activities that generate this data. A big challenge—and also a great opportunity—is building robust and rich mathematical models of human behavior. In our earlier work, we demonstrated these ideas by showing improved system performance after accounting for human behavior in the context of applications to social networks [SW09] and web search [Che+12; Sin+14a] while working with petabytes scale data. Inspired by the techniques developed in this dissertation, an interesting direction to explore would be designing active interventions whereby the system can collect more informative data actively from the users.
14.3. Learning with and from People

Accounting for Constraints of Privacy and Fairness

Another major challenge is that of protecting user privacy as we design systems that interact with users’ sensitive personal data, such as medical history or financial statements. One of the main research questions is designing privacy-preserving methodologies that are easy to interpret and understandable by users, an important step towards giving them the ultimate control of their data. As a first step towards this goal, in our recent work Singla et al. [Sin+14c], we introduced a new approach to privacy that we refer to as stochastic privacy. Stochastic privacy depends critically on harnessing inference and decision making to make choices about data collection within the constraints of a guaranteed privacy risk. Another big challenge that has not been explored much in literature is that of fairness, ensuring that two users with similar demographics or features are treated similarly by the system (e.g., regarding recommendations, payments, or the output of a resource allocation policy). It turns out that many simple algorithmic techniques, including those developed in this dissertation, fail to satisfy certain characteristics of fairness. This requires a rethinking of the current statistical and optimization techniques. There is a need to develop new mathematical frameworks that allow us to enforce privacy and fairness by specifying them as constraints, just like constraints on the budget or computational resources.

Information Acquisition from Strategic Agents

In this dissertation, we studied the problem of information acquisition from people in the context of community sensing applications. However, we considered a restricted setting where the users’ strategic behavior is limited only to misreporting their private costs. In real-world systems, it is natural to expect more intricate human behaviors: for instance, a user may produce low-quality data unless incentivized to exert effort. Given that the ground truth data is not available (in which case this information acquisition would be redundant), this poses major challenges to acquiring data from these strategic agents in an incentive-compatible manner. An interesting direction of research for tackling this class of problems is to combine peer-prediction mechanisms (cf., [WP12; WP13; RFJ16; Shn+16]), which are designed for truthful information elicitation, with machine learning algorithms. For instance, a concrete problem to look at is that of designing an active learning algorithm where the cost of querying an agent is defined by the payments dictated by peer-prediction mechanisms. Another
interrelated problem is to design new discrete optimization techniques (e.g., for maximizing submodular set functions) under budget constraints on the payments defined by peer-prediction mechanisms.

**Learning to Aggregate Recommendations from Learners**

As we mentioned above, many machine learning techniques are designed to interact with *domain experts*, for instance, active learning algorithms query labels from a (possibly noisy) domain expert. In the emerging crowd-powered systems, these experts are getting replaced by inexpert participants who could themselves be learning (e.g., volunteers in citizen science projects). In this dissertation, we also explored this idea and tackled a related problem of teaching these participants to improve their classification accuracy. However, this paradigm shift creates new challenges for many learning and optimization frameworks. As a concrete problem, let us consider the classical online learning framework of *learning using expert advice with bandit feedback* (cf., [CBL06]), where an online algorithm seeks prediction recommendations from a set of available experts. Replacing these experts with learning entities leads to a small but challenging twist in this formulation. Intuitively, it adds the following complexity in the classic tradeoff between exploration and exploitation: The algorithm needs to explore not just to identify the best expert, but also to ensure that the best expert gets to learn so that it can receive better rewards and be identified. In our recent work [SHK17], we show that this small twist substantially increases the learning complexity of the problem: the well-studied online learning algorithms such as *EXP3* (cf., [BCB12]) fail to provide any guarantees for this modified problem. It is an open research question to develop online learning algorithms for this new setting with no-regret guarantees and to prove lower bounds on the minimax regret.
Part VI

Appendices
A

Proofs of Part II

A.1 Components Contributing to the Regret: Proof of Lemma 4.1

We begin by expressing the expected utility of a mechanism in terms of \( N^T_i \) in Lemma A.1. We use \( I^t_i \) as indicator variable indicating that price \( p_i \) was offered at time \( t \).

**Lemma A.1.** The expected utility of the mechanism \( M \) is given by 
\[
U(M, B) = \sum_{i=1}^{K} \mathbb{E}[N^T_i] \cdot F_i
\]

**Proof.** From definition of \( S^t_i \), we have
\[
U(M, B) = \mathbb{E}_T \left[ \mathbb{E} \left[ \sum_{t=1}^{T} S^t_i \mid T \right] \right] = \sum_{i=1}^{K} \mathbb{E}_T \left[ \mathbb{E} \left[ \sum_{t=1}^{T} I^t_i \cdot S^t_i \mid T \right] \right]
\]
\[
= \sum_{i=1}^{K} \mathbb{E}_T \left[ \mathbb{E} \left[ \sum_{t=1}^{T} I^t_i \mid T \right] \cdot \mathbb{E} \left[ S^t_i \mid T \right] \right] \tag{1}
\]
\[
= \sum_{i=1}^{K} \mathbb{E}_T \left[ \mathbb{E} \left[ \sum_{t=1}^{T} I^t_i \mid T \right] \right] \cdot F_i = \sum_{i=1}^{K} \mathbb{E}_T \left[ \mathbb{E} \left[ N^T_i \mid T \right] \right] \cdot F_i \tag{2}
\]
\[
= \sum_{i=1}^{K} \mathbb{E} \left[ N^T_i \right] \cdot F_i
\]

Unlike UCB1, \( T \) is a random variable here. Step 1 uses the fact that \( S^t_i \) only depends on the order of bids and is therefore independent of the mechanism (i.e., of \( I^t_i \) and \( T \)). Step 2 follows from definitions of \( N^T_i \) and \( F_i \). \( \square \)
Next, in Lemma A.2, we provide a lower bound on the expected timesteps \( T \) in the execution of a mechanism, unlike in standard UCB1 where \( T \) is fixed. This ensures that the mechanism’s regret coming from the component “Wasted budget through overpayment” in Lemma 4.1 is bounded.

**Lemma A.2.** The expected number of timesteps \( T \) in the execution of the mechanism has lower bound as follows:

\[
\mathbb{E}[T] > \frac{B}{p^* \cdot F^*} - \frac{c_{\min}}{p^* \cdot F^*} - \sum_{i:p_i > p^*} \frac{\mathbb{E}[N_i^T] \cdot (p_i \cdot F_i - p^* \cdot F^*)}{p^* \cdot F^*}
\]

**Proof.** The algorithm terminates when either \( i) \ t > N \) or \( ii) B^t < c_{\min} \). In \( i) \), \( T = N \) and hence bounds in the equation hold trivially since \( N = \frac{B}{p^* \cdot F^*} \). We will prove the bounds for \( ii) \) below by bounding the sum of the prices offered and accepted by workers \( \sum_{t=1}^{T} S_t^i \cdot p^i \).

\[
B - c_{\min} < \mathbb{E}_T \left[ \mathbb{E} \left[ \sum_{t=1}^{T} S_t^i \cdot p^i \mid T \right] \right] = \sum_{i=1}^{K} \mathbb{E}[N_i^T] \cdot F_i \cdot p_i
\]

\[
\mathbb{E}[T] \cdot F^* \cdot p^* > B - c_{\min} - \sum_{i:p_i < p^*} \mathbb{E}[N_i^T] \cdot (F_i \cdot p_i - F^* \cdot p^*)
\]

\[
- \sum_{i:p_i > p^*} \mathbb{E}[N_i^T] \cdot (F_i \cdot p_i - F^* \cdot p^*)
\]

Step 1 follows by using the same arguments as in Lemma A.1. Step 2 replaces \( \sum_{i=1}^{K} N_i^T \) by \( T \) and we get the desired bounds by using the fact that \( (F_i \cdot p_i - F^* \cdot p^*) < 0 \) for \( p_i < p^* \). \( \square \)

We now prove Lemma 4.1 by using the above results.

**Proof of Lemma 4.1.** Consider an alternate mechanism \( M' \) which has access to arm corresponding to price \( p^* \) instead of \( p' \). That is, whenever our mechanism offers price \( p' \), we can consider that it is offering \( p^* \). We first analyze the regret of \( M' \) below.

\[
R_{\text{M'}}(B) = U(p^*, B) - U(M', B) = \frac{B}{p^*} - \sum_{i=1}^{K} \mathbb{E}[N_i^T] \cdot F_i
\]

\[
= \frac{B}{p^*} + \sum_{i=1}^{K} \mathbb{E}[N_i^T] \cdot (F^* - F_i) - \sum_{i=1}^{K} \mathbb{E}[N_i^T] \cdot F^*
\]
A.2. Regret Bounds for BP-DGREEDY

Proof of Theorem 4.1

\[ R_M = \frac{B}{p^*} - \mathbb{E}[T] \cdot F^* + \sum_{i:p_i < p^*} \mathbb{E}[N_i^T] \cdot (F^* - F_i) \]
\[ + \sum_{i:p_i > p^*} \mathbb{E}[N_i^T] \cdot (F^* - F_i) \]
\[ \leq \frac{B}{p^*} - \mathbb{E}[T] \cdot F^* + \sum_{i:p_i < p^*} \mathbb{E}[N_i^T] \cdot (F^* - F_i) \]
\[ < \frac{c_{\min}}{p^*} + \sum_{i:p_i < p^*} \mathbb{E}[N_i^T] \cdot (F^* - F_i) \]
\[ + \sum_{i:p_i > p^*} \mathbb{E}[N_i^T] \cdot \left( \frac{p_i \cdot F_i - p^* \cdot F^*}{p^*} \right) \]

Step 1 replaces \( \sum_i N_i^T \) by \( T \) and step 2 uses the fact that \( (F^* - F_i) < 0 \) for \( p_i > p^* \). And, step 3 follows from the results of Lemma A.2.

The regret of mechanism \( M \) is given by \( R_M(B) = R_{M'}(B) + R_{p'}(B) \) where \( R_{p'}(B) = \left( \frac{B}{p'} - U(p', B) \right) \). The regret \( R_{p'}(B) \) is bounded in the Lemma A.3 below. Using the value of \( R_{M'} \) from above completes the proof.

\[ \square \]

A.2  Regret Bounds for BP-DGREEDY: Proof of Theorem 4.1

In Lemma A.3, we show that discretization to a power of \((1 + \alpha)\) results in loss of utility by at most a factor of \((1 + \alpha)\), similar to the online auctions as shown in [Blu+03; Bab+12].

**Lemma A.3.** The regret component “discretization” in Lemma 4.1 is upper-bounded by \( R_{p'}(B) \leq \frac{\alpha \cdot B}{p^*} \).

**Proof.** Consider price \( p^h \) given by:

\[ p^h = \inf_{p_i} \{ \forall i \in [1 \ldots K] \text{ s.t. } p_i \geq p^* \} \]

By the design of discretization, \( p^h < (1 + \alpha) \cdot p^* \). Now, consider that price \( p^h \) is offered to every worker instead of \( p' \).

\[ R_{p'}(B) = U(p^*, B) - U(p', B) \leq \frac{B}{p^*} - \frac{B}{p^h} \leq \frac{\alpha \cdot B}{p^*} \]

We use the fact that for \( p^h \geq p^* \), \( U(p^h; B) = \frac{B}{p^h} \). \[ \square \]
Lemma A.4. \( \forall i \text{s.t. } p_i < p', \) expected number of times a suboptimal arm \( i \) is played is upper-bounded by \( \mathbb{E}[N_{i}^T] \leq \frac{8}{\Delta_i^2}. \)

**Proof.** A suboptimal arm \( i \) is picked at time \( t \) when \( V_t^i \geq V_t^j \) where \( V_t^i = \min \left\{ F'_t, \frac{B}{N \cdot p_j} \right\} \). We consider the following cases:

**Case a)** \( p' \leq p^* \) and \( F'_i \leq \frac{B}{N \cdot p'} \):

\[
\frac{F_t^i - F_t^j}{2} + \left( \frac{F_t^i - F_t^j - F_t^j - F_t^i}{2} \right) \geq 0
\]

\[
\left( (F_t^i - F_t^j) - \frac{\Delta_i}{2} \right) + \left( (F_t^j - F_t^i) - \frac{\Delta_i}{2} \right) \geq 0
\] (1)

**Case b)** \( p' \leq p^* \) and \( F'_i > \frac{B}{N \cdot p'} \):

\[
F_t^i \geq \frac{B}{N \cdot p'} \geq F_t^j \implies F_t^i - F_t^j \geq \Delta_i
\] (2)

**Case c)** \( p' > p^* \) and \( F'_i \leq \frac{B}{N \cdot p'} \):

\[
\frac{F_t^i - F_t^j}{2} + \left( \frac{F_t^i - F_t^j - F_t^j - F_t^i}{2} \right) \geq 0
\]

\[
\left( (F_t^i - F_t^j) - \frac{\Delta_i}{2} \right) + \left( (F_t^j - F_t^i) - \frac{\Delta_i}{2} \right) \geq 0
\] (3)

**Case d)** \( p' > p^* \) and \( F'_i > \frac{B}{N \cdot p'} \):

\[
F_t^i \geq \frac{B}{N \cdot p'} + F_t^i - F_t^j \implies F_t^i - F_t^j \geq \Delta_i
\] (4)

Using Chernoff-Hoeffding inequality and the fact that \( \mathbb{P}(A + B \geq 0) \leq \mathbb{P}(A \geq 0) + \mathbb{P}(B \geq 0) \), we bound step 1 and 3 as:

\[
\mathbb{P}\left( \left( F_t^i - F_t^j - \frac{\Delta_i}{2} \right) + \left( F_t^j - F_t^i - \frac{\Delta_i}{2} \right) \geq 0 \right) \leq 2 \cdot e^{-\frac{\Delta_i^2}{2} \cdot t}
\]

For step 2 and 4, we have the following bounds:

\[
\mathbb{P}(F_t^i - F_t^j \geq \Delta_i) \leq e^{-2 \cdot \frac{\Delta_i^2}{2}} \leq 2 \cdot e^{-\frac{\Delta_i^2}{2} \cdot t}
\]
Combining the bounds for above cases, we have:

\[ \mathbb{E}[N^T_i] \leq \sum_{t=1}^{T} e^{-2 \Delta_i^2 \cdot t} \leq \sum_{t=1}^{\infty} 4 \cdot e^{-\Delta_i^2 \cdot t} = \frac{8}{\Delta_i^2} \]

\[ \square \]

**Lemma A.5.** \( \forall i \) s.t. \( p_i > p' \), expected number of times a suboptimal arm \( i \) is played is upper-bounded by \( \mathbb{E}[N^T_i] \leq \frac{1}{2 \Delta_i^2} \).

**Proof.** A suboptimal arm \( i \) is picked at time \( t \) when \( V'_i > V^*_i \). We consider the following cases and the conditions that need to hold for picking arm \( i \).

**Case a)** \( p' \leq p^* \):

\[ F'_i < \frac{B}{N \cdot p_i} = F' - \Delta_i \implies F' - F'_i > \Delta_i \] (1)

**Case b)** \( p' > p^* \):

\[ F'_i < \frac{B}{N \cdot p_i} = \frac{B}{N \cdot p'} - \Delta_i \implies F' - F'_i > \Delta_i \] (2)

Using Chernoff-Hoeffding inequality, step 1 and step 2 are bounded by \( e^{-2 \Delta_i^2 \cdot t} \). Combining the bounds for above cases, we have:

\[ \mathbb{E}[N^T_i] \leq \sum_{t=1}^{T} e^{-2 \Delta_i^2 \cdot t} \leq \sum_{t=1}^{\infty} e^{-2 \Delta_i^2 \cdot t} = \frac{1}{2 \Delta_i^2} \]

\[ \square \]

**Proof of Theorem 4.1.** The proof directly follows from the bounds of \( N_i \) from Lemmas A.4 and A.5. By putting in these bounds in Lemma 4.1 along with the regret of discretization from Lemma A.3, we get the desired results.

\[ \square \]

### A.3 Regret Bounds for BP-UCB: Proof of Theorem 4.2

**Lemma A.6.** \( \forall i \) s.t. \( p_i < p' \), expected number of times a suboptimal arm \( i \) is played is upper-bounded by

\[ \mathbb{E}[N^T_i] \leq \frac{8 \cdot \ln(h/\epsilon_{\min})}{\Delta_i^2} + \frac{\pi^2}{2} + 1. \]
Proof. A suboptimal arm $i$ is picked at time $t$ when $\tilde{V}_i^t \geq \tilde{V}_j^t$ where $\tilde{V}_j^t = \min \left\{ F_j^t + \sqrt{\frac{2 \cdot \ln(t)}{N_j^t}}, \frac{B}{N \cdot p_j} \right\}$. We consider the following cases:

Case a) $p' \leq p^*$ and $F_i^t + \sqrt{\frac{2 \cdot \ln(t)}{N_i^t}} \leq \frac{B}{N \cdot p^*}$:

$$F_i^t + \sqrt{\frac{2 \cdot \ln(t)}{N_i^t}} \geq F_{i'}^t + \sqrt{\frac{2 \cdot \ln(t)}{N_{i'}^t}}$$

$$\left((F_i^t - F_{i'}) - \sqrt{\frac{2 \cdot \ln(t)}{N_i^t}}\right) + \left((F_{i'}^t - F_i^t) - \sqrt{\frac{2 \cdot \ln(t)}{N_{i'}^t}}\right) + \left(2 \cdot \sqrt{\frac{2 \cdot \ln(t)}{N_{i'}^t}} - \Delta_i\right) \geq 0 \quad (1)$$

Case b) $p' \leq p^*$ and $F_i^t + \sqrt{\frac{2 \cdot \ln(t)}{N_i^t}} > \frac{B}{N \cdot p^*}$:

$$F_i^t + \sqrt{\frac{2 \cdot \ln(t)}{N_i^t}} \geq \frac{B}{N \cdot p^*} \geq F'$$

$$\left((F_i^t - F_{i'}) - \sqrt{\frac{2 \cdot \ln(t)}{N_i^t}}\right) + \left(2 \cdot \sqrt{\frac{2 \cdot \ln(t)}{N_{i'}^t}} - \Delta_i\right) \geq 0 \quad (2)$$

Case c) $p' > p^*$ and $F_i^t + \sqrt{\frac{2 \cdot \ln(t)}{N_i^t}} \leq \frac{B}{N \cdot p^*}$:

This case is analogous to (1) in Case (a) with some algebraic manipulations.

Case d) $p' > p^*$ and $F_i^t + \sqrt{\frac{2 \cdot \ln(t)}{N_i^t}} > \frac{B}{N \cdot p^*}$:

This case is analogous to (2) in Case (b) with some algebraic manipulations.

Using the Chernoff-Hoeffding inequality, step 1 and step 2 are bounded by $2 \cdot t^{-4}$ and $t^{-4}$ respectively. Once suboptimal arm $i$ has been played sufficient number of times, given by $N_i^t \geq \lceil \frac{8 \cdot \ln(B/c_{\min})}{\Delta_i^2} \rceil$, we have:

$$\mathbb{P} \left( 2 \cdot \sqrt{\frac{2 \cdot \ln(t)}{N_i^t}} - \Delta_i > 0 \right) = 0$$

Combining these together, we have the following:

$$\mathbb{E} [N_i^T] \leq \left\lceil \frac{8 \cdot \ln(B/c_{\min})}{\Delta_i^2} \right\rceil + \sum_{t=1}^{T} 3 \cdot t^{-4}$$
A.3. Regret Bounds for BP-UCB

Proof of Theorem 4.2

\[ \leq \left\lceil \frac{8 \cdot \ln(B/\epsilon_{\text{min}})}{\Delta_i^2} \right\rceil + \sum_{t=1}^{\infty} 3 \cdot t^{-4} < \frac{8 \cdot \ln(B/\epsilon_{\text{min}})}{\Delta_i^2} + \frac{\pi^2}{2} + 1 \]

Lemma A.7. \( \forall i \) s.t. \( p_i > p' \), expected number of times a sub-optimal arm \( i \) is played is upper-bounded by \( \mathbb{E}[N_i^T] \leq \frac{\pi^2}{6} \).

Proof. A suboptimal arm \( i \) is picked at time \( t \) when \( \tilde{V}_t^i > \tilde{V}_t^{i'} \). Irrespective of the relation between \( p' \) and \( p^* \), the following must hold true:

\[ F_i + \sqrt{\frac{2 \cdot \ln(t)}{N_i^{i'}}} < \frac{B}{N \cdot p_i} < F' \implies F' - F_i > \sqrt{\frac{2 \cdot \ln(t)}{N_i^{i'}}} \]

Using Chernoff-Hoeffding inequality, above case is bounded by \( t^{-4} \). We have:

\[ \mathbb{E}[N_i^T] \leq \sum_{t=1}^{T} t^{-4} \leq \sum_{t=1}^{\infty} t^{-4} < \frac{\pi^2}{6} \]

Proof of Theorem 4.2. The proof follows by using exactly the same arguments as in Theorem 4.1.
Proofs of Part III

B.1 Truthfulness of the Mechanism: Proof of Theorem 8.1

Let $\mathcal{S}$ denote the set of participants allocated by $\pi_{\mathcal{M}}$ along with making observations $y_s$. We use $Z_{W,\mathcal{S}} = [y^1, y^2 \ldots y^r \ldots y^Z]$, where $Z = |Z_{W,\mathcal{S}}|$, to denote the set of possible realizations of $Y_W = y_W \subseteq W \times O$ consistent with $y_s$. In Lemma B.3, we first prove the truthfulness of the payment $\theta^d_s(y^r)$ made for each of these possible realizations $y^r \in Z_{W,\mathcal{S}}$ (also denoted as $\theta^d_s(y^r)$). To prove Lemma B.3, we first show allocation rule is monotone (Lemma B.1) and allocated users are paid threshold payments (Lemma B.2).

**Lemma B.1.** For a given $y_W$, allocation policy of the mechanism is monotone i.e. $\forall i \in [n]$ and for every $b_{-i}$, if $b_i' \leq b_i$ then $i \in \pi(b_i, b_{-i})$ implies $i \in \pi(b_i', b_{-i})$

**Proof.** The monotonicity of the greedy scheme is easy to see: By lowering her bid, any allocated participant would only increase their marginal gain per unit cost and thus jump ahead in the sorting order considered by the allocation policy.

**Lemma B.2.** Payment $\theta^d_s$ for a given $y_W$ is a threshold payment, i.e., payment to each winning bidder is $\inf\{b_i' : i \notin \pi(b_i', b_{-i})\}$

**Proof.** The threshold payment for participant $s = i$ is given by $\theta^d_i = \max_{j \in [k'+1]}(\theta^d_{i(j)})$ where $\theta^d_{i(j)} = \min(b_{i(j)}, \rho_{i(j)})$ as the bid that $i$ can declare to replace $j$ in $S'$. We have
Appendix B. Proofs of Part III

\[ b_{i(j)} = \frac{\Delta_{i(j)} \cdot b_i}{\Delta_i} \] and \[ \rho_{i(j)} = \frac{b_i}{r} \cdot \frac{\Delta_{i(j)}}{\sum_{j' \in \Gamma_i} \Delta_{i(j')} + \Delta_{i(j)}}. \] Let us consider \( r \) to be the index for which \( \theta_i = \min(b_{i(r)}, \rho_{i(r)}) \). Declaring a bid of \( \min(b_{i(r)}, \rho_{i(r)}) \) ensures that \( s \) would definitely get allocated at position \( r \) in the alternate run of the policy. Let us consider the following four cases:

**Case 1:** \( b_{i(r)} \leq \rho_{i(r)} \) & \( b_{i(r)} = \max_j b_{i(j)} \)

Reporting a bid higher than \( b_{i(r)} \) places the \( i \) after the unallocated user \( k' + 1 \) in the alternate run of the mechanism, thereby \( i \) would not be allocated.

**Case 2:** \( b_{i(r)} \leq \rho_{i(r)} \) & \( b_{i(r)} < \max_j b_{i(j)} \)

Consider some \( j \) for which \( b_{i(r)} < b_{i(j)} \). Because of the maximal condition for \( r \), it must be the case that \( \rho_{i(j)} \leq b_{i(r)} \leq b_{i(j)} \). Thus, declaring a bid higher than \( b_{i(r)} \) would violate the proportional share allocation condition and hence \( i \) would not be allocated. For some other \( j \) for which \( b_{i(r)} \geq b_{i(j)} \), declaring a bid higher than \( b_{i(r)} \) would put \( i \) after \( j \) and hence \( i \) would not be allocated at considered position \( j \).

**Case 3:** \( \rho_{i(r)} \leq b_{i(r)} \) & \( \rho_{i(r)} = \max_j \rho_{i(j)} \)

Reporting a bid higher than \( \rho_{i(r)} \) violates the proportional share allocation condition at each of the indices in \( j \in [k' + 1] \), hence \( i \) would not be allocated.

**Case 4:** \( \rho_{i(r)} \leq b_{i(r)} \) & \( \rho_{i(r)} < \max_j \rho_{i(j)} \)

Consider some \( j \) for which \( \rho_{i(r)} < \rho_{i(j)} \). Because of the maximal condition for \( r \), it must be the case that \( b_{i(j)} \leq \rho_{i(r)} \leq \rho_{i(j)} \). Thus, declaring a bid higher than \( \rho_{i(r)} \) would put \( i \) after \( j \) and hence \( i \) would not be allocated. For any other \( j \) for which \( \rho_{i(r)} \geq \rho_{i(j)} \), declaring a bid higher than \( b_{i(r)} \) would violate the proportional share allocation condition and hence \( i \) would not be allocated at considered position \( j \).

The analysis of above four cases completes the proof. \( \square \)

**Lemma B.3.** Payment \( \theta_s^d \) for a given \( y_W \) is truthful.

**Proof.** To prove this, we use the well-known characterization of [Mye81]. For the case of deterministic settings in single parameter domains, a mechanism is truthful if the allocation rule is monotone and the allocated agents are paid threshold payments. \( \square \)

**Proof of Theorem 8.1.** The final payment made to participant \( s \) is given by

\[ \theta_s = \sum_{y' \in Z_{W,s}} P(Y_W = y' | y_s) \cdot \theta_s^{d,r}. \]
From Lemma B.3, each of the payments $\theta_s^{d,r}$ are truthful, i.e., the profit of a user cannot be increased by deviating from their true cost. Taking a linear combination of these payments ensures truthful payment as well.

## B.2 Individually Rationality: Proof of Theorem 8.2

In Lemma B.4, we first prove the individual rationality of the payment $\theta_s^d(y^r)$ made for each of these possible realizations $y^r \in Z_{W,S}$ (also denoted as $\theta_s^{d,r}$).

**Lemma B.4.** Payment $\theta_s^d$ for a given $y_W$ is individually rational i.e. $\theta_s^d \geq b_s$

**Proof.** Consider the bid that $i$ can declare to be allocated at position $j = i$ (i.e. back at its original position) in the alternate run of the mechanism. $\theta_{i(i)}^d = \min(b_{i(i)}, \rho_{i(i)})$. We will show that $b_i \leq \theta_{i(i)}^d$.

**Showing $b_{i(i)} \geq b_i$**

$$b_{i(i)} = \frac{\Delta_{i(i)} \cdot b_j}{\Delta_j} \geq \frac{\Delta_i \cdot b_j}{\Delta_i} = b_i$$

In step 1, the second equality holds from the fact that the first $i - 1$ allocated elements in both runs of the policies are the same and hence $\Delta_{i(i)} = \Delta_i$ and $\Delta_j' = \Delta_j$. In step 2, the first inequality holds from the fact that $\frac{b_j}{\Delta_j'} \geq \frac{b_i}{\Delta_i}$ since $i$ was allocated in the original run of the policy after $i - 1$, instead of user $j$.

**Showing $\rho_{i(i)} \geq b_i$**

$$\rho_{i(i)} = \frac{B}{a} \cdot \frac{\Delta_{i(i)}}{\sum_{s' \in [i-1]} \Delta_{s'} + \Delta_{i(i)}} = \frac{B}{a} \cdot \frac{\Delta_i}{\sum_{s \in [i-1]} \Delta_s + \Delta_i} \geq b_i$$

In step 3, the first equality holds from the fact that the first $i - 1$ allocated elements in both the runs of the policies are same. The second inequality follows from the
Appendix B. Proofs of Part III

proportional share criteria used to decide the allocation of $i$ after $i - 1$ users were allocated already.

Now, we have $b_i \leq \theta_{i(i)} \leq \max_{j \in [k'+1]} (\theta_{i(j)}) = \theta_i$

\boxed{	ext{Proof of Theorem 8.2.}} The final payment made to participant $s$ is given by

$$\theta_s = \sum_{y' \in \mathcal{Z}_{W,S}} P(Y_W = y'|y_S) \cdot \theta_{s,r}^{d,r}.$$ 

From Lemma B.4, each of the payment $\theta_{s,r}^{d,r} \geq b_s$. Taking a linear combination of these payments ensures individual rationality in expectation as well.

B.3 Budget Feasibility: Proof of Theorem 8.3

The theorem rests on the following Lemma B.5 which upper bounds the payments made to each participant by $\alpha \geq 1$ times their marginal contribution to the total utility of the final set of participants.

\textbf{Lemma B.5.} When full budget $B$ is used by mechanism, the maximum raise in bid $b'_s$ that a participant $s$ can make, keeping the bids of others same, to still get selected by mechanism is upper bounded by $\alpha \cdot \Delta_s \sum_{s \in S} \Delta_s \cdot B$ where $\alpha \leq 2$.

\textbf{Proof.} Consider any random realization $Y_W = y_W$. Let $S$ be the set of participants selected by policy along with making observations $y_S$. Let us renumber the users in which they were allocated by mechanism $S = \{1, 2, \ldots, i, i(= s), \ldots, k\}$ and let’s analyze the upper bound on the threshold payment for participant $s = i$. Irrespective of the payment scheme used, we consider how much raised bid participant $i$ ($b'_i$ raised from $b_i$) can declare to still selected by the mechanism, keeping the bids of other users ($b_{-i}$) same. We use $B = (b_i, b_{-i})$ to denote original bids and $B' = (b'_i, b_{-i})$ to denote modified bids. Consider running policy on alternate bids $B'$ and let $S' = \{1, 2, \ldots, j-1, i(= s), \ldots, k'\}$ be the allocated set (users again renumbered based on order of allocation). For distinction, we use $\Delta$ and $\Delta'$ to denote the marginal contributions of the users in the above two different runs of the policy. Let $T'$ denote the subset of participants from $S'$ which were allocated just before $s$ was allocated at position $j$. Let us consider following two cases:
B.3. Budget Feasibility

Proof of Theorem 8.3

Case 1: $S \setminus T' = \emptyset$.

This condition also implies that $T' \cup \{s\} = T' \cup S$. Let $\Delta'(s|y_{T'})$ denote marginal contribution of $s$ when added by policy after $T'$. We have

\[ b'_i \leq B \cdot \frac{\Delta'(s|y_{T'})}{g(y_{T'} \cup \{s, y_s\})} = B \cdot \frac{\Delta'(s|y_{T'})}{g(y_{T'} \cup y_S)} \]

(1)

\[ \leq B \cdot \frac{\Delta'(s|y_{T'})}{g(y_S)} \leq B \cdot \frac{\Delta_s}{g(y_S)} \]

(2)

Setting $b'_i = \alpha \cdot B \cdot \frac{\Delta_s}{g(y_S)}$, we get

\[ \alpha = 1 \]

(3)

First inequality in step 1 follows from the proportional share allocation criteria and second equality follows from the fact that $T' \cup \{s\} = T' \cup S$. In step 2, first inequality follows from monotonicity of function $g$ and second inequality follows from the fact that increasing the bid by $s$ can only pushes her position lower in the allocation, decreasing the marginal contribution. Note that here $\Delta_s$ is used to denote the marginal contribution of $s$ when it was allocated at position $i$ in the original run of the policy. Finally, in step 3, the inequality holds for $\alpha = 1$.

Case 2: $S \setminus T' = R$

We have

\[ b'_i \leq B \cdot \frac{\Delta'(s|y_{T'})}{g(y_{T'} \cup \{s, y_s\})} \leq B \cdot \frac{\Delta_s}{g(y_{T'} \cup \{s, y_s\})} \]

(4)

Setting $b'_i = \alpha \cdot B \cdot \frac{\Delta_s}{g(y_S)}$, we get

\[ \frac{g(y_{T'} \cup \{s, y_s\})}{g(y_S)} \leq \frac{1}{\alpha} \]

(5)

Now, consider adding some user on top of $y_{T'} \cup \{s, y_s\}$. For some $r_0 \in R$, it must hold that marginal value by unit cost of adding $r_0$ is higher than that of adding whole $R$. We have,

\[ \frac{g(y_R \cup y_{T'} \cup \{s, y_s\}) - g(y_{T'} \cup \{s, y_s\})}{B'(R)} \leq \frac{\Delta'(r_0|y_{T'} \cup \{s, y_s\})}{b'_{r_0}} \leq \frac{\Delta'(s|y_{T'})}{b'_i} \]

(6)
In step 6, first inequality holds from submodularity of $g$ and second holds from that fact that $s$ was chosen to be added on set $T'$ compared to $r_0$ at position $j$ by the alternate run of the mechanism. In step 7, first inequality follows from the fact that increasing the bid by $s$ can only pushes her position lower in the allocation, decreasing the marginal contribution. The second inequality holds by setting $b'_i = \alpha \cdot B \cdot \frac{\Delta_s}{g(y_S)}$.

Now, using the fact that $B'(R) \leq B$, and $g(y_S) \leq g(y_S \cup y_{T'}) = g(y_R \cup y_{T'} \cup \{s, y_s\})$, we have

\[
\frac{g(y_S) - g(y_{T'} \cup \{s, y_s\})}{B} \leq \frac{g(y_S)}{B \cdot \alpha}
\]  

(8)

\[
\frac{g(y_{T'} \cup \{s, y_s\})}{g(y_S)} \geq (1 - \frac{1}{\alpha})
\]  

(9)

Combining step 5 and step 9, we get an upper bound on $\alpha = 2$. \hfill \square

**Proof of Theorem 8.4.** Consider running the mechanism with reduced budget of $\frac{B}{2}$ (i.e. setting parameter $\alpha = 2$ in the mechanism). Let a set $S$ allocated by mechanism and $\theta_S$ be the payments made to participants. By summing over these payments, we get :

\[
\sum_{s \in S} \theta_s \leq \sum_{s \in S} \alpha \cdot \frac{\Delta_s}{\sum_{s \in S} \Delta_s} \cdot \frac{B}{2} \leq B.
\]

The inequality here holds from Lemma B.5 which bounds the maximum threshold payment for a participant $s$ by $\alpha \leq 2$. \hfill \square

### B.4 Guarantees on the Utility: Proof of Theorem 8.4

Proof of Theorem 8.4 rests on proving following two lemmas. In Lemma B.6, we first prove an upper bound on the utility of optimal sequential (untruthful) mechanism SEQOPT as $\frac{e}{(e - 1)}$ times the utility on sequential greedy mechanism SEQGREEDY, with an extra additive factor of $f_{max}$. Then, in Lemma B.7, we show that, because of diminishing returns property of the utility functions, the stopping criteria used by the mechanism based on proportional share and using only $\alpha$ proportion of the budget still allows the allocation of sufficiently many participants to achieve a competitive
amount of utility for the application. Additionally, we use the fact that in our settings, the utility contribution of each participant is small compared to the overall utility achieved by the mechanism.

We use \( \pi_{OPT}, \pi_G, \pi_{TG} \) to denote the allocation policies of mechanisms SEQOPT, SEQGREEDY and SEQTGREEDY. Also, we use \( g_{avg}(\pi) \) to denote the average expected utility obtained by running the allocation policy \( \pi \). We use the terms mechanism and policy interchangeably whenever clear from the context.

**Lemma B.6.** Expected utility of optimal sequential policy SEQOPT is bounded by the utility of sequential greedy policy SEQGREEDY as \( g_{avg}(\pi_{OPT}) \leq c/(c-1)[g_{avg}(\pi_G) + f_{max}] \).

**Proof.** Let \( \pi_G \) executes for \( l \) steps allocating a set \( S_l \). Let us renumber the users in order of which they were considered during execution of \( \pi_G \) and denote \( S_{i+1} = \{1, 2, \ldots , i-1, i, \ldots , l, l+1\} \) where \( l + 1 \) is the first unallocated user because of budget constraint. Consider the step when participant \( i \) is added by the policy on top of \( S_{i-1} \). We consider the expected marginal utility of executing the whole \( \pi_{OPT} \) after step \( i - 1 \), conditioned on observations \( y_{S_{i-1}} \). Let \( y_T \) be the final set of participants along with observations obtained by executing \( \pi_{OPT} \) after \( S_{i-1} \), where \( y_{S_{i-1}} \subseteq y_T \). Let \( r \in T \setminus S_{i-1} \). Given the submodularity of \( g \), it must hold that:

\[
\frac{g_{avg}(y_T \cup y_{S_{i-1}}) - g_{avg}(y_{S_{i-1}})}{B(T) - B(S_{i-1})} \leq \frac{\Delta_r}{b_r} \leq \frac{\Delta_i}{b_i} \tag{1}
\]

\[
\frac{g_{avg}(\pi_{OPT}) - g_{avg}(y_{S_{i-1}})}{B} \leq \frac{g_{avg}(y_{S_i}) - g_{avg}(y_{S_{i-1}})}{b_i} \tag{2}
\]

\[
g_{avg}(y_{S_i}) \geq \frac{b_i}{B} \cdot g_{avg}(\pi_{OPT}) + (1 - \frac{b_i}{B}) \cdot g_{avg}(y_{S_{i-1}}) \tag{3}
\]

Step 1 uses the fact that \( i \) was chosen over \( r \) by greedy policy. Step 2 uses the definition of \( \Delta_i = g_{avg}(y_{S_i}) - g_{avg}(y_{S_{i-1}}) \) and \( g_{avg}(\pi_{OPT}) \leq g_{avg}(y_{T}) \). By recursively applying step 3 results for \( l + 1 \) steps, we get:

\[
g_{avg}(y_{S_{i+1}}) \geq \left[1 - \prod_{i \in [1..l+1]} \left(1 - \frac{b_i}{B}\right)\right] \cdot g_{avg}(\pi_{OPT})
\]

\[
\geq \left[1 - \left(1 - \frac{B(S_{i+1})}{B} \cdot \frac{1}{l+1}\right)^{l+1}\right] \cdot g_{avg}(\pi_{OPT}) \tag{4}
\]

\[
\geq \left[1 - \left(1 - \frac{1}{l+1}\right)^{l+1}\right] \cdot g_{avg}(\pi_{OPT}) \tag{5}
\]

\[
\geq (1 - \frac{1}{e}) \cdot g_{avg}(\pi_{OPT}) \tag{6}
\]
Step 4 uses the fact that minimum of the product for \( n \) variables \( 1 - \prod_{i \in [1..n]} \left( 1 - \frac{x_i}{n} \right) \) is achieved when all the variables take value as \( x_i = \frac{X}{n} \) (where \( X = \sum_{i \in [1..n]} x_i \)). Step 5 uses the fact that \( B(S_{l+1}) > B \) and Step 6 uses the limiting value of the equation.

\[
g_{\text{avg}}(\pi_{\text{OPT}}) \leq (\frac{e}{e-1}) \cdot (g_{\text{avg}}(y_{S_l}) + \Delta_{l+1})
\]

(7)

\[
\leq (\frac{e}{e-1}) \cdot (g_{\text{avg}}(\pi_G) + f_{\text{max}})
\]

(8)

In step 7, we used the fact that \( g_{\text{avg}}(y_{S_{l+1}}) = g_{\text{avg}}(y_{S_l}) + \Delta_{l+1} \). In step 8, we used the fact that \( \Delta_{l+1} \leq f_{\text{max}} \) and \( g_{\text{avg}}(\pi_G) = g_{\text{avg}}(y_{S_l}) \).

**Lemma B.7.** Expected utility of sequential greedy policy \( \text{SEQGREEDY} \) is bounded by the utility of truthful greedy policy \( \text{SEQTGREEDY} \) as \( g_{\text{avg}}(\pi_G) \leq (1 + \alpha)g_{\text{avg}}(\pi_{TG}) + \alpha f_{\text{max}} \).

**Proof.** Let \( \pi_G \) executes for \( l \) steps allocating a set \( S_l \) and \( \pi_{TG} \) terminates after \( k \leq l \) steps because of additional stopping criteria allocating a set \( S_k \subseteq S_l \). Let us renumber the users in order of which they were considered during execution of \( \pi_G \) and denote \( S_l = \{1,2,\ldots,k,k+1,\ldots,l\} \). Since \( k + 1 \) was not allocated by the \( \pi_{TG} \), we have:

\[
b_{k+1} > \frac{B}{\alpha} \cdot \frac{\Delta_{k+1}}{\sum_{i \in S_k} \Delta_i + \Delta_{k+1}}
\]

Also, because of decreasing marginal utility by cost ratio of the users considered by the policy, we get:

\[
\frac{b_j}{\Delta_j} \geq \ldots \geq \frac{b_j}{\Delta_j} \geq \ldots \geq \frac{b_{k+1}}{\Delta_{k+1}} > \frac{B}{\alpha} \cdot \frac{1}{\sum_{i \in S_k} \Delta_i + \Delta_{k+1}}
\]

\[
\Rightarrow \forall j \in [k+1 \ldots l], b_j > \frac{B}{\alpha} \cdot \frac{\Delta_j}{\sum_{i \in S_k} \Delta_i + \Delta_{k+1}}
\]

\[
B \geq \sum_{j \in [k+1 \ldots l]} b_j > \frac{B}{\alpha} \cdot \frac{\sum_{j \in [k+1 \ldots l]} \Delta_j}{\sum_{i \in S_k} \Delta_i + \Delta_{k+1}}
\]

\[
\alpha \cdot (g_{\text{avg}}(\pi_{TG}) + \Delta_{k+1}) \geq (g_{\text{avg}}(\pi_G) - g_{\text{avg}}(\pi_{TG}))
\]

(9)

\[
g_{\text{avg}}(\pi_G) \leq (1 + \alpha)g_{\text{avg}}(\pi_{TG}) + \alpha f_{\text{max}}
\]

(10)

In step 9, we used the fact that \( g_{\text{avg}}(\pi_G) = \sum_{i \in S_l} \Delta_i \) and \( g_{\text{avg}}(\pi_{TG}) = \sum_{i \in S_k} \Delta_i \). In step 10, we used the fact that \( \Delta_{k+1} \leq f_{\text{max}} \).

**Proof of Theorem 8.4.** Combining the results of above two lemmas, we get:

\[
g_{\text{avg}}(\pi_{OPT}) \leq \frac{(1 + \alpha) \cdot e}{e - 1} \cdot \left[g_{\text{avg}}(\pi_{TG}) + f_{\text{max}}\right]
\]

\[
= \frac{(1 + \alpha) \cdot e}{e - 1} \cdot \left(1 + \frac{f_{\text{max}}}{g_{\text{avg}}(\pi_{TG})}\right) \cdot g_{\text{avg}}(\pi_{TG})
\]
Now, we set \( \alpha = 2 \). Also, using the fact that \( \frac{f_{\text{max}}}{g_{\text{avg}}(\pi_{TG})} \ll 1 \) (i.e. each user can only contribute to a maximal of \( f_{\text{max}} \) utility to the application which, for a large-scale application, is very small compared to utility achieved by mechanism under given budget), we get an approximation factor of \( 1/4.75 \) (\( = 0.22 \)). \( \square \)
C

Proofs of Part IV

C.1 Hardness Results: Proof of Proposition 11.1

\textbf{Proof of Proposition 11.1.} We reduce from set cover. Suppose we are given a collection of finite sets $S_1, \ldots, S_n$ jointly covering a set $W$. We reduce the problem of finding a smallest subcollection covering $W$ to the teaching problem with the special case $\alpha = \infty$.

Let $H = W \cup \{h^*\}$, that is, each element in $W$ is a hypothesis that misclassifies at least one data point. We use a uniform prior $p(h) = \frac{1}{|W|+1}$. For each set $S_j$, we create a teaching example $x_j$. The label output by hypothesis $h(x_j) = 1$ iff $h \in S_j$, otherwise $h(x_j) = -1$. We set $h^*(x) = -1$ for all examples. Thus, selecting $S_i$ in the set cover problem is equivalent to selecting example $x_i$. It is easy to see that constructing the examples can be done in polynomial (in fact, linear) time.

The expected error after showing a set of examples is less than $\frac{1}{|W|+1} n$ if and only if sets indexed by $A$ cover $W$. Thus, if we could efficiently find the smallest set $A$ achieving error less than $\frac{1}{|W|+1} n$, we could efficiently solve set cover. \hfill $\Box$
C.2 Useful Lemmas

Before proving the main theorems, we state an important lemma that will be needed throughout the analysis.

Lemma C.1. Assume that the learner’s current hypothesis $h_t$ is governed by the stochastic process described in Section 11.3. Then, the marginal distribution of $h_t$ is given by $P_{t-1}(h)$ in every time step $t$.

Proof. Let the marginal distribution of $h_t$ denoted by $P'_t(h)$. We will show by induction that for every $t$, $P'_t = P_t$.

Obviously, $P'_0 = P_0$ by definition. Now, as for the induction hypothesis, let us assume that $P'_{t-1} = P_{t-1}$. By the definition of the stochastic process we have

$$P'_t(h) = \frac{1}{Z_t} (P'_{t-1}(h)\mathbb{I}\{y_t = h(x_t)|h,x_t\} + P_t(h)\mathbb{I}\{y_t \neq h(x_t)|h,x_t\})$$

$$= \frac{1}{Z_t} (P_{t-1}(h)\mathbb{I}\{y_t = h(x_t)|h,x_t\} + P_{t-1}(h)P(y_t|h,x_t)\mathbb{I}\{y_t \neq h(x_t)|h,x_t\})$$

$$= \frac{1}{Z_t} P_{t-1}(h)\mathbb{I}\{y_t = h(x_t)|h,x_t\} + P(y_t|h,x_t)\mathbb{I}\{y_t \neq h(x_t)|h,x_t\}$$

$$= \frac{1}{Z_t} P_{t-1}(h)P(y_t|h,x_t)\mathbb{I}\{y_t \neq h(x_t)|h,x_t\} = P_t(h),$$

as stated. 

C.3 Guarantees on the Convergence of STRICT: Proof of Theorem 11.1

Proof of Theorem 11.1. Clearly, $F(A)$ can be written as

$$F(A) = \sum_{h \in \mathcal{H}} P_0(h)G_h(A)\text{err}(h,h^*),$$

where

$$G_h(A) = 1 - \prod_{\substack{x \in A \\ y(x) \neq \text{sgn}(h(x))}} P(y(x)|h,x).$$
It is easy to see that $G_h(A)$ is submodular for every $h \in H$. Thus, $F(A)$ is also submodular.

Let us start to upper bound the expected error of the learner. For that, we need the following simple observation:

$$\frac{P(h|A)}{P(h^*|A)} = \frac{Q(h|A)}{Q(h^*|A)} = \frac{Q(h|A)}{P_0(h^*)}.$$ 

Now for the upper bounding:

$$\sum_{h \in H} P(h|A) \text{err}(h,h^*) \leq \sum_{h \in H} \frac{P(h|A)}{P(h^*|A)} \text{err}(h,h^*)$$

$$= \frac{1}{P_0(h^*)} \sum_{h \in H} Q(h|A) \text{err}(h,h^*)$$

$$= \frac{1}{P_0(h^*)} (E - F(A)),$$

where $E = \sum_{h \in H} P_0(h) \text{err}(h,h^*)$ is an upper bound on the maximum of $F(A)$. This means that if we choose a subset $A$ such that $F(A) \geq E - P_0(h^*)\epsilon$, it guarantees an expected error less than $\epsilon$. In the following, we assume that $F(X) \geq E - P_0(h^*)\epsilon/2$. If this assumption is violated, the Theorem still holds, but the bound is meaningless, since OPT($P_0(h^*)\epsilon/2$) = $\infty$ in this case.

Since $F(A)$ is submodular (and monotonic), we can achieve $E - P_0(h^*)\epsilon$ “level” with the greedy algorithm, as described below. We use the following result of the greedy algorithm for maximizing submodular functions:

**Theorem** ([KG14], based on [NWF78]). Let $f$ be a non-negative monotone submodular function and let $S_t$ denote the set chosen by the greedy maximization algorithm after $t$ steps. Then we have

$$f(S_t) \geq \left(1 - e^{-\ell/k}\right) \max_{S:|S| = k} f(S)$$

for all integers $k$ and $l$.

Let $k^*$ be the cardinality of the smallest set $A^*$ such that $F(A^*) \geq E - P_0(h^*)\epsilon/2$. Thus we know that

$$\max_{A:|A| = k^*} F(A) \geq E - P_0(h^*)\epsilon/2.$$
Appendix C. Proofs of Part IV

Now we set \( \ell = k^* \log \frac{2E}{P_0(h^*)} \) and we denote \( A_\ell \) the result of the greedy algorithm after \( \ell \) steps, and we get

\[
F(A_\ell) \geq \left( 1 - e^{-1/k^*} \right) \left( E - \frac{P_0(h^*)e}{2} \right)
= \left( 1 - \frac{P_0(h^*)e}{2E} \right) \left( E - \frac{P_0(h^*)e}{2} \right)
\geq E - p(h^*)e,
\]
proving that running the greedy algorithm for \( \ell \) steps achieves the desired result.

\[\square\]

C.4 Teaching Complexity for Linear Separators: Proof of Theorem 11.2

Proof of Theorem 11.2. We introduce a randomized teaching policy called Relaxed-Greedy Teaching Policy (sketched in Policy C.1) and prove that with positive probability, the policy reduces the learner error exponentially. Then, we use the standard probabilistic argument: positive probability of the above event implies that there must exist a sequence of examples that reduce the learner error exponentially. We finish the proof of the theorem by using the result of Theorem 11.1.

Based on our model, the way the learner updates his/her belief after showing example \( x_t \in \mathcal{X} \) and receiving answer \( y_t = \text{sgn}(h^*(x_t)) \) is as follows:

\[
P_{t+1}(h) = \frac{1}{Z_t} P_t(h) w_{t}^{(1-\zeta_t(h))/2},
\]
where \( \zeta_t(h) = \text{sgn}(h(x_t)) \cdot y_t \), the term \( Z_t \) is the normalization factor, and \( 0 < w_t < 1 \) is a parameter by which the learner decreases the weight of inconsistent hypotheses. Note that \( w_t \) may very well depend on the examples shown, i.e., for hard examples \( w_t \) is typically larger than those of the easy ones as the learner is more certain about his/her answers. However, here we assume that \( w_t \leq w_0 < 1 \) and that \( w_0 \) is known to the teacher. In other words, the teacher knows the minimum weight updates imposed by the learner on inconsistent hypotheses. As a result, the teacher can track \( P_{t+1}(h) \) conservatively as follows:

\[
P_{t+1}^{(l)}(h) = \frac{1}{Z_t} P_{t}^{(l)}(h) w_{t}^{(1-\zeta_t(h))/2}. \tag{C.1}
\]
Policy C.1: Relaxed-Greedy Teaching Policy (RGTP)

1. **Input**: examples $\mathcal{X}$, hypothesis $\mathcal{H}$, prior $P_0$, error $\epsilon$.
2. **Initialize**: $t = 0, P_0^0(h) = P_0$
3. **while** $1 - P_t^t(h^*) > \epsilon$ **do**
   4. **if** there exists two neighboring polytopes $\mathcal{P}$ and $\mathcal{P}'$ s.t. $\sum_h P_t^t(h) h(\mathcal{P}) > 0$ and $\sum_h P_t^t(t) h(\mathcal{P}') < 0$ **then**
      5. select $x_t$ uniformly at random from $\mathcal{P}$ or $\mathcal{P}'$.
   **else**
      6. select $x_t$ from polytope $\mathcal{P} = \arg \min_{\mathcal{P} \in \Pi} |\sum_h P_t^t(h) h(\mathcal{P})|$
   **end**
7. **∀h ∈ H** update $P_{t+1}^t(h)$ according to (C.1) and $t \rightarrow t + 1$.

**Theorem C.1.** Let $\mathcal{H}$ be a collection of $n$ linear separators and choose an $0 < \epsilon < 1$. Then, under the condition that $\mathcal{X}$ is $m$-rich, RGTP guarantees to achieve

$$\Pr(1 - P_m(h^*) > \epsilon) < \frac{(1 - \epsilon)(1 - P_0(h^*))}{\epsilon \cdot P_0(h^*)} e^{-m(1 - w_0)/4},$$

by showing $m$ examples in total. In other words, to have $\Pr(1 - P_m(h^*) > \epsilon) < \delta$, RGTP uses at most the following number of examples:

$$m = \frac{4}{1 - w_0} \log \left( \frac{(1 - \epsilon)(1 - P_0(h^*))}{\delta \cdot \epsilon \cdot P_0(h^*)} \right).$$

The above theorem requires that $\mathcal{X}$ gets a richer space for obtaining better performance. When we have a uniform prior $P_0 = 1/n$, the above bounds simplify to

$$m = \frac{4}{1 - w_0} \log \left( \frac{(1 - \epsilon)n}{\delta \cdot \epsilon} \right).$$

As at least $\log n$ queries is required to identify the correct hypothesis with probability one, the above bound is within a constant factor from $\log n$ for fixed $\epsilon$ and $\delta$.

The proof technique is inspired by [BZ74], [KK07], and in particular beautiful insights in [Now11]. To analyze RGTP let us define the random variable

$$\eta_t^t = \frac{1 - P_t(h^*)}{P_t(h^*)}.$$
This random variable $\log(\eta_t)$ was first introduced by [BZ74] in order to analyze the classic binary search under noisy observations (for the ease of exposure we use $\eta_t$ instead of $\log(\eta_t)$). It basically captures the probability mass put on the incorrect hypothesis after $t$ examples. Similarly, we can define

$$\eta^{(t)}_t = \frac{1 - P^{(t)}_t(h^*)}{P^{(t)}_t(h^*)}.$$ 

A simple fact to observe is the following lemma.

**Lemma C.2.** For any sequence of examples/labels $\{(x_t, y_t)\}_{t \geq 0}$, and as long as $0 \leq w_l \leq w_0 \leq 1$ we have $\eta^{(l)}_t \leq \eta^{(t)}_t$.

Note that RGTP is a randomized algorithm. Using Markov’s inequality we obtain

$$\Pr(1 - P_m(h^*)) > \epsilon) \leq \Pr(1 - P^{(t)}_h(h)) > \epsilon)$$

$$= \Pr \left( \eta_t > \frac{\epsilon}{1 - \epsilon} \right)$$

$$\leq \frac{(1 - \epsilon)\mathbb{E}(\eta^{(t)}_t)}{\epsilon}.$$ 

The above inequalities simply relates the probability we are looking for in Theorem C.1 to the expected value of $\eta^{(t)}_t$. Hence, if we can show that the expected value decreases exponentially fast, we are done. To this end, let us first state the following observation.

**Lemma C.3.** For any sequence of examples/labels $\{(x_t, y_t)\}_{t \geq 0}$, and for $0 \leq w_0 < 1$ the corresponding random variable $\{\eta^{(t)}_t\}_{t \geq 0}$ are all non-negative and decreasing, i.e.,

$$0 \leq \eta^{(s)}_t \leq \eta^{(t)}_t \leq \frac{1 - P_0(h^*)}{P_0(h^*)}, \quad s \geq t.$$ 

The above lemma simply implies that the sequence $\{\eta_t\}_{t \geq 0}$ converges. However, it does not indicate the rate of convergence. Let us define $\mathcal{F}_t = \sigma(P_0^{(t)}, p_1^{(t)}, \ldots, p_t^{(t)})$ the sigma-field generated by random variables $P_0^{(t)}, p_1^{(t)}, \ldots, p_t^{(t)}$. Note that $\eta^{(t)}_t$ is a function of $p_t^{(t)}$ thus $\mathcal{F}_t$-measurable. Now, by using the towering property of the expectation we obtain

$$\mathbb{E}(\eta^{(t)}_t) = \mathbb{E}(\mathbb{E}(\eta^{(t)}_t | \mathcal{F}_{t-1})) = \mathbb{E}(\mathbb{E}(\eta^{(t)}_t | \mathcal{F}_{t-1}) \eta^{(t)}_{t-1}) \eta^{(t)}_{t-1} | \mathcal{F}_{t-1}).$$

Since $\eta^{(t)}_{t-1}$ is $\mathcal{F}_{t-1}$-measurable we get

$$\mathbb{E}(\eta^{(t)}_t) = \mathbb{E}(\eta^{(t)}_{t-1} \mathbb{E}(\eta^{(t)}_t | \mathcal{F}_{t-1}))$$
\[ \leq \mathbb{E}(\eta_{t-1}^{(t)}) \max_{\mathcal{F}_{t-1}} \mathbb{E}(\frac{\eta_{t-1}^{(t)}}{\eta_{t-1}^{(t)}} | \mathcal{F}_{t-1}). \]

The above inequality simply implies that

\[ \mathbb{E}(\eta_{t}^{(t)}) = \frac{1 - P_0(h^*)}{P_0(h^*)} \left( \max_{0 \leq s \leq t-1} \mathbb{E}(\frac{\eta_{s+1}^{(t)}}{\eta_{s}^{(t)}} | \mathcal{F}_{s}) \right)^t \] (C.2)

In the remaining of the proof we derive a uniform upper bound (away from 1) on \( \mathbb{E}(\frac{\eta_{t}^{(t)}}{\eta_{t-1}^{(t)}} | \mathcal{F}_{t-1}) \), which readily implies exponential decay on \( \Pr(1 - P_m(h^*) > \epsilon) \) as the number of samples \( m \) grows. For the ease of presentation, let us define the (weighted) proportion of hypothesis that agree with \( y_t \) as follows:

\[ \delta_t = \frac{1}{2} \left( 1 + \sum_h P_t^{(t)}(h) \xi_t(h) \right). \]

Along the same line, we define the proportion of hypothesis that predict + on polytope \( \mathcal{P} \) as follows

\[ \delta^+_\mathcal{P} = \frac{1}{2} \left( 1 + \sum_h P_t^{(t)}(h) h(\mathcal{P}) \right). \]

Now, we can easily relate \( \delta_t \) to the normalization factor \( Z_t \):

\[ Z_t = \sum_h P_t^{(t)}(h) w_o^{(1-\xi_t(h))/2} = (1 - \delta_t) w_o + \delta_t. \]

As a result

\[ P_{t+1}^{(t)}(h) = \frac{P_t(h) w_o^{(1-\xi_t(h))/2}}{(1 - \delta_t) w_o + \delta_t}. \]

In particular for \( P_{t+1}^{(t)}(h^*) \) we have

\[ P_{t+1}^{(t)}(h^*) = \frac{P_t(h)}{(1 - \delta_t) w_o + \delta_t}. \]

To simplify the notation, we define

\[ \gamma_t = (1 - \delta_t) w_o + \delta_t. \]

Hence,

\[ \frac{\eta_{t+1}^{(t)}}{\eta_t^{(t)}} = \frac{\gamma_t - P_t^{(t)}(h^*)}{1 - P_t^{(t)}(h^*)}. \]
Note that since \( P_t(h^*) \) is \( F_t \)-measurable the above equality entails that
\[
\mathbb{E} \left( \frac{\eta_{t+1}^{(t)}}{\eta_{t}^{(t)}} \mid F_t \right) = \frac{\mathbb{E}(\gamma_t \mid F_t) - P_t(h^*)}{1 - P_t(h^*)}.
\]

Thus we need to show that \( \mathbb{E}(\gamma_t \mid F_t) \) is bounded away from 1. To this end, we borrow the following geometric lemma from [Now11].

**Lemma C.4.** Let \( H \) consists of a set of linear separators where each induced polytope \( P \in \Pi \) contains at least one example \( x \in X \). Then for any probability distribution \( p \) on \( H \) one of the following situations happens

1. either there exists a polytope \( P \) such that \( \sum_h p(h)h(P) = 0 \), or
2. there exists a pair of neighboring polytopes \( P \) and \( P' \) such that \( \sum_h p(h)h(P) > 0 \) and \( \sum_h p(h)h(P') < 0 \).

The above lemma essentially characterizes Ham Sandwich Theorem [LMS94] in discrete domain \( X \) that is 1-rich. In words, Lemma C.4 guarantees that either there exists a polytope where (weighted) hypothesis greatly disagree, or there are two neighboring polytopes that are bipolar. In either case, if an example is shown randomly from these polytopes, it will be very informative. This is essentially the reason why RGTP performs well.

Now, let \( P_t \) be the polytope from which the example \( x_t \) is shown. Then, based on \( y_t \) we have two cases:

- if \( y_t = + \) then \( \gamma_t^+ = \gamma_t = (1 - \delta_{P_t}^+)w_o + \delta_{P_t}^+ \),
- if \( y_t = - \) then \( \gamma_t^- = \gamma_t = \delta_{P_t}^+ w_o + 1 - \delta_{P_t}^+ \).

Note that for any \( x_t \) picked by RGTP we have \( 0 < \delta_{P_t}^+ < 1 \), since it never shows an example that all hypothesis agree on. As a result, both \( \gamma_t^+ \) and \( \gamma_t^- \) are between 0 and 1.

Based on Lemma C.4 there are only two cases. Let us define the auxiliary random variable \( s_t \) that simply indicate in which case we are. More precisely, \( s_t = 1 \) indicates that we are in case 1 and \( s_t = 2 \) indicates that we are in case 2. To be formal we define \( G_t = \sigma(P_0^{(t)}, p_1^{(t)}, \ldots, p_{t-1}^{(t)}, s_t) \). Note that \( F_t \subset G_t \) and thus \( \mathbb{E}(\gamma_t \mid F_t) = \mathbb{E}(\mathbb{E}(\gamma_t \mid G_t) \mid F_t) \).

We need to prove the following technical lemma.
Lemma C.5.
\[ \mathbb{E}(\gamma_t | G_t) \leq \max \left\{ \frac{3 + w_o}{4}, \frac{1 + w_o}{2}, 1 - \frac{(1 - w_o)(1 - P_i^{(t)}(h^*))}{2} \right\} . \]

**Proof.** Let us first condition on \( s_t = 1 \). Then, RGTP chooses an \( x_t \in P_t \) in which case \( \delta^+_p = 1/2 \) and results in \( \gamma^+_t = \gamma^-_t = (w_o + 1)/2 \). Hence, given \( s_t = 1 \), we have
\[ \mathbb{E}(\gamma_t | G_t) = (w_o + 1)/2. \] (C.3)

The conditioning on \( s_t = 2 \) is a little bit more elaborate. Recall that in this case RGTP randomly chooses one of \( P \) and \( P' \). Note that \( \delta^+_p > 1/2 \) and \( \delta^+_p < 1/2 \). Now we encounter 4 possibilities:

1. \( h^*(P) = h^*(P') = + \): condition on \( s_t = 2 \) we have
   \[ \mathbb{E}(\gamma_t | G_t) = \frac{\gamma^+_t + \gamma^-_t}{2} \leq \frac{1 + (1 - \delta^+_p) w_o + \delta^+_p}{2} \leq \frac{3 + w_o}{4} \] (C.4)
   where we used the fact that \( \gamma^+_t \leq 1 \) and \( (1 - \delta^+_p) w_o + \delta^+_p \) is an increasing function of \( \delta^+_p \) and that \( \delta^+_p < 1/2 \).

2. \( h^*(P) = h^*(P') = - \): similar argument as above shows that
   \[ \mathbb{E}(\gamma_t | G_t) \leq \frac{3 + w_o}{4} . \]

3. \( h^*(P) = -, h^*(P') = + \): In this case we have
   \[ \mathbb{E}(\gamma_t | G_t) = \frac{\gamma^+_t + \gamma^-_t}{2} = \frac{\delta^+_p w_o + 1 - \delta^+_p + (1 - \delta^+_p) w_o + \delta^+_p}{2} = \frac{1 - \frac{1 - w_o}{2}}{2 (1 + \delta^+_p - \delta^+_p')} \leq \frac{1 + w_o}{2} \] (C.5)
   where we used the fact \( 0 \leq \delta^+_p - \delta^+_p' \leq 1 \).

185
4. $h^*(P) = +, h^*(P') = -$: since $P$ and $P'$ are neighboring polytopes, $h^*$ should be the common face. Hence, we have $\delta^+_P - \delta^+_P = P_i(t)(h^*)$. As a result

$$
\mathbb{E}(\gamma_t|G_t) = \frac{\gamma^+_t + \gamma^-_t}{2} = \frac{(1 - \delta^+_P)w_o + \delta^+_P + \delta^+_P \cdot w_o + 1 - \delta^+_P}{2} = \frac{1 + \delta^+_P - \delta^+_P + w_o(1 - \delta^+_P + \delta^+_P)}{2} \leq 1 - \frac{(1 - w_o)(1 - P_i(t)(h^*))}{2}.
$$

(C.6)

By combining (C.4), (C.5) and (C.6) we prove the lemma. □

Lemma C.5 readily implies that

$$
\mathbb{E} \left( \frac{\eta_{i+1}^{(t)}}{\eta_{i}^{(t)}} | F_t \right) = \frac{\mathbb{E}(\gamma_t|F_t) - P_i^{(t)}(h^*)}{1 - P_i^{(t)}(h^*)} \leq \frac{3 + w_o}{4}.
$$

Hence,

$$
\mathbb{E}(\eta_i^{(t)}) = \frac{1 - P_0(h^*)}{P_0(h^*)} \left( 1 - \frac{1 - w_o}{4} \right)^t \leq \frac{1 - P_0(h^*)}{P_0(h^*)} \exp(-t \cdot (1 - w_o)/4)
$$

This finishes the proof of Theorem C.1. Now, to finish the proof of Theorem 11.2, we just set $\delta$ to $1/2$ and use the probabilistic argument mentioned in the beginning of the proof, resulting in an upper bound on OPT. Since we use the logistic likelihood function, $w_o$ can be bounded by $\frac{1}{2}$. Then, we apply the result of Theorem 11.1. □
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Key ingredients of our approach with a goal towards building self-sustainable and intelligent crowd-powered systems.</td>
<td>7</td>
</tr>
<tr>
<td>1.2</td>
<td>Learning about incentives with applications to balancing bike sharing systems (Part II of this dissertation). These results are based on Singla and Krause [SK13b] and Singla et al. [Sin+15a].</td>
<td>10</td>
</tr>
<tr>
<td>1.3</td>
<td>Incentives for learning with applications to community-based air quality sensing (Part III of this dissertation). These results are based on Singla and Krause [SK13a].</td>
<td>12</td>
</tr>
<tr>
<td>1.4</td>
<td>Learning as an incentive with applications to educating citizen scientists for biodiversity monitoring (Part IV of this dissertation). These results are based on Singla et al. [Sin+14b].</td>
<td>14</td>
</tr>
<tr>
<td>4.1</td>
<td>Interaction with a user in the \textit{posted-price} model: the user with private cost ( c_u ) accepts the offered payment ( p ) if ( p \geq c_u ).</td>
<td>30</td>
</tr>
<tr>
<td>4.2</td>
<td>( b/p ) represents the budget constraint, \textit{i.e.}, maximum utility that can be achieved given an infinite pool of users. ( N \cdot F(p) ) represents the utility with unlimited budget. The optimal price ( p^* ) lies at the intersection of these two curves. Discretized ( K ) prices are used by our mechanism, ( p' ) corresponds to the optimal price among these ( K ) prices.</td>
<td>33</td>
</tr>
</tbody>
</table>
5.1 Acquired utility for simulated distributions when varying budget. (a) and (b) show results for i.i.d. settings for bidding and posted-price model, respectively. In (b), BP-UCB outperforms PP’12 by over 150% increase in utility for the stochastic settings. (c) considers workers arriving in order of ascending bids. (d) considers two groups of workers with bids uniformly distributed in $[0.1, 0.5]$ and $[0.5, 0.9]$ arriving one after another.

5.2 Uniform distribution in $[0.1, 0.9]$, stochastic settings. In (b), no-regret properties of BP-UCB can be seen as the average regret diminishes with increase in budget. (c) shows better convergence rate of BP-UCB compared to PP’12. (d) shows that BP-UCB makes low offers in beginning, in contrast to PP’12 which quickly exhausts the budget.

5.3 Survey study on the Mechanical Turk platform about an option to participate in a hypothetical advertisement system for a social networking site. (a) Distribution of workers’ bids, (b) correlation with usage time, and (c) correlation with the number of friends online.

5.4 Utility for the workers’ distribution for bids in $[10, 100]$ collected from the Mechanical Turk platform (MTurk), varying budget. In (b), BP-UCB outperforms PP’12 by over 180% increase in utility. Also, BP-UCB and BP-DGReedy are robust against all the online settings above.

6.1 Overview of the system, our contributions are shown in the dotted box.

6.2 Survey study with customers of a real-world BSS in the city of Mainz, Germany where we deployed our incentives system.

6.3 Simulation results based on historical data of Boston’s Hubway BSS and survey data from customers of the BSS in the city of Mainz, Germany.

6.4 Results from deployment of our incentives system through smartphone app on a real-world BSS in the city of Mainz, Germany in beta-test phase for 30 days.

8.1 Illustration of the protocol by which the proposed system interacts with the users.
8.2 The sensing region is uniformly discretized into a set of locations $\mathcal{V}$ indicated by the dots. (a) illustrates a population of users, along with their sensing profiles in (b). The set of users selected by the system in absence of privacy are shown in (c). However, to protect privacy, users only share an obfuscated location with the system in (d) and a collection of sensing profiles ($\{y_{w1}, y_{w2}, y_{w3}\}$ for user $w$) in (e). The privacy profile of user $w$, given by $Y_w$, is the uniform distribution over these sensing profiles, given by $P(Y_w = y_{w}) = \frac{1}{3}$. (f) shows the selection of the participants in presence of uncertainty introduced by privacy profiles. The actual sensing profile is only revealed to the system after a user has been selected. .................................................. 86

9.1 (a) Bids ($) and sensitivity ([1−100]) for different levels of privacy trade-off. (b) Distribution of bids for sharing location at a granularity level of zip codes. ................................................................. 101

9.2 (a) NODE+ wireless sensor platform with expansion sockets on both ends to attach different sensor modules. (b) shows the set up of calibration experiments at EMPA: NODE+ sensors are installed at the vent which feeds the air to the sensing device employed by EMPA. (c) shows that the NODE+ CO2 gas sensor has high-quality measurements. .... 103

9.3 (a) Two NODE+ sensors mounted on a bike, transferring the sensor measurements in real-time to the user’s smartphone via Bluetooth connectivity. (b) shows a screenshot of the NODE+ app running on the smartphone displaying the measurements in real-time. ............... 104

9.4 (a) Red dots represent a set of 300 sensing locations $\mathcal{V}$ in the center of New York City obtained using the OpenStreetMap data. (b) Heat map of rides on Strava in the New York City used to get probability distribution of users across different locations. .................................................. 106

9.5 (a) and (b) compares SEQGREEDY using $\alpha = 2$ w.r.t. a variant using an optimized value of $\alpha$. ................................................................. 108

9.6 For a fixed obfuscation level of 400 meters radius, (a) varies the budget given and (b) varies the desired utility. ................................. 109
9.7 Results from varying the obfuscation level. (b) and (c) measure utility acquired for a given budget of 10$ and show about 5% adaptivity gain. (e) and (f) measure the budget required (in $) to achieve a utility of 150 and show up to 30% adaptivity gain. .......................... 110

11.1 Illustration of the process of teaching the crowd. Given a large set of images, the teacher (randomly) picks a small teaching set. For this set, expert labels, as well as candidate features and hypotheses used by the crowd are elicited (cf., Chapter 12). The teacher then uses this information to teach the rest of the crowd to label the rest of the data, for which no features or labels are available. The teacher sequentially provides an unlabeled example from the teaching set to the participant, who attempts an answer. Upon receipt of the correct label, the participant may update her hypothesis before the next example is shown. .................. 119

12.1 Task of classification of synthetic insect images into two hypothetical species Vespula and Weevil (denoted as VW). (a) shows sample images of the dataset $\mathcal{X}$. (b) shows the 2-D embedding of synthetic images for the features: head/body size proportion ($f_1$) and head/body color contrast ($f_2$), normalized around origin. It shows four of the hypotheses in $\mathcal{H}$, with the target hypothesis $h^*$ in red. .................. 131

12.2 Task of distinguishing butterflies and moths on real images (denoted as BM). (a) shows sample images of the dataset $\mathcal{X}$. (b) shows the 2-D embedding of images of this data set, and the hypotheses for a small set of participants, as obtained using the approach of Welinder et al. [Wel+10]. .................. 133

12.3 Task of identification of birds belonging to an endangered species of woodpeckers from real images (denoted as WP). (a) shows sample images of the dataset $\mathcal{X}$. (b) shows the 13 features used for representation of woodpecker images and the $w_{h^*}$ vector of the target hypothesis. It also lists the average number of times a particular feature is present in the images of a given species. .................. 135
12.4 (a) compares the algorithms’ teaching performance in terms of simulated learners’ test error (VW task). (b) shows the robustness of STRICT w.r.t. unknown $\alpha$ parameters of the learners. Thus, a noise-tolerant teacher (i.e., $\alpha < \infty$) performs much better than noise-free teaching, even with misspecified $\alpha$. (c) shows how the difficulty of STRICT’s examples naturally increase during teaching.

12.5 The order of examples picked (top 5) by the teaching algorithms. For the VW and BM tasks, we embed the picked examples in the 2-D feature space in Figures 12.1(b) and 12.2(b), respectively.

12.6 (a-c) show the teaching performance of our algorithm measured in terms of test error of human learners (participants recruited from the Mechanical Turk platform) on hold out data. STRICT is compared against Set-Cover and Random teaching, as we vary the length of teaching.
1.1 Algorithms and mechanisms presented in this dissertation, along with their theoretical guarantees. .................................................. 15

5.1 Statistics of the data reported for different years .......................... 52

8.1 Different information settings and existing mechanisms that fall short of either privacy-preservation, adaptivity, or truthfulness. Our main mechanism $\text{SEQGREEDY}$ satisfies all the desirable properties. ....... 89
Bibliography


BIBLIOGRAPHY


[Bai+06] Xiaole Bai, Santosh Kumar, Dong Xuan, Ziqiu Yun, and Ten H Lai. “Deploying wireless sensors to achieve both coverage and connectivity”. In: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing. ACM. 2006, pp. 131–142 (cit. on p. 105).


[BH05] Avrim Blum and Jason Hartline. “Near-optimal online auctions”. In: SODA. 2005 (cit. on pp. 37, 40).


BIBLIOGRAPHY


Matthew Richardson and Ryen W. White. “Supporting synchronous social Q&A throughout the question lifecycle”. In: WWW. 2011 (cit. on p. 79).


Jasper Schuijbroek, Robert Hampshire, and Willem-Jan van Hoeve. Inventory rebalancing and vehicle routing in bike sharing systems. 2013 (cit. on p. 73).


