

HARVARD UNIVERSITY
Graduate School of Arts and Sciences



DISSERTATION ACCEPTANCE CERTIFICATE

The undersigned, appointed by the


School of Engineering and Applied Sciences

have examined a dissertation entitled:

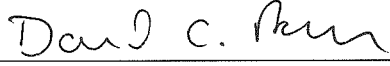
“Experimental Studies of Human Behavior in Social Computing Systems”

presented by : Qiushi Mao

candidate for the degree of Doctor of Philosophy and here by
certify that it is worthy of acceptance.

Signature 


Typed name: Professor Y. Chen

Signature 

Typed name: Professor D. Parkes

Signature 

Typed name: Dr. E. Horvitz

Signature 

Typed name: Dr. D. Watts

Date April 30, 2015

THIS PAGE INTENTIONALLY LEFT BLANK

Experimental Studies of Human Behavior
in Social Computing Systems

A dissertation presented

by

Qiushi Mao

to

The School of Engineering and Applied Sciences

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in the subject of

Computer Science

Harvard University

Cambridge, Massachusetts

April 2015

© 2015 Qiushi Mao

All rights reserved.

Experimental Studies of Human Behavior in Social Computing Systems

Abstract

Social computing systems, fueled by the ability of the Internet to engage millions of individuals, have redefined computation to include not only the application of algorithms but also the participation of people. Yet, the true impact of social computing in the future depends on a systematic understanding of how to design interventions that produce desirable system-wide behavior. Behavioral experiments, with their fundamental ability to study causality, are an important methodology in reaching this goal.

This dissertation presents several examples of how novel experimental approaches to studying social computing systems can not only improve the design of such systems, but improve our understanding of human behavior. We investigate the differences in motivation and effects of incentives in a crowdsourcing task across paid and volunteer crowdsourcing systems, finding that varying financial incentives even at the same wage can be used to implicitly invoke different biases. We discover the effects of assembling teams of different size in a social, collaborative crisis mapping task, finding that larger groups exert less effort per individual, but make up for this loss in their ability to coordinate. We investigate the accuracy of voting in a human computation setting and how statistical models can be used to discover patterns of decision making across a population of individuals. Finally, we present the design and implementation of *TurkServer*, an software system that enabled these experimental studies.

The work in this dissertation suggests that experiments in social computing present an opportunity both for understanding design factors of social computing systems and for developing generalizable models of human behavior—and ultimately better theories of how people communicate and interact in our interconnected world.

Contents

1	Introduction	1
1.1	Technical Contributions	4
1.2	Methods Contributions	6
1.3	Dissertation Overview	7
2	Background	8
2.1	Design Factors in Social Computing Systems	8
2.1.1	Economic Incentives	9
2.1.2	Social and Non-monetary Incentives	10
2.1.3	Collective Interaction and Coordination	13
2.1.4	Preferences, Perception, and Information Aggregation	17
2.2	Behavioral Experiments for Social Computing Systems	18
2.2.1	What Exactly is an Experiment?	18
2.2.2	System Design and Evaluation	20
2.2.3	Modeling Human Behavior	22
2.3	Conclusion	24
3	Volunteer and Paid Crowdsourcing: From Galaxy Zoo and Planet Hunters to Amazon Mechanical Turk	25
3.1	Preliminaries	25
3.1.1	Engagement and Attention in Volunteer Crowdsourcing	26
3.1.2	Effects of Payment Schemes in Crowdsourcing	27
3.2	Predicting Engagement in Volunteer Crowdsourcing	29
3.2.1	Data, Outcomes, and Model	30
3.2.2	Galaxy Zoo as Testbed	31
3.2.3	Evaluation	39
3.3	Comparison of Paid and Unpaid Crowdsourcing	46
3.3.1	Task Model	47
3.3.2	Planet Hunters	48
3.3.3	Experiment Design	50
3.3.4	Results	55
3.4	Discussion	63
3.5	Acknowledgments	65

4	Performance and Team Size on a Crisis Mapping Task	66
4.1	Preliminaries	66
4.1.1	Crisis Mapping and the Standby Task Force	67
4.1.2	Typhoon Pablo Deployment	68
4.2	Experiment Design	69
4.2.1	Collaborative Real-time Mapping Application	71
4.2.2	Input Data	73
4.2.3	Subject Recruitment and Training	73
4.2.4	Informed Consent	74
4.2.5	Group Assignment	74
4.2.6	Worker Incentives and Monitoring	76
4.3	Evaluation Methods	76
4.3.1	Computation of “Person-Hours”	77
4.3.2	Constructing the Gold Standard	77
4.3.3	Performance Measures	79
4.3.4	Intermediate Group Performance	80
4.3.5	Comparison with SBTF Deployment	81
4.3.6	Synthetic Groups	82
4.3.7	Measuring Effort	82
4.3.8	Measuring Collaboration	83
4.4	Results	84
4.5	Discussion	88
4.6	Acknowledgments	90
5	Voting and Probabilistic Ranking Models for Social Computing	91
5.1	Preliminaries	91
5.1.1	Voting in Human Computation	93
5.1.2	Social Choice Theory	95
5.1.3	Ranking Models and Random Utility	97
5.2	Comparison of Voting Via Synthetic Data	100
5.3	Design of Experimental Voting Data	102
5.3.1	Sliding Puzzles	102
5.3.2	Pictures of Dots	103
5.3.3	Comparison of Domains	104
5.3.4	Methodology	105
5.4	Comparison of Voting Via Human Data	107
5.5	Variation and Uncertainty in Ranking Data	109
5.5.1	Sushi Ranking Dataset	109
5.5.2	Voting Decision Data	113
5.6	Properties of Ranking Models	117
5.7	Discussion	119
5.8	Acknowledgments	121

6	Design and Implementation of a Web-Based Experimental System	122
6.1	Deploying Web-based Experiments	122
6.1.1	Advantages of Online Experiments	123
6.1.2	Challenges for System Implementation	124
6.1.3	The Experiment Design Triangle	124
6.1.4	Participant Comprehension and Attention	126
6.1.5	Limitations of Amazon Mechanical Turk	126
6.2	Designing and Conducting Experiments using TurkServer	128
6.2.1	Web Technology and Software	128
6.2.2	Software Abstractions	130
6.2.3	Interactive Tutorials	131
6.2.4	Improved Recruitment and Scheduling	133
6.2.5	Qualitative Observation	134
6.2.6	Monitoring and Data Logging	136
6.2.7	Randomization Methodology	139
6.3	Imagining the Future of Virtual Experiment Labs	140
7	Conclusion	142
7.1	Summary of Contributions	142
7.2	Challenges for Experimental Design	144
7.3	Future Directions in Online Experiments	146
	Bibliography	149
A	Tutorial Text for Crisis Mapping Experiment	166

Acknowledgments

I consider myself exceptionally fortunate to have learned from many excellent mentors over the course of my graduate studies.

My advisor Yiling Chen has continually provided a nurturing environment in which I was able to develop my research interests into where they stand today. Yiling has been extremely patient and supportive of me for many years, and always challenged me to keep the important questions in mind as I often missed the forest for the trees. Without a doubt, Yiling's mentorship provided latitude for both of us to discover and pursue new, interesting problems.

David Parkes has been an unparalleled mentor and advocate for our research group, and allocates a superhuman amount of time for both his students and for improving our department. He inspires me with his very sharp process of thought and ability to approach just about any problem, even in the span of a short meeting. Our first project together, despite creating quite a mess, nevertheless started me on the road where I stand today.

Much of the work in this thesis emerged through very fruitful collaboration with colleagues at Microsoft Research. Eric Horvitz and Ece Kamar graciously accepted me as an intern on short notice during the tumultuous spring of 2012, resulting in a wonderfully productive summer. Working with Ece taught me how engaging a good research collaboration can be, and Eric has always challenged me to think far outside the box of my limited initial assessment of any research question.

Collaborating with Sid Suri and Duncan Watts at Microsoft Research NYC defied my expectations for what was possible in novel interdisciplinary research. My work stands on the shoulders of giants like Sid, who has conveyed to me an immeasurable amount of knowledge about conducting behavioral experiments. I also aspire to learn from Duncan's talent in communicating powerfully through writing and effortlessly bridging multiple disciplines. Both Sid and Duncan broke a customary rule of avoiding remote collaboration to work on the crisis mapping project, and for that I am sincerely appreciative.

The EconCS group at Harvard and my other colleagues in the computer science depart-

ment provided a wonderful place to both work and grow as a person. Through many late nights at the office, John Lai and I developed a strong friendship that endures beyond graduate school. In working closely on many research projects, Alice Gao has become a good friend as well as a co-conspirator in pushing forward the boundaries of online experiments. Hossein Azari is not only a sharp research partner, but a highly skilled Nerf gun marksman who often surprised me with a shot to the head from a foam dart. Michael Gelbart and I became fast friends through table tennis and gastronomy, and he repeatedly helped me summon the courage to complete this dissertation. For that, I will be ever grateful.

My friends in Boston over the years, including at the Harvard Graduate Christian Fellowship and Highrock Church, have greatly enriched my life and also supported the inception and growth of my marriage. Amy, my dearly beloved wife, has been incredibly supportive and understanding despite riding alongside me on the graduate school rollercoaster, and has often provided for me in perhaps the most fundamental way—the cooking of many delicious meals. And although my parents never ceased to inquire about the progress of my research and pretended to be only marginally satisfied at my dissertation defense, they have always believed in and supported me.

Finally, I thank God, who has given me everlasting joy and purpose in my life, and blessed me with the abilities and opportunity to complete this work.

Dedication

To my wife Amy,
who has been the perfect life partner in every way;
and to my parents,
who made it possible to be where I am today.

Chapter 1

Introduction

Man is by nature a social animal; an individual who is unsocial naturally and not accidentally is either beneath our notice or more than human. Society is something that precedes the individual. Anyone who either cannot lead the common life or is so self-sufficient as not to need to, and therefore does not partake of society, is either a beast or a god.

— Aristotle, *Politics*

Over two thousand years ago, the Greek philosopher Aristotle remarked that people are social creatures by nature. Indeed, this led to the creation of *society*—through our interpersonal relationships from groups of few people to kingdoms and even countries, we have explored new lands, made innumerable scientific advances, and developed a variety of unique, beautiful culture all over the world.

Yet, the social nature of our world has undergone dramatic changes merely in the last few decades. The global expansion of the Internet, connecting the world together at the speed of light, has brought everyone closer together and created a new form and medium in the nature of interpersonal communication and social behavior. It is now possible to send a message instantly to someone thousands of miles away, without the verbal intonation and body language that accompany physical communication. Electronic communication has brought together groups and organizations that are much too large, or too decentralized, to

fit in any physical space, and created new paradigms of organization. Our social fabric is increasingly digital, and our computational systems increasingly social.

As a result, we are now in the age of *social computing*. The Internet, through connecting people together through Web-based systems and integrating computational algorithms with human participation, has enabled new paradigms of production and innovation that were unimaginable not long ago. Wikipedia now contains thousands of times as much information in any print encyclopedia, and is updated continually through the efforts of volunteers. May large, complex software projects such as the Linux kernel and the Python programming language are developed and supported not by for-profit corporations, but by decentralized organizations of volunteers contributing effort to open-source projects. Social networks such as Twitter and Facebook facilitate the spread of information at faster rates than ever before. Platforms such as Yelp, AirBnB, and Uber connect people together not just in the digital world, but in the physical world as well.

The ubiquity of social computing systems also yields a transformative approach for understanding many aspects of human behavior. Through the growth of new Web technologies, the increasing adoption of Internet-connected smartphones, and greater technological literacy, “digital breadcrumbs” reveal information for all types of human behavior (Lazer et al. 2009). Detailed data about the activity of not just individuals, but also groups and organizations, is available in such copious quantities that the problem is often not about having *enough* data but simply finding the *right* data to answer a particular question. We have entered a revolutionary era where it is possible to study the behavior of people at greater levels of scale and interaction, as well as in novel paradigms, that were impossible in the past.

Despite the apparent success of social computing and the torrent of available data, the design of these hybrid systems of computers and people still bears resemblance to a trial-and-error process, resulting from the challenge of causally predicting how people will behave in different contexts. In contrast to the predictable, well-understood mechanics of machines, people may act in surprising and unexpected ways. Yet, fundamental to designing systems that operate as intended is the ability to choose interventions that causally affect outcomes.

Despite significant research into the processes underlying social computing in the fields of *human computation* (Quinn and Bederson 2011), *crowdsourcing*, *computational social science* (Watts 2013), *algorithmic game theory*, and *algorithmic mechanism design*, we argue that computer science has still far to go in understanding and making causal predictions about human behavior in social computing.

As a result, we are faced with a distinct choice when designing for desirable outcomes in social computing systems. On one hand, we may use the approach of treating behavior as a black box, using observational data and intuition to make design decisions. This is common, but risks finding solutions that only work in particular cases, with a general lack of understanding about *why*. Alternatively, we can engage in a more systematic test of hypotheses about how interventions affect behavior. In this case, we may not only design a better system, but also learn general knowledge about human behavior extending to other contexts and building new theory about human behavior along the way.

The randomized experiment is the most compelling method to cleanly test *causal* predictions about behavior. In avoiding the expense of experiments, Watts (2014) argues that fields such as sociology tend to understand human behavior by simply projecting oneself into a situation and rationalizing “common sense”, resulting in at best unscientific and at worst misleading research. However, online experiments are increasingly attractive as they both scale and generalize beyond traditional methods and can be applied to novel paradigms of behavioral interaction that are only seen online. As a result, experiments are an integral aspect of a systematic approach to studying and modeling human behavior for designing social computing systems.

In this dissertation, I argue that social computing systems and experimental design are a promising mix for innovation, answering novel questions using new methods. Experiments allow the systematic testing of causal hypotheses for designing desirable interventions for human participants. At the same time, experiments in social computing systems can run at a scale much larger than possible with past experimental methodology, and with more realistic environments to study human behavior. As a result, such experiments can produce

generalizable knowledge for the study of human behavior at large.

1.1 Technical Contributions

The core technical contributions in this dissertation center around the use of novel experimental methods in tandem with existing techniques for understanding design factors in social computing systems. Observations about behavior in a particular context often apply more generally, allowing for broader application to system design.

Crowdsourcing on the Internet has become a scalable method for recruiting many people for applying their intellect to a variety of activities. However, there is still much to learn about how incentives affect participation and what drives user interest across paid and volunteer systems. Chapter 3 studies the relationship of *incentives* and *engagement* from volunteer to paid crowdsourcing. We first engage in an study of regularities of volunteer contribution in Galaxy Zoo, a volunteer citizen science platform, finding that disengagement (dropout) rates for users can be predicted with high accuracy, especially when taking into account a user’s past behavior. We then study a more nuanced citizen science task with variable difficulty, based on the Planet Hunters citizen science project, in a paid crowdsourcing setting. We design an experiment comparing different types of monetary incentives at the same wage level. This experiment makes the novel comparison of both volunteers and paid crowd workers on the same task, and additionally finds that paying an hourly wage can yield better results than existing payment methods.

Although many online activities including citizen science are intensely social, much current research still focuses on individual paradigms of work because of its simplicity of study. Chapter 4 presents an experimental study of social *group interaction and coordination* in an online setting. We experimentally vary the sizes of teams that are working on a crisis mapping task—a real-world task used by digital volunteers to help respond to natural disasters. Using novel and extensive instrumentation techniques, we not only measure the effectiveness of teamwork, but also capture the process of how individuals collaborate and coordinate together. We find that although individuals working independently are able to produce more

raw output in the absence of coordination costs in teams, the ability of teams to collaborate and coordinate is valuable and outweighs the cost of coordination. Additionally, our experiment demonstrates that digital crisis mapping using appropriate software and infrastructure can potentially significantly outperform existing volunteer organizations. This work has implications for both the application of volunteer crisis mapping and the study of organization and collective intelligence at large.

Finally, a common problem in many social computing systems is the process of reliably aggregating noisy information from many individuals. Chapter 5 explores how experimental data can be used to discover patterns of noise and mistakes in voting and ranking problems, and combined with statistical models to achieve a better understanding of human behavior. In order for information aggregation processes to be effective, we must first accurately characterize cognitive behavior. Chapter 5 first explores how a carefully designed experiment can be used to evaluate aggregation properties of different voting mechanisms in realistic settings, and compared to common modeling approaches. We find that the commonly used plurality voting rule achieves similar performance on human-generated data to other, more complex voting mechanisms. Then, we apply a probabilistic ranking model to this experimental data as well as other types of ranking data and show how it can be used to summarize preferences and perception across a population of people.

Experiment	Treatment Conditions	Summary of Findings
Payment Incentives (Section 3.3)	Piece-rate payment methods at the same wage	Paying by time (an hourly wage) compares favorably to other conditions and minimizes strategic behavior
Group Coordination (Chapter 4)	Group sizes from 1 to 32 individuals	People produce more work alone than working in groups, but groups are valuable in the ability to coordinate
Voting Aggregation (Section 5.3)	Ranking problems of different difficulty	Plurality voting achieves similar results to other voting mechanisms, while being simpler to implement

Table 1.1: Summary of experimental findings

Table 1.1 summarizes the experiments, treatments, and results described in this dissertation. These findings, though general to many contexts, still only scratch the surface of

design factors that are important to understand in social computing, and point to the need for additional studies to better understand human behavior.

1.2 Methods Contributions

Alongside the technical contributions, this dissertation also presents novel experimental methods, leveraging software techniques, that extend the possibilities of experimental studies. These methods and techniques are presented mainly in Chapter 6 as part of the *TurkServer* software framework, but also reflected in the experiments presented in Chapters 3–5. The work described in this thesis exemplifies the newfound methodological power of software-controlled, web-based experimental techniques.

First, we show that experiments for social computing systems have *flexibility to be larger-scale and more realistic* than past lab-based studies of human behavior. Not only do experiments conducted online have access to more participants than possible in physical studies, but they can also study them in natural environments as opposed to a contrived lab environment. Chapter 3 leverages this by studying a task from a volunteer citizen science setting in an online labor market for paid crowdsourcing, thus using a well-motivated and real task to measure the effects of financial incentives. Chapter 4 simulates a realistic crisis mapping deployment scenario while still allowing for significant experimental control and observation.

Second, the use of software *greatly expands the availability of observable data* in experimental studies. Just as “big data” refers to the flood of behavioral data available about Internet users, “big experiments” can collect either more data or simply more fine-grained data to better support studies. Chapter 4 shows how detailed observational data can be collected in a realistic experimental setting, including the implementation of a software “one-way window” with live observation and replay capabilities. This allows for the observation of not just the effects of an experimental treatment but also evaluation of many potential explanations for *why*. Chapter 6 describes the software implementation that facilitated this real-time, fine-grained data collection.

Finally, experiments can be a source of *designed data* for studying and modeling behavior.

Despite the myriad behavioral data available on the Internet, it can be often more of a challenge to find the *right data* for a particular context or as a subset of another dataset. A properly designed experiment can produce precisely the data needed to study a particular problem. This is exemplified by the experiments in Chapter 4 in collecting detailed data for understanding how coordination and interaction occur in groups, and in Chapter 5 for using probabilistic models to understand human perception and preferences.

In summary, experiments are a powerful and complementary tool to existing methods for designing social computing systems to produce desirable outcomes. Software-based experiments conducted on social computing systems are only in their infancy, and innovations in methodology—including those in this dissertation—have the potential to transform the scale and detail at which human behavior can be studied, both as a way to both design better systems and to produce generalizable knowledge.

1.3 Dissertation Overview

The dissertation is organized as follows. Chapter 2 reviews existing work in social computing systems, particular questions of interest, and experimental methodology. Chapter 3 presents empirical and experimental work comparing users in volunteer and paid crowdsourcing systems. Chapter 4 presents an experiment on collaborative online problem solving, studying collaboration in virtual groups of different sizes. Chapter 5 presents an experiment to study noise in voting and information aggregation settings, and how appropriate models can be used to characterize user behavior. Chapter 6 describes the design of the TurkServer experimental platform and general systems and methods for supporting software-controlled experiments. Chapter 7 concludes.

Chapter 2

Background

The effective study of social computing systems necessarily draws on a wide range of techniques and disciplines, including computer science, economics, psychology, and sociology. This chapter aims to review a range of relevant literature. Section 2.1 illustrates different design factors in social computing by example and motivates the behavioral paradigms that require understanding within them. Section 2.2 argues why experiments are a compelling technique to answer these questions and how they complement other methods.

2.1 Design Factors in Social Computing Systems

The term *social computing* has been used in different contexts to refer to various definitions, such as the decentralization of information technology in the information systems literature (Parameswaran and Whinston 2007) and the “*computational facilitation ... of human social dynamics*” (Wang et al. 2007). Quinn and Bederson (2011) position social computing as overlapping with *crowdsourcing* and *human computation* in the general realm of *collective intelligence*, with the emphasis that social computing focuses on facilitating “natural human behavior”.

In this work, we will use social computing in a more literal sense to refer to systems of people and computers that are electronically connected, with computation done through both social and algorithmic mechanisms. Critically, the notion of *computation* refers beyond

simply the task of electronic computers to also include the products of human processes and interaction. This dissertation focuses on three different areas of social computing—economic and non-monetary *incentives*, group *interaction and coordination*, and aggregation of *preferences and perception*. This section motivates these areas in more detail.

2.1.1 Economic Incentives

Arguably the most significant question in social computing systems is how people are motivated to contribute, collaborate, and otherwise invest personal effort. The clearest form of incentives are economic or monetary, where users are paid for their time and effort. Numerous crowdsourcing applications, such as those fielded on Amazon Mechanical Turk (MTurk), reimburse people with monetary payments for their efforts on tasks. In paid crowd work, workers are compensated for completing tasks created by *requesters* in a marketplace or other assignment mechanism. Amazon Mechanical Turk, the predominant example of an *online labor market*, makes a sizable force of workers available for paid crowdsourcing (Ipeirotis 2010, Horton and Chilton 2010). MTurk hosts a large variety of tasks, including data verification, language translation, and audio transcription. Other tasks include studies of human computation techniques and behavioral experiments (Ipeirotis 2010). Workers performing tasks through MTurk are often aware of their compensation and self-organize to find the best-paying and most interesting tasks (Chandler et al. 2013). MTurk has also seen adoption as a participant pool for research using human subjects in many fields (Paolacci et al. 2010, Sprouse 2011, Berinsky et al. 2012, Mason and Suri 2012a, Crump et al. 2013).

Online labor markets also include other systems such as ODesk, CrowdFlower, and MobileWorks, as well as hybrid digital/physical systems like TaskRabbit and Elance. *Open innovation* platforms such as 99designs, InnoCentive, and Kaggle run contests for participants to submit the designs, machine learning algorithms, or other techniques in a competition to determine the best option. Companies such as TaskRabbit and Uber have created markets for crowdsourcing embedded in the real world, where physical transactions occur aided by communication through and matching of the system.

In the context of paid crowdsourcing, researchers have studied how the magnitude of financial incentives affects work produced. Horton and Chilton (2010) conducted an experiment to estimate the reservation wage of workers in MTurk. Harris (2011) studied performance-contingent financial incentives (both rewards and penalties) and showed that the quality of work was higher in the presence of such incentives than in their absence.

There is a disconnect between experiments such as those above and theory in this area—the latter often relies on *rational agent* models from economics, where agents are able to maximize a utility function in choosing their actions (e.g. Singer and Mittal (2013) and Ho et al. (2014)). However, there has also been a growing literature on the limitations of rational agent models (Gigerenzer and Selten 2002) in the field of *behavioral economics*—particularly pertinent for considering people in the design of social computing systems. Online labor markets make use of microtasks and very small payments, and there is strong evidence of deviation from rational behavior: prior work has variously demonstrated the effects of anchoring (Mason and Watts 2009, Yin et al. 2013) and task-specific features (Ho et al. 2015), as well as interaction with non-monetary incentives as in the following section. Section 3.3 builds upon these findings in a different direction—instead of varying the magnitude of incentives, we explore the effects of different piece-rate payments at the *same wage rate* on a crowdsourcing task in a real online labor market.

2.1.2 Social and Non-monetary Incentives

Despite the absence of economic incentives or any explicit rewards, many social computing systems still exhibit large amounts of user participation. Non-monetary incentives often combine with or substitute for monetary incentives in so-called *peer production systems* where users may be rewarded with reputation, recognition, or intrinsic interest. Other social norms such as reciprocity and altruism also affect user contribution (Fehr and Schmidt 2006). Benkler (2009) argues that peer production is an alternative to traditional notions of economic organization such as the firm, and should be studied and promoted as a new cooperative mode of production. Examples of peer production include Wikipedia, which has far eclipsed

any previous encyclopedia in terms of the amount of knowledge and its ability to keep up with recent developments, but operates on a model of almost open editing and relying on community members to update articles and fix vandalism (Priedhorsky et al. 2007, Kittur and Kraut 2008). The extensive collaboration on projects in open-source communities like GitHub is also driven by the varied motivations of participating users (Dabbish et al. 2012).

Many social computing systems *depend* on user participation to be successful. Internet aggregators such as Reddit and Hacker News use a mix of user participation and ranking algorithms to identify and share desirable content. These systems must be designed to both promote user contributions and identify quality content in order to build a sustainable community (Stoddard 2015). Q&A systems like StackOverflow rely the actions of both committed users and occasional visitors in a non-economic incentive system to build a repository of useful programming knowledge (Mamykina et al. 2011, Anderson et al. 2013).

Effectively designing non-monetary incentives is arguably one of the most difficult problems facing designers of social computing systems. For example, consider the history of the answer voting system in StackOverflow. After a user asks a programming question, others vote on the quality of submitted answers, and good answers gain prominence and their authors gain reputation—by all appearances, a very simple mechanism. When the site launched in 2008, the answers were sorted in an apparently straightforward way by their votes—first by vote count, then by age of submission, with older answers first. This led to the **fastest gun in the west** (FGITW) problem (Mamykina et al. 2011)¹, whereby users would post low-quality answers to a question very quickly, and gain votes simply because they were displayed first, and earlier.

After observing this issue, system designers tried to fix it by randomly ordering all answers with the same vote count. This led to a different problem, the **slowest cheater in the east** (SCITE), where question answerers would strategically downvote others’ answers in an attempt to garner more votes by appearing first, then remove those downvotes after sufficient popularity (as down-voting incurs a penalty). This problem was “fixed” through the current

¹See also: <http://meta.stackexchange.com/questions/18014/what-is-fgitw-and-scite-on-mso>

rather convoluted system where votes on any answer would be locked in after 5 minutes unless the answer was edited, such that the penalty effectively became permanent.

Many social computing systems arise in similar ad hoc ways, with initial designs guided using intuition and further changes through a trial-and-error process. As another example, consider the many open-source software communities competing for users’ attention—SourceForge, Google Code, and Microsoft’s Codeplex, among others—prior to the current popularity of GitHub. Yet, GitHub’s social model has become so phenomenally successful that Google and Microsoft, among other websites, have closed their respective code repository services (Google Code² and Codeplex³) and encouraged users to switch to GitHub. Clearly, there is a distinct difference in the organization of successful versus unsuccessful open-source software, but earlier system designers failed to account for the social patterns of open-source software development.

Understanding the motivation of volunteers is also essential in crowdsourced *citizen science*; one of the best known examples is the *Zooniverse*⁴, connecting scientists seeking human eyes on large amounts of data with participants interested in contributing to science (known as *citizen scientists*), and has been successful in producing valuable data for research. Examples of Zooniverse projects include Galaxy Zoo (Lintott et al. 2008), where galaxies are classified according to their shapes, and Planet Hunters⁵ (Fischer et al. 2012, Schwamb et al. 2012, Lintott et al. 2013), where participants identify potential signals of planets orbiting distant stars. We study the predictability of volunteer engagement in Galaxy Zoo in Section 3.2.

Citizen science systems rely solely on voluntary contributions of amateur participants without providing any monetary compensation, and volunteers run the gamut from a core community with strong intrinsic motivation (e.g. interest in a scientific discipline) to casual participants who visit the site once and leave (Raddick et al. 2013). Citizen science platforms can further engage users through other mechanisms. FoldIt, the online protein folding game,

²Google Code has shut down as of March 2015: <http://google-opensource.blogspot.com/2015/03/farewell-to-google-code.html>

³Microsoft has open-sourced many important projects such as the Roslyn compiler, TypeScript, and ASP.NET on GitHub.

⁴<http://www.zooniverse.org>

⁵<http://www.planethunters.org>

uses a smooth interface and the prospect of competition to entice all users, not just those interested in science, to play (Khatib et al. 2011). Other citizen science projects, such as bird identification and tracking over large regions, use bird watchers’ engagement with a local community to promote participation (McCaffrey 2005, Sullivan et al. 2009).

Many volunteer-based systems may be vulnerable in the future to the problems around public goods and free-riding (Ledyard 1994)⁶, and the varying degrees of success in such systems suggests that it is important to understand why some work and others do not. Similarly, the attractiveness of systems such as the Zooniverse may depend as much on the scientific interest of participating users as their intrinsic generosity to contribute (Raddick et al. 2013). In order to design systems that work as intended on the first deployment, we must not only draw on computational skills, but gain a better understanding of human behavior using approaches from economics, psychology, and sociology. Chapter 4 embodies this approach, and suggests that future research in social computing systems promises to be an intensely interdisciplinary activity.

2.1.3 Collective Interaction and Coordination

Internet connectivity has created new paradigms of interpersonal communication, allowing information to be disseminated to interested parties at a much faster rate than traditional media or basic Internet browsing. Early forms of social online communication included decentralized systems of e-mail, instant messaging, and blogs, which required significant manual effort by users. Despite suggestions that electronic media may allow for increased deception in the absence of emotional and physical cues in communication (Carlson et al. 2004), Hancock et al. (2007) found that media such as e-mail actually increased honesty through the electronic record that it leaves in perpetuity.

More recently, social networks such as Facebook and Twitter connect certain users together based on relationships and facilitate the spreading and consumption of information. Such networks can potentially have a significant impact on the behavior of their users—a

⁶See Suri and Watts (2011) for an example of an Web-based public goods experiment

2010 experiment at Facebook increased voter turnout in the US Congressional elections by an estimated 340,000 people (Bond et al. 2012). The propagation of information on Twitter has been proposed as a way to build social earthquake detectors (Sakaki et al. 2010) and for analyzing collective sentiment in predicting movements in the stock market (Bollen et al. 2011).

Perhaps paradoxically, the increased connectivity in social computing systems may lead to more extreme individual and collective behavior through electronic communication and anonymity (Sia et al. 2002) and aggregate effects of groupthink (Janis 1972). In particular, evidence that social influence may hinder independent thinking (Lorenz et al. 2011) suggests that increased connections in large networks may lead to more overconfident and polarized beliefs across virtual subgroups of the population—groups that may not have formed in the physical world. Although there is some evidence that the Internet may lead to increased polarization over its supposed social integration (DiMaggio et al. 2001), a recent study of Facebook users found that this effect may in fact be weaker than previously hypothesized (Bakshy et al. 2015).

The scale of interaction enabled by large social computing systems means that we can observe social phenomena that are impossible in the physical world. The field of *network science* studies the structure and links of social networks, leading to observed regularities such as the *friendship paradox* (Hodas et al. 2013) and *small-world networks* (Travers and Milgram 1969, Watts and Strogatz 1998, Dodds et al. 2003). Salganik et al. (2006) conducted a seminal study of whether the “rich get richer” in a music market, finding that while good quality alternatives are generally recognized, lesser options can be significantly influenced by randomness.

Another question is whether social computing can develop mechanisms that maintain the integrity of information in electronic networks while mitigating the impact of negative, anti-social behavior. Several terrifying phenomena have emerged recently, facilitated by the use of social computing systems. Cyberbullying is a growing problem for youth, but much less visible than physical bullying (Smith et al. 2008). Virtual mobs have variously misidentified

the Boston Marathon bombers (Starbird et al. 2014), and engaged in public shaming through human-flesh search engines (Wang et al. 2010), and herding on Twitter (Bercovici 2013). In the fast growth of social computing systems, we may continue to observe very negative phenomena from virtual crowds that can grow to be much bigger than what is possible in the physical world.

Of specific interest to this dissertation in the domain of online social behavior is how large groups can coordinate and collaborate to achieve complex objectives and solve problems. Some observations can be drawn from past work, but electronic collaboration may differ from physical, real-world collaboration, both in the scale of participation and structure of communication. Although there has been some pioneering work on investigating the organizational structures of online organizations such as Wikipedia (Priedhorsky et al. 2007, Kittur and Kraut 2008) and projects on GitHub (Dabbish et al. 2012), there is still significant room for research into how communication patterns emerge.

Since Von Ahn (2005) defined the term *human computation*, there has been significant interest in designing systems to use human intelligence to for tackling problems for that are difficult for machines, or for which machine intelligence is comparably inefficient. The concept of *games with a purpose* (Von Ahn and Dabbish 2008), where humans doing an innately interesting task would also produce useful computation as a side effect, eventually led to the development of the reCAPTCHA system for digitizing books through web security (Von Ahn et al. 2008).

Moving beyond simple human computation tasks has led to the development of *workflows* for as algorithms for managing the input of multiple participants responsible for different subtasks (Little et al. 2010b, Kulkarni et al. 2012) as well as *crowd-powered systems*, which allow implicit or explicit communication between users, aided by a particular system design, to achieve goals such as copyediting (Bernstein et al. 2010), video segmentation (Bernstein et al. 2011b), and collective conversation (Lasecki et al. 2013). Crowd-based human computation can also be embedded in the physical world, as in the example of the winning approach to the DARPA Red Balloon Challenge (Pickard et al. 2011).

There is strong evidence that effective system and user interface design can have profound effects on the ability of groups to work together. Bernstein (2012) demonstrated how crowd-powered systems may facilitate the sharing of context and working memory between individuals. Lasecki et al. (2012) showed that a group of individuals can share a collective, common memory even accounting for turnover, and used this technique to carry on a personal conversation between a collective group and a single user (Lasecki et al. 2013).

In this dissertation, we move beyond directed forms of interaction to study how direct collaboration and communication of people can be effective for problem solving, outside of the structured design of human computation. Prior work has explored on how network structure affects collaboration in groups: Kearns et al. (2006) showed that different network structures had strong effects on group performance on an anti-coordination problem, and Mason and Watts (2012a) showed that an efficient (that is, short average path length) structure actually promotes learning in groups, in contrast to agent-based simulations predicting otherwise. Such studies have generally been restricted to abstract tasks.

There is also evidence that group intelligence is differently represented from the abilities of individuals. Perhaps the canonical measurement of human intelligence was Spearman’s (1904) proposed g factor, but recent research demonstrates that this is quite uncorrelated with the characteristics of high-performing groups (Woolley et al. 2010) and that this observation potentially carries over to electronic settings as well (Engel et al. 2014).

Although existing studies have generally been limited to abstract tasks or artificial contexts, social computing provides novel directions for a new approach. The *Digital Humanitarian Network* (Meier 2015) is a decentralized group of humanitarian organizations that collaborate in the aftermath of natural disasters, epidemics, or political instability to help aid organizations respond effectively by providing information. In particular, the Standby Task Force⁷ is one such organization that uses social, volunteer crowdsourcing to monitor and geolocate reports of crisis events that emerge through social media. Chapter 4 takes studies of group problem solving further by producing a more nuanced understanding of how indi-

⁷<http://blog.standbytaskforce.com/>

viduals collaborate and interact through a highly instrumented experiment on a simulated crisis mapping deployment using real data.

2.1.4 Preferences, Perception, and Information Aggregation

Social computing systems often make use *information elicitation and aggregation* to collect information from many parties and combine it into a useful result. This can take the form of explicit economic mechanisms such as *prediction markets*, shown to be remarkably accurate in comparison to other information forecasting methods (Wolfers and Zitzewitz 2004, Arrow et al. 2008). Markets are prevalent both visibly and inconspicuously on many social computing systems, and have inspired significant research in the field of *algorithmic game theory*, combining approaches using economics and computation. eBay, founded in 1995, is perhaps the earliest example of an auction market in social computing. However, advertisers conduct auctions continually for bidders interested in users’ attention, along a market that includes websites, ad aggregators, advertising firms, and businesses. The auctions in the online advertising market have been studied extensively by both economists and computer scientists (Edelman et al. 2005, Lahaie et al. 2007).

Other elicitation methods such as *peer prediction* may include payments outside the setting of a market (Miller et al. 2005), or explicit mechanisms to promote the sharing of information in the case of the *Delphi method* (Linstone et al. 1975). However, many methods for information elicitation or aggregation are studied purely in the consideration of rational economic agents, and there are examples that suggest a more behavioral approach should be considered to verify the efficacy of theoretical results (Gao et al. 2014).

Another way to aggregate information from users is to use algorithmic approaches based on models of user preference; one large category in particular are *recommender systems* (Resnick and Varian 1997), which attempt to predict users’ preferences based on existing data. Netflix uses the movie preferences and similarities of users to recommend new movies to watch, with the particularly well-known example of the Netflix prize (Bennett and Lanning 2007). Some recommender systems facilitate direct social interaction; for example, dating

sites such as OKCupid even recommend people directly to each other (Rudder 2014).

The application of human computation also often requires aggregation of information. Many types of tasks are easy for humans to do, but require significant domain knowledge and powerful computation for computers to perform similarly, such as image recognition and audio transcription. However, leveraging this relative advantage still requires the ability to reliably aggregate noisy input from multiple people. Little et al. (2010b) showed that this approach has been used to achieve higher quality aggregate solutions for various tasks. In Chapter 5, we explore whether the noisy perception of people can be modeled in a more precise way for better information aggregation, and use these models to summarize a population at large.

2.2 Behavioral Experiments for Social Computing Systems

We now turn to the use of behavioral experiments on the Internet as a primary approach to study the various questions discussed in the previous section.

2.2.1 What Exactly is an Experiment?

The word *experiment* is often used as a term of art in computer science to refer to a simulation where methods or algorithms are compared to each other. In this article, we use experiments to refer specifically to the process of testing for causal effects from *treatment* conditions by *random allocation* of participants across the treatment groups. For example, the hypothesis

*Showing users on Facebook an **I Voted!** button will increase their probability of voting.*

can be tested by an experiment where users are randomly chosen to see the button or not. Validating such a hypothesis is important because it maps system design choices into a resulting (potentially desirable) change in user behavior.

The randomized experiment is the most reliable tool to combat the myriad issues of using anecdotes, intuition, and correlation to derive treatment effects from observational data (Gerber and Green 2012). While intuition is seemingly desirable for explaining hypothe-

ses, common sense often leads us astray when there are two equally intuitive explanations for contrasting results (Watts 2014). On the other hand, using data correlation to support causal hypotheses requires proving that no extraneous *confounding* factor is responsible for the relationship between two observed variables, resulting in a potentially endless list of values that must be measured and controlled for. Yet, there are always potential *unobserved* or unknown factors that are not known to exist or cannot be measured at all (Pearl 2000).

Conducting an experiment conveniently sidesteps these issues by random assignment to treatment conditions, such that any observed *and* unobserved factors are equal, on average, across the treatment groups. Additionally, the experiment aims to *control* or reduce variance in these other factors as much as possible, to maximize the ability to measure the effect of the only source of systematic variation—the treatment condition. As the randomized experiment is in principle a simple and effective idea, it is perhaps surprising that it was not widely adopted in social science until the 1950s-60s (Gerber and Green).

For decades, economists, psychologists, and other social scientists have conducted behavioral research mainly by using small groups of university undergraduates in *behavioral labs*. As a result, many observations about human behavior have been drawn from a sample of WEIRD—Western, educated, industrialized, rich, and democratic—subjects (Henrich et al. 2010). Yet, the lab environment is important for control over the experiment procedure and variability, resulting in less measurement noise despite a more artificial environment.

A carefully designed study might also eschew the procedural control of the lab for a suitable opportunity to conduct an experiment in the *field*, or real world. A field experiment can reach more participants and provide a more convincing test of context-specific hypotheses, but often requires sacrificing the control available in the lab, in addition to requiring significant manpower to coordinate with real-world organizations. The spectrum from the lab to the field (List 2008) strikes a balance between *internal validity*—did the treatment have the hypothesized effect?—and *external validity*—does the observed effect generalize to other contexts? Experiments on the Internet are compelling as they allow for designs that lie anywhere on this spectrum, combining desirable attributes of both lab and field experiments.

For an extensive discussion of experiment design, we refer the reader to Gerber and Green (2012).

The immense number of individuals in constant interaction within social computing systems offers a rich environment for experimental studies of behavior. Online experiments complement other methods such as data modeling and predictive modeling and are particularly promising for two techniques: evaluating system performance and improving models of online human behavior. In this section, we give examples of novel online experiments and methods addressing questions in social computing that are interesting from both a computational and behavioral perspective.

2.2.2 System Design and Evaluation

The most direct application of experimental techniques is to evaluate interventions and design decisions as part of field experiments on live systems, testing hypotheses in a natural environment and at large scale. In the most straightforward form, such experiments are exemplified by A/B or multivariate testing, where arriving users are bucketed into two or more groups, each of which is assigned a different intervention. Although common for many years in marketing, the pace of A/B testing has increased significantly through software infrastructure, and testing frameworks have been adopted by many Internet companies (Kohavi et al. 2009, Tang et al. 2010, Bakshy et al. 2014) with service providers such as Optimizely⁸ also catering to small and medium-sized businesses.

Beyond tests of minor interventions, online field experiments require significant integration between the experiment design and a particular web system. Generally, this is still less costly than physical field experiments, allowing great improvements in scale. Muchnik et al. (2013) used mass participation in a news aggregator to study herding behavior, showing that an arbitrary up-vote on users' comments ultimately raised total votes by up to 25%. Bond et al.'s (2012) experiment on 61 million users in Facebook during the 2010 U.S. congressional dwarfed the scale of past voter turnout experiments in political science, showing that different

⁸<https://www.optimizely.com/>

news feed messages could significantly affect voting turnout and highlighting both the power and responsibility wielded by a large social network. Anderson et al. (2014) found that different presentation of badges on a massive open online course (MOOC) had significant effects on student engagement in the course forums. All of these experiments leveraged the ability of social computing systems to collect more data at lower cost.

Online field experiments in a social computing system can also more convincingly test system design compared to the common approach of evaluation using historical data. Fradkin et al. (2014) investigated several possible reasons for highly skewed review distributions on AirBnB, finding that users reviewed positively due to reciprocity and social interaction while often omitting negative sentiment. *Dataclysm* (Rudder 2014) details the results of several experiments on the OkCupid dating site, showing that users judge each other almost entirely based on pictures, and much more likely to carry on conversations when pictures were temporarily removed. Another experiment tested the effectiveness of OkCupid’s recommendation algorithm by comparing the probability of poor matches initiating conversations relative to good matches. This example shows how experiments are a compelling way to evaluate the overall performance of a mixed algorithmic and behavioral system, which is difficult through data mining alone.

When it is impossible to control or co-opt a live system for experimental purposes, online experiments can also be conducted through participation in existing systems. For example, Hossain and Morgan (2006) showed that eBay buyers did not rationally consider shipping costs when bidding on identical items. Moreover, online labor markets such as Amazon Mechanical Turk (MTurk) and ODesk provide APIs for recruitment of Internet users for almost any task on short notice and for varying amounts of time. These allow for analogues to physical labs on the Internet, proving to be just as reliable in many instances (Horton et al. 2011, Mason and Suri 2012a, Germine et al. 2012), but also allow for the flexibility of field-style experiments in natural settings such as crowdsourcing. Goldstein et al. (2013) designed an experiment on MTurk to test voluntary participation and accuracy of users on an e-mail categorization task in the presence of annoying versus good display advertisements,

finding that while good ads have an indiscernible effect from no advertising, bad ads caused users to both abandon their task sooner and make more mistakes. By using an online labor market to replicate a common user experience across the web, this experiment drew general implications for the externalities of poorly designed display advertising across the Internet.

Finally, experiments can be used to test prototype designs of social computing systems that don't exist in practice. This approach has been used extensively in the human computation literature for developing new paradigms of crowd-powered work. *Workflows* demonstrate how complex tasks can be decomposed into small and skill-insensitive tasks that can be done by any worker (Little et al. 2010b, Kulkarni et al. 2012). Other studies have shown that crowds can also work together in collective interfaces, such as for a near-instant response to user queries (Bigham et al. 2010), collectively editing a paper (Bernstein et al. 2010), and carrying on a personal conversation as a collective (Lasecki et al. 2013). While Bernstein et al. (2011a) point out that research in such systems often chooses between demonstrating a novel system (engineering) or studying a generalizable hypothesis (social science); we argue that experiments can serve the purpose of both testing a system design and also gathering data and testing hypotheses about general human behavior.

2.2.3 Modeling Human Behavior

Beyond simply evaluating a particular system, a broader objective for studying behavior is to develop generalizable models that apply in different contexts. Much of social computing is driven by how users voluntarily participate, collectively interact, and respond to different types of incentives. How do we design optimal incentives for crowdsourcing and online communities? What social incentives improve effectiveness of crowdsourced systems? In fields such as economics and psychology, a strong experimental tradition complements theory and models of behavior. For example, experiments and other empirical work in behavioral economics have resulted in a large literature of models accounting for limits to rational behavior (Gigerenzer and Selten 2002).

Yet, while computer science has developed deep connections to economics in the field

of algorithmic game theory, using the tools of mechanism design, auction theory, and game theory to model systems of online advertising, networks, peer production, and crowdsourcing (Nisan et al. 2007), there has been little experimental research alongside the growing literature of theoretical work. Despite the predominant use of rational agent models for microtask payments, experiments have shown that financial incentives are significantly affected by anchoring effects (Mason and Watts 2009, Yin et al. 2013) and task’s receptiveness to a user’s effort (Ho et al. 2015). When purely theoretical work proceeds independently of empirical verification, we risk solving poorly framed problems or developing models that are far removed from reality.

In many cases, it is difficult to formulate new theory without controlled behavioral data to guide modeling assumptions. In the study of communication and collaboration in groups and networks, Mason and Watts (2012a) showed experimentally that networks with more efficient communication outperformed those with less efficient communication in a collaborative problem with exploration and exploitation—a contrasting finding to previous simulations using agent-based models. Experiments in *collective intelligence* (Woolley et al. 2010, Engel et al. 2014) showed that groups of humans can be quite collaborative, and benefit from coordination mechanisms for working together and sharing knowledge, both in person and through electronic media. Software-based experiments, including the system we describe in Chapter 6, can go beyond treatment effects and use fine-grained data to answer questions about *why* such effects exist, and inspire and inform new models of collective behavior.

Experimentally generated behavioral data has several advantages over the common approach of using data mining to find appropriate examples from the real world. Data mining can be problematic for finding both the *right* data and *counterfactual* data. boyd and Crawford (2012) argue that large datasets are limited to what is available, often hide details apparently in smaller datasets, and inevitably force one to discard lower-level details during use. On the other hand, experiments can produce *designed* datasets (Salganik 2014) of medium size consisting of the particular data of interest and under conditions that may not exist in the real world. When it is difficult to model novel behavioral paradigms directly,

experiments provide a way to gather data, compare context-specific effects, and build a foundation for theoretical work. This in turn leads to more informed experimental studies that validate new theory.

2.3 Conclusion

In summary, experimental methods are an important complement to the common existing approaches of data mining and modeling to draw causal conclusions about how system interventions will affect user behavior. This understanding is not only useful for the design of any particular social computing system, but also for better characterization of human behavior in general. The following chapters describe experiments using novel methods that improve our understanding of collective social behavior.

Chapter 3

Volunteer and Paid Crowdsourcing: From Galaxy Zoo and Planet Hunters to Amazon Mechanical Turk

3.1 Preliminaries

Over the last decade, crowdsourcing has emerged as an efficient way to harness human intelligence for solving a wide range of tasks. While some crowdsourcing is unstructured and organic, such as efforts to coalesce knowledge on topics in Wikipedia and software applications created by open source projects, several crowdsourcing systems provide a structured environment that connects participants or *workers* with microtasks that are well-defined and self-contained. These systems typically do not require workers to be experts or to have strong familiarity with a task before starting to contribute.

Sections 2.1.1 and 2.1.2 discuss the myriad incentives from unpaid or volunteer crowdsourcing to paid crowdsourcing in online labor markets. Volunteers in unpaid crowdsourcing systems are driven by different motivations than workers of paid crowdsourcing platforms;

volunteer crowd workers seek different objectives, and some may be more knowledgeable about a specific task than most workers in paid systems.

The many differences in motivation and incentives between paid and unpaid crowd work are not yet well understood, and this chapter presents two studies that shed light on a better understanding of user engagement and accuracy in the two types of systems.

3.1.1 Engagement and Attention in Volunteer Crowdsourcing

There is significant evidence that attention and fame drive some kinds of participation in social computing systems, and may even be predicted with modeling techniques. Huberman et al. (2009) show that video upload activity on YouTube strongly depends on the number of views of previous videos. Kittur and Kraut (2008) find that the quality of articles on Wikipedia critically depends on the activity of numerous editors and their method of coordination. Cosley et al. (2006) describe how online communities can produce quality contributions, and gives several examples where various methods of mediating worker contributions have succeeded and failed. Beyond efforts in crowdsourcing, there have been several methodologically related studies of user attention in the context of web browsing. These efforts include work by Adar et al. (2008) that examines patterns of browsing across web browsing sessions and by Sculley et al. (2009) that explores supervised learning for making predictions about individual behavior on the web.

Beyond simple aggregation of user input, there have emerged principled approaches to guiding crowdsourcing using decision-theoretic methods. The CrowdSynth project by Kamar et al. (2012) introduces a decision-theoretic methodology for reducing volunteer effort while maintaining accuracy by integrating the efforts of machine vision with human perception and computing the value of acquiring additional information from workers. Efforts on TurKontrol (Dai et al. 2010b; 2011) provide mechanisms for choosing different workflows in a crowdsourcing system. In addition to analyses of worker quality, research on the behavior of workers in crowdsourcing platforms include observational studies on task prices, task completion time, worker availability (Ipeirotis 2010), worker incentives (Kaufmann et al. 2011),

and on implicit and explicit motivations of workers (Rogstadius et al. 2011). Understanding, sustaining and improving worker engagement has been mentioned as a future challenge for the crowdsourcing community (Kittur et al. 2013).

Section 3.2 explores the challenge of learning from data to predict signals of the attention and effort that workers allocate to tasks. Such models for estimating the time and effort invested by workers are useful for understanding worker behavior and improving existing systems. For instance, predictions about engagement can help explain the influence of different interaction designs on user attention and effort at points within and across sessions of crowd work. Studies of engagement could reveal patterns of engagement for different groups of users, predict users' disengagement, direct the assignment of task sequences to volunteers so as to enhance interest, effort and attention, and measure and influence the likelihood that users will return to continue volunteer efforts at a later time. The ability to predict forthcoming disengagement of individual workers would allow systems to make targeted *interventions*, such as providing especially interesting tasks to workers at risk of becoming bored, directing support to struggling new workers, helping with the timing of special auxiliary materials or rewards, and encouraging workers to return in the long run. Data collection and modeling of engagement is also promising for the comparative study of different designs such task structures or workflows (Kulkarni et al. 2012, Lin et al. 2012), programs such as achievement-based badges that provide different intrinsic incentives (Anderson et al. 2013), and their influence on different types of workers.

3.1.2 Effects of Payment Schemes in Crowdsourcing

Sections 2.1.1 and 2.1.2 outlines the plethora of incentives that must be considered in understanding the motivations of crowd workers. In online labor markets, perhaps the most important effects are those of financial incentives. However, there is evidence that straightforward economic rational agent models may not be sufficient to capture effects of the micro-payments used in paid online crowdsourcing.

Mason and Watts (2009) examined financial rewards for two tasks, where workers were paid a fixed payment for each task completed and had the option of continuing to work on more tasks. They found that workers completed more tasks for a higher fixed payment, but that quality did not improve. Rogstadius et al. (2011) made a similar observation in their experiments. Yin et al. (2013) found that, while the magnitude of performance-contingent payments alone did not influence the quality of work produced, the change in the payment level for tasks in the same session did—increasing and decreasing payments increased and decreased the quality of work, respectively. Ho et al. (2015) demonstrated that task-specific features can significantly affect the results of financial incentives. A large literature in economics and social psychology explores the relationships between the magnitude of financial compensation and productivity. We refer interested readers to a comprehensive review and meta-analysis by Camerer and Hogarth (1999).

Less attention has been focused on the influence of different payment schemes on the quality and quantity of work produced. Mason and Watts (2009) experimentally compared piece-rate schemes, where workers are paid for each task, and quota-based payment schemes, where workers are paid only after completing a bundle of tasks. They found that the quota-based scheme elicited higher effort from workers, while workers completed fewer tasks under the piece-rate scheme. Shaw et al. (2011) compared 14 financial, social, and hybrid incentive schemes, including performance-contingent reward and penalty, in their MTurk experiments. They identified two schemes where higher-quality work is produced in situations where workers' payments depend on the responses of her peers. Prior work in economics explored the influence of providing piecewise payments versus an hourly wage. In a comprehensive study of the Safelite Glass Corporation (Lazear 2000), where workers install glass windshields in automobiles, a switch from hourly wages to piece-rate pay resulted in the firm becoming 44% more productive and workers earning higher wages overall. These results were obtained under an intrinsic policy that discouraged the temptation to do low-quality piece-rate work.

Section 3.3 characterizes how different types of financial incentives influence the behavior of paid workers relative to volunteers. These differences are especially interesting with regard

to the influence of incentives on the performing of tasks that are ambiguous or difficult, as different financial incentives may influence the amount of time workers spend on tasks and the quality of work performed. If workers are motivated solely by monetary compensation on a platform with no quality control, economic theory predicts that they will shirk and produce work of minimally acceptable quality. For example, the method currently used in paid crowdsourcing markets is to pay for each task, and this may naturally cause workers to complete tasks as fast as possible at the potential expense of accuracy. Even if workers exert a good-faith effort, the method of payment may still influence their work, as the requester and even workers themselves may not be explicitly aware of the way their work is influenced by financial incentives.

3.2 Predicting Engagement in Volunteer Crowdsourcing

We construct predictive models of worker engagement from large-scale usage data collected from a crowdsourcing platform. We focus on predicting that a volunteer worker will disengage within a given number of tasks or minutes, based on data about volunteers' characteristics and activities logged in histories of interaction and sensed in real time. We focus our studies on a citizen science platform called Galaxy Zoo (Lintott et al. 2008). Using supervised learning, we learn models for predicting worker engagement and evaluate them on data collected from Galaxy Zoo. The results demonstrate that learned models can successfully identify workers that are soon to disengage. We study various notions of engagement and compare the importance of different factors in accurately predicting worker engagement. Finally, given that real-world crowdsourcing systems accumulate data continuously over their lifetimes, we evaluate the amount of data and retraining needed to learn such models accurately. These studies help with understanding the factors that influence workers' engagement and provide insights about deploying predictive models in real crowdsourcing systems.

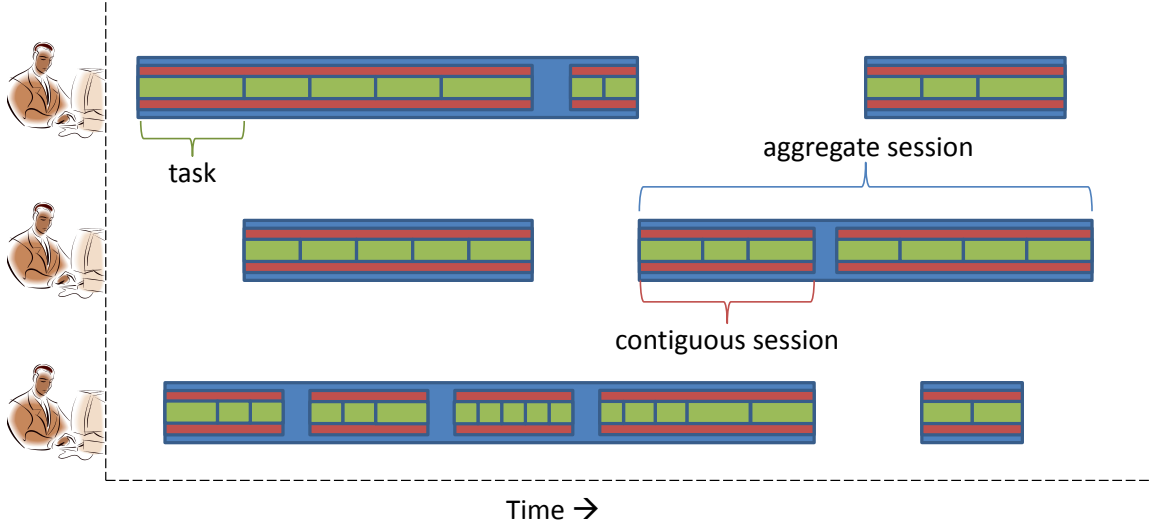


Figure 3.1: Model of worker sessions in crowdsourcing.

3.2.1 Data, Outcomes, and Model

We consider crowdsourcing settings where workers complete a series of *tasks* over time. These settings can include tasks on paid crowdsourcing platforms or volunteer efforts such as commonly seen with citizen science tasks. A task is the smallest indivisible unit of work that can be completed, e.g., a single classification in a citizen science system or a human intelligence task (HIT) on Amazon Mechanical Turk (MTurk). We consider sessions of a worker on a crowdsourcing platform to be the periods of time that workers spend engaged with the platform. Workers complete multiple tasks over the course of a task-centric *session*. The progress of a worker can be interrupted for various reasons. Short-lived demands for attention such as bathroom breaks or brief conversations divide a sequence of contiguous tasks into *contiguous sessions* of uninterrupted work, divided by short breaks where workers intend to return to the task. Workers can also decide to stop working for longer periods of time or end their work for a variety of reasons; these longer pauses in activity divide the activity into *aggregate sessions*, comprised of one or more contiguous sessions.

Contiguous and aggregate sessions may have different properties in terms of the engagement of a worker. Workers are likely to maintain the cognitive context of previous tasks for contiguous sessions that start soon after the end of the prior session. Workers starting a new

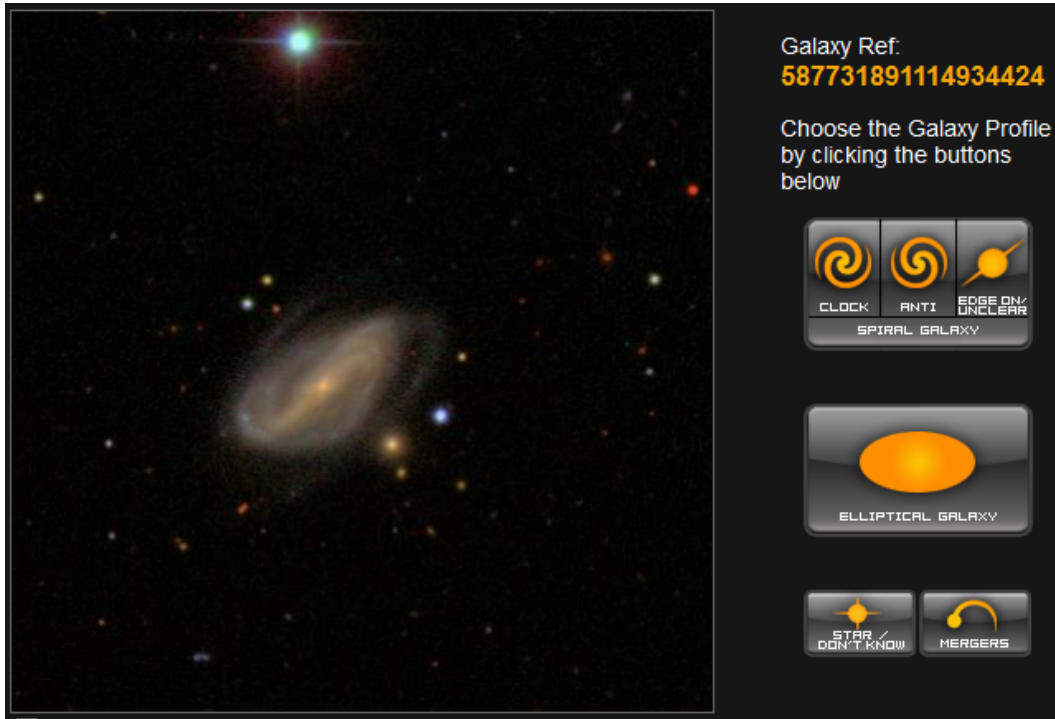


Figure 3.2: The Galaxy Zoo 1 classification interface.

session after the end of an aggregate session can be assumed to return without such mental context. Because engagement within contiguous and aggregate sessions may have different implications for the crowdsourcing platform, we study them separately.

Figure 3.1 shows a visual representation of worker activity over time under these session definitions. Each inner segment (green) represents a task. Workers may complete tasks at different rates and the width of the segment is the length of time used to complete the task. Groups of tasks divided by brief interruptions comprise contiguous sessions (red). A sequence of one or more contiguous sessions defines an aggregate session (blue). As shown in the figure, individual workers may differ in terms of the frequency of their contiguous and aggregate sessions, the amount of time they spend in sessions, and the number of tasks they perform.

3.2.2 Galaxy Zoo as Testbed

Galaxy Zoo (Lintott et al. 2008) is a citizen science project that began in 2007, harnessing

the power of many to classify images of galaxies from the Sloan Digital Sky Survey (SDSS) via the internet. Volunteer citizen scientists (workers) engaging with Galaxy Zoo are asked to evaluate the morphologies of galaxies in the survey. To date, volunteers have examined nearly a million SDSS images. Currently in its fourth iteration, Galaxy Zoo is one of the longest running, most publicized, and most established examples of an unpaid, volunteer crowdsourcing system.

We study data about task completion from the first version of Galaxy Zoo. In that version, workers are shown a picture of a celestial object, and press one of six buttons to classify the object into categories such as an elliptical galaxy, spiral galaxy, or other type of object (See Figure 5.5). The dataset collected from the Galaxy Zoo system enables a large-scale study of engagement of workers in crowdsourcing platforms. The dataset includes 34 million votes collected from 100,000 participants about 886,000 galaxies.

Worker Behavior

The tasks and associated patterns of interaction on Galaxy Zoo are nicely captured by the representation of contiguous and aggregate tasks: each new effort at completing a classification represents the completion of a new task, which can be as short-lived as a couple of seconds. Workers complete many tasks over time, represented as one or more sessions of work divided by breaks. Some workers spend a great deal of time on the site; one worker classified nearly 12,000 galaxies in a single session, while another spent more than 17 hours making contributions. In both of these cases, no break was longer than 30 minutes.

We define the end of a contiguous session as a break of more than 5 minutes, since it is unlikely for a worker to spend this amount of time on a Galaxy Zoo task without activity. With this definition of disengagement for contiguous sessions, the average amount of time spent on each Galaxy Zoo task is 9 seconds with a standard deviation of 21 seconds.

To define a disengagement criteria for aggregate sessions, we study the distribution of the time it takes for workers to return to the platform after disengaging for more than 5 minutes (end of a contiguous session). Figure 3.3 shows the cumulative distribution of time between

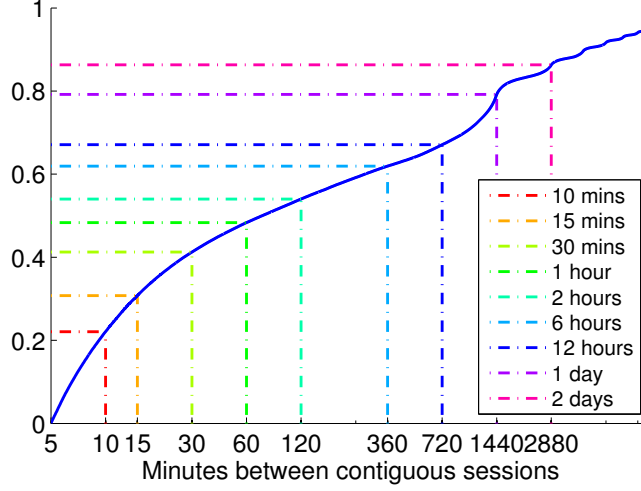


Figure 3.3: Cumulative distribution of inter-session times on Galaxy Zoo.

tasks when pauses are greater than 5 minutes. As displayed in the figure, many contiguous sessions are followed with a new session from the same worker in a few hours. Indeed, 41% of workers start a new contiguous session within 30 minutes of the end of their previous session. Since, 30 minutes may still preserve the context of the earlier contiguous session, we admit adjacent contiguous sessions that are less than 30 minutes apart into a single aggregate session. In the rest of the paper, we use breaks of lengths **5** and **30** minutes as the definitions of disengagement from contiguous and aggregate sessions, respectively. In Galaxy Zoo, a worker completes an average of 81 tasks ($\sigma = 146$) and spends on average 693 seconds ($\sigma = 938$) in a contiguous session. On average workers complete 135 tasks ($\sigma = 233$) within an aggregate session and the average time spent is 1629 seconds ($\sigma = 2282$). These composites of task completion and engagement times naturally resemble power law distributions.

The interval distribution shown in Figure 3.3 shows several aspects of the engagement behaviors of workers. Although the distribution reveals a power-law taper, visible jumps appear in the distribution at around one day, with smaller jumps at two days, three days, etc. This suggests that the longer breaks between worker sessions are not smoothly distributed, which suggests that workers have strong patterns of engagement. Indeed, for some noticeable fraction of workers, there is a high probability of returning at the same time each day—these

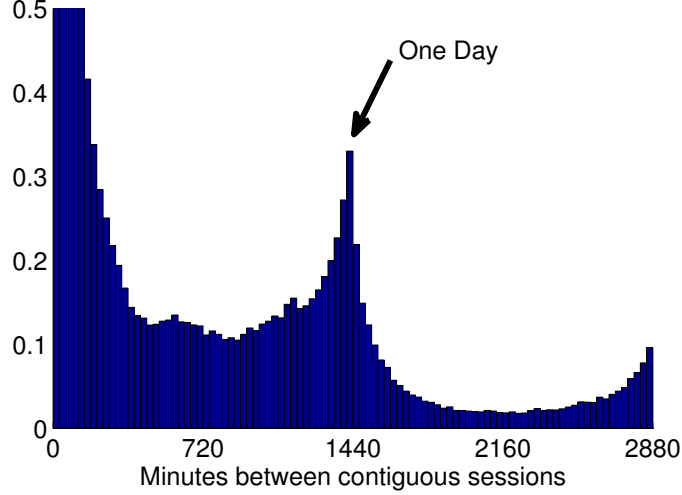


Figure 3.4: The distribution of inter-session times on Galaxy Zoo over a two-day period.

workers have a relatively predictable schedule for investing time. This trend is more visible in Figure 3.4, which displays a histogram of such times for periods up to two days, using a linear scale on the time axis. Much of the mass is concentrated in the period of several hours shortly after completing a task. However, the exponential decay of return rate after completing the task is interrupted by a jump leading up to the one-day mark. If the worker does not return within one day, the distribution is similar for the second day. However, the marginal probability of returning is much lower for returns on later days. We shall now turn to analyses using machine learning and inference.

Instance Generation

We model the problem of predicting worker engagement as a binary classification problem. Each interaction of a worker with a task becomes an instance that is considered as a case in a library of events for training and testing predictive models. We define for each instance a set of features describing the state of the worker’s interaction (including the worker’s historical behavior). The label on the outcome for each instance is assigned by observing the state of the worker’s engagement. We define the prediction challenges as follows:

Given the current state of a worker’s session, will the worker stop participating

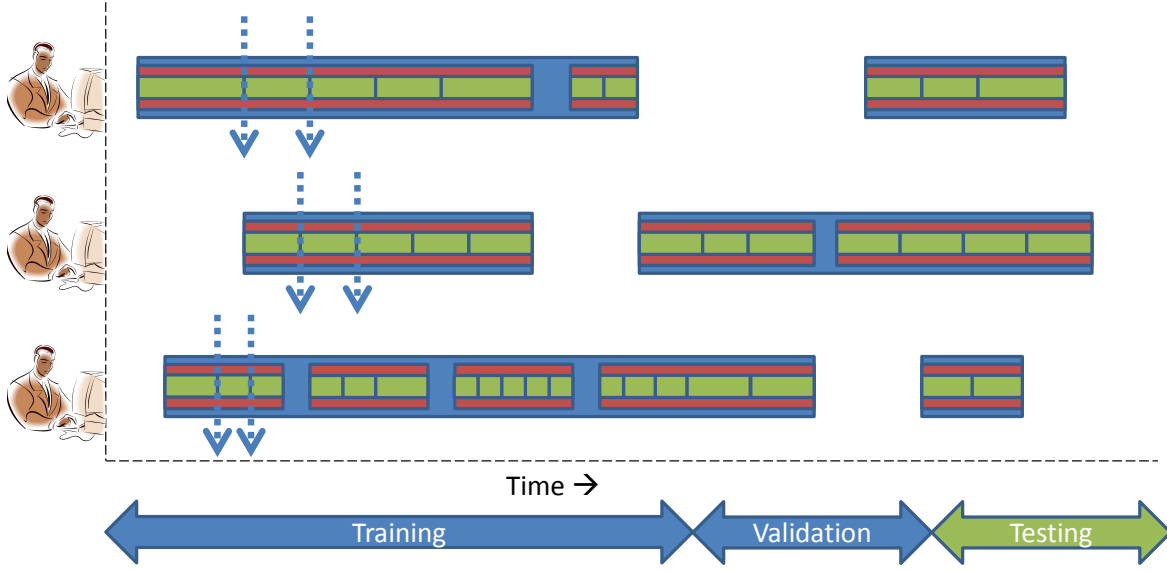


Figure 3.5: Instances created as cases for learning from Galaxy Zoo dataset. An instance is created for each task that each worker has completed, including features describing the historical behavior of the worker and the system.

*within a given **number of tasks** or **minutes of time**?*

If the condition defined above holds for the current state of a worker, the corresponding instance is assigned a positive label for either the time-based or task-completion versions of the prediction challenge. Otherwise, the instance is assigned a negative label.

Figure 3.5 shows a graphical depiction of the definition of instances. When each worker finishes a task, we create a data instance capturing the state of the worker’s session, a set of features about the worker’s recent and past activities (described in more detail below), and a corresponding label about engagement. The dataset consists of all worker-task interactions. We use different portions of this data to train, validate, and test learned models for predicting engagement.

The data extraction and analysis is complicated by the temporal nature of the activities and outcomes. Since each instance includes historical information about the behavior of a worker and the system, careless ordering of the data could lead to potential inconsistencies where instances in the training set contain information about instances in the test set. To avoid this potential confounding, we select training and validation instances that come strictly

before test instances, as shown in Figure 3.5. This methodology mimics how predictions would be used in practice: training and validation instances would be constructed from existing data, and predictions would be used to make decisions about subsequent worker sessions at run time.

Labels

In the context of the instances above, we can employ several labels on outcomes that describe disengagement over time. We may be interested in how soon a worker will stop working, or how many more tasks they will perform. Some outcomes may be easy to predict. Other outcomes may be less easy to predict but make for more valuable predictions.

The challenge of predicting whether a worker will stop working in the next 30 seconds is significantly different from the challenge of predicting whether the worker will stop within 30 minutes. These predictions would likely be used in different ways in the operation of a crowdsourcing system. The former outcome has very few positive instances, and training a classifier for such biased data sets can be challenging. We focus on binary outcomes on disengagement—on the challenge of predicting whether the worker’s session will end within a given amount of time or number of tasks, and report our findings in the following section.

Features

As shown in Figure 3.4, workers may have strong patterns of engagement, including recurrent activities with time-of-day resonances. Such patterns of effort can inform the construction of features. We formulate features and group them under three general categories.

Task-Based Features. Without considering behavioral changes over time, workers may be affected simply by the tasks that they observe; this assumption underpins many worker/task latent-feature models in crowdsourcing (see Raykar et al. 2010 for an example). The Galaxy Zoo team shared with us anecdotal evidence suggesting that workers tend to have a longer session if the first few galaxies they see are interesting, high-quality pictures, rather than the more common less interesting or low-quality galaxy images. We can evaluate this objectively

by computing features that capture worker behaviors in response to sequences of difficult or banal tasks, based on the activity of other workers. These features include those based on use of an estimate of the running difficulty of the last X tasks, computed by considering differences in votes on objects by others. We summarize differences in votes on objects via computing the entropy of a set of answers.

Session Features. We also consider attributes that characterize workers’ activities within the current session. We consider statistics around the number of tasks completed in the current session versus completed in typical sessions for each worker. We also compute statistics about the *dwelt time*, capturing the amount of time spent on each task, and the worker’s *vote entropy*, which represents the diversity of workers’ classifications. We believed these statistics could serve as signals of a worker’s attention to tasks at hand. For example, a running average of dwell time as compared to the average dwell for sessions can measure whether the worker is starting to pay less attention or struggling on a given task. Similarly, a worker providing a set of votes with low vote entropy on a spectrum of randomly sorted tasks may be selecting the same classification for many tasks in the absence of deep analysis, and thus paying less attention than someone who is providing input that is better matched to the distribution of cases. All of the features mentioned can be derived from behavior in the current session regardless of the worker’s histories or habits. We compute these features for both contiguous and aggregate sessions as the characteristics may be different.

Worker Features. We can also compute multiple features that characterize workers based on their history and habits. Such features are a rich source of information for learning to predict future engagement of individual workers. These features include the following classes:

- **Summary features.** These features include the typical time spent on tasks, number of past sessions, and average time spent on sessions. These features implicitly distinguish among segments of the worker population.

- **Start-/end-time features.** Features build on periods of time when workers engage with the system, including comparison of the current time of day to the typical time of day that the worker has started or ended a session in the past.
- **Session history features.** Features describing the worker’s behavior in aggregate sessions, including the number of short breaks that are taken and the length of contiguous sessions.
- **Inter-session features.** These features capture information about the period of time (gap) since the worker’s last session and how this compares with past gaps.
- **Dwell time features.** Features on the amount of time that the worker spends on tasks, including consideration of statistics of different running averages compared to previous computed averages over the worker’s history.
- **Session task features.** These features include a set of compound features that compare the worker’s task-related statistics on the current session with the mean statistics observed on past sessions, including number of tasks completed and amount of time spent.

We compute statistics on features for the complete history of a worker and also for the most recent history (i.e., last 10 sessions) to identify behavioral changes. The worker features also implicitly include session features, as they compare a worker’s current session to longer histories of behavior. Overall, we computed nearly 150 features for our data instances.

The features that we compute do not explicitly encode domain-specific knowledge about the specific task of classifying galaxies. For example, no feature depends on the results of any automated classification of galaxies, or a prior distribution of the types of galaxies. While using domain-specific features may improve predictive performance, we focused on how well we can detect worker engagement using features that are applicable to numerous other types of tasks. We believe that the methods can be generalized to similar crowd work settings.

3.2.3 Evaluation

We seek the construction of statistical models that can predict that a worker will disengage within some horizon of time or tasks. We generate our datasets for experiments on these predictions from the three months of Galaxy Zoo data using the methodology described in the earlier section. We remove from consideration workers for whom we observed little activity (less than 10 contiguous sessions). The generated data set consists of over 24 million instances, corresponding to each task that was performed by the set of workers that we considered. For each experiment, we randomly sample 500,000 training instances, 250,000 validation instances, and 250,000 test instances, preserving temporal consistency per above. This approach ensures that all of the methods use the same amount of data when possible. Unless otherwise noted, we used the complete set of features in the experiments.

Predicting the instances described below typically results in biased data sets, containing very few positive instances where users disengage. As a result, we consider the measure of area under the receiver-operator characteristic curve (AUC) to evaluate the relative performance of different classification algorithms. The AUC measure can be interpreted as the likelihood that a classifier will distinguish a randomly selected positive instance from a randomly selected negative instance. A random classifier that assigns each instance the prior probability of the dataset has an AUC of 0.5, and a classifier that can always distinguish positive from negative instances has an AUC of 1.0. The AUC is invariant to the prior distribution of labels in different datasets, which can be highly skewed. We additionally measure the log-loss reduction (LLR) achieved by each classifier as compared to the random classifier as a measure of the accuracy of probabilities assigned to each instance by the model. Higher positive log-loss reduction values predict more accurate probability estimates. In all of our experiments, classifiers with higher AUC values showed higher log-loss reduction metrics. For simplicity of presentation, we report AUC values for the experiments, since they provide a robust metric for comparing the success of predicting different outcomes.

For each classification task, we performed a validation phase to learn the best classifier. We explore the predictive power of models constructed with boosted decision trees, linear

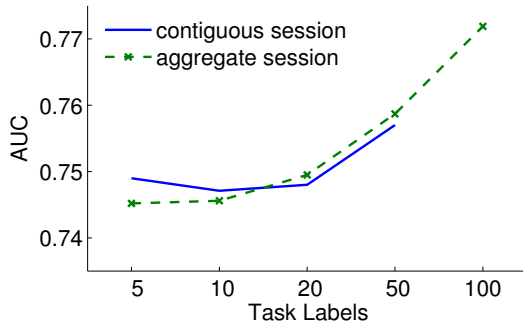
SVM, and logistic regression for predicting the outcomes described below. For each procedure, we normalized the data and performed parameter sweeps using the validation set. We created a single best classifier for each task by identifying the procedure and parameters that performed the best on the validation data. We report the results of the final models on the test set below. In our experiments, boosted decision trees consistently outperformed SVM and logistic regression on the validation set, and thus was used to train all of the final classification models.

Outcomes of Interest

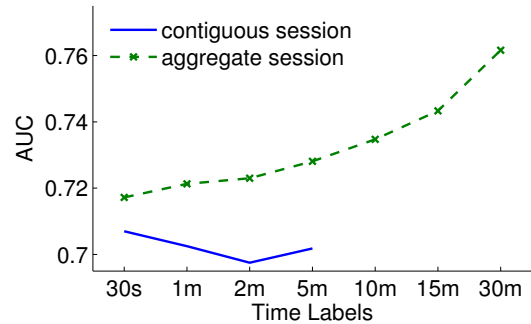
For each instance in our dataset, we are defining outcomes according to the definition below:

Does the worker’s current (**contiguous** / **aggregate**) session end within (X **tasks** / Y **minutes**)?

For example, if the particular outcome of interest is whether the aggregate session ends in 20 tasks, then a positive instance indicates that the worker will stop within the next 20 tasks and that they will not return for at least 30 minutes.



(a) Predicting outcomes defined in terms of number of tasks.



(b) Predicting outcomes defined in terms of time.

Figure 3.6: Prediction performance with different outcomes, using all features.

This definition is quite general; it includes definitions of disengagement outcomes based on different session definitions. The closeness to disengagement can be defined based on the number of tasks or the amount of time, and the degree of closeness can vary with different

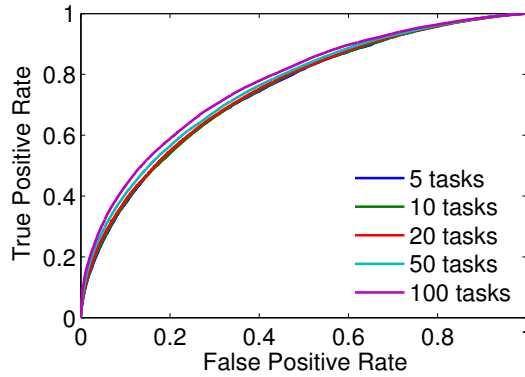


Figure 3.7: ROC curves of different models predicting the end of an aggregate session by number of tasks.

X or Y values. While some outcomes may be more easily predictable than others, specific predictions may be particularly useful for a domain in guiding decisions, such as interventions aimed at increasing effort allocated by volunteers. Designs for valuable target predictions and best uses of inferences will typically be domain-centric exercises. Given the range of potential uses of predictions about disengagement, we do experiments over a spectrum of outcomes.

Figure 3.6 shows the performance of predictions for different target outcomes, as indicated on the x -axis of the graphs. Generally, we can better predict the end of an aggregate session (where the worker does not return for at least 30 minutes) than the end of a contiguous session (the worker does not return for at least 5 minutes), especially in terms of time. As might be expected, we can better predict whether a session will end within a larger number of tasks or longer period of time than within a small period. Figure 3.7 shows the ROC curves for predicting the end of an aggregate session by number of tasks. The AUC monotonically increases as we increase the number of tasks.

Despite these trends, the analyses show that extreme differences do not exist among the predictability of different outcomes. We shall focus on the example of predicting the outcome that a worker will quit an aggregate session within 20 tasks since the target outcome is far enough ahead to allow for the execution of targeted interventions.

To put the performance of the predictive model in a usage context, consider the following: suppose that we seek to guide interventions based on identifying workers who will leave an

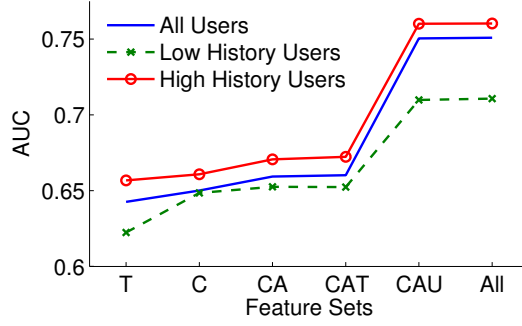


Figure 3.8: Model trained with different subsets of features and tested on different subpopulations.

aggregate session within 20 tasks. Using random targeting, only 14% of interventions would reach our target group. However, using the predictions on outcome to target the top 0.1% of workers likely to leave, 85% would reach our target group, a significant increase from the 14% made by random choice. This number would be 79%, 72%, 54%, and 44% by targeting the top 0.5%, 1%, 5% and 10% respectively, giving a tradeoff between accuracy and number of targets reached.

Feature Selection

The next set of experiments study which sets of features are most predictive of disengagement within a horizon. We wish to understand the accuracy of predictions as models are provided with larger sets of features. We are also interested in the relative influence of different feature sets on predicting disengagement for workers when we have small versus larger histories of interaction. For example, worker features may be more discriminatory when we have a great deal of historical information. We study the influence of quantity of historical data on prediction performance by sampling two additional test sets, consisting of data instances in the top and bottom quartiles of worker history activity by number of past aggregate sessions.

Figure 3.8 shows the prediction performance for small amounts of history history, large amounts of history, and for all workers for combinations of the following sets of features: task (T), contiguous session (C), aggregate session (A), and worker features (U). The results show that all feature sets individually help to predict worker engagement. However, adding

worker features with session features results in a large boost in prediction performance, and the benefit of including task features is diminished. We also see that workers with larger histories are more predictable even when the models do not include worker features. On the other hand, adding features describing past behavior produces less improvement (relative to the population at large) for workers with small amounts of history, as one would expect.

For the model trained with all of the features, the most predictive features, as measured by information gain in the boosted decision tree ensemble, are primarily about contiguous and aggregate sessions and the worker’s history. The most informative feature is the average number of tasks in recent (the last 10) aggregate sessions, followed by the number of tasks over the worker’s entire history, and over the last 10 contiguous sessions. Other informative features compare past behavior with recent behavior (e.g., difference of the average number of tasks done in an aggregate session in the entire history versus completed more recently) and features about the current session (e.g., average dwell time in the current session).

Worker Groups

Our results in the previous section suggest that the performance of predictive models depends on the specific worker subgroup at focus of attention. Hence, we consider whether we can gain a further advantage by training a prediction model for only specific subsets of workers. For example, we may be particularly interested in using targeted interventions to enhance the engagement of workers with small amounts of history so as to guide them early on to becoming more involved with a specific citizen science community.

Figure 3.9 shows the results for predicting the engagement of workers with small and large histories when models are trained with the data collected only from each class of workers. The results show that training for specific subsets of the workers does not improve the performance of predictions. These results suggest that, when there is a large amount of data available from a crowdsourcing system to generate instances and create features, a relatively complex classifier trained on the entire dataset may generalize well to specific subcategories of workers.

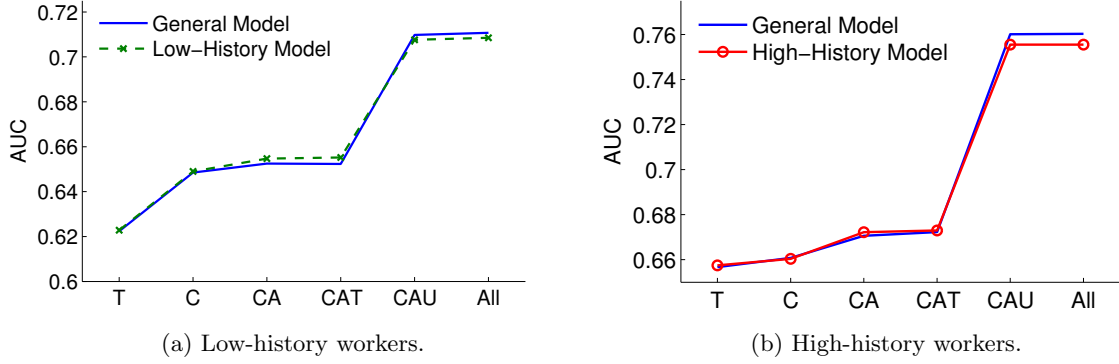


Figure 3.9: Comparison of the general model applied to a subpopulation with one trained explicitly on the subpopulation.

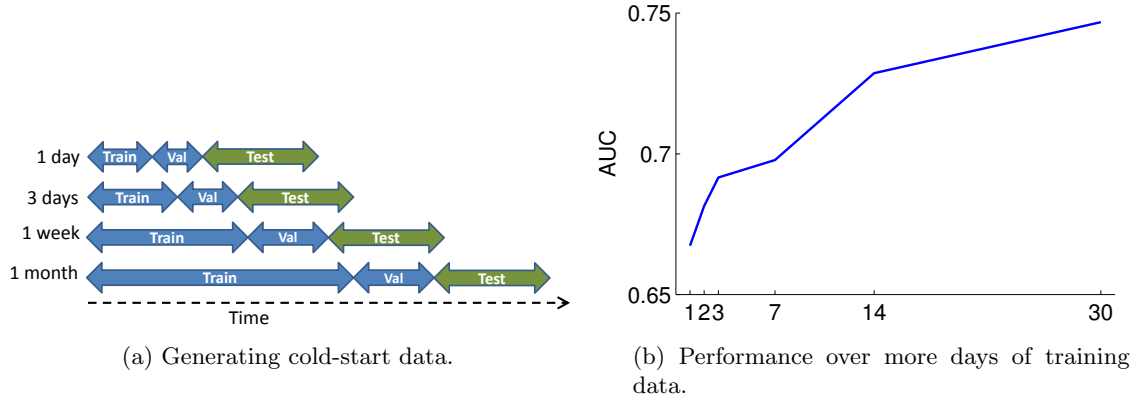


Figure 3.10: Testing prediction performance in a cold-start setting.

Cold-Start Performance

In a typical scenario, a crowdsourcing system begins soliciting contributions on a new type of task. At the outset of the use of the system, there is little data about workers and it may be difficult to make predictions about engagement when few workers have extensive histories. A best current model may not generalize well to future predictions as worker habits evolve and change. In our next set of experiments, we study the effect of the amount of data collected about workers on prediction performance.

Figure 3.10a demonstrates the approach for studying this “cold-start” problem. In each experiment, starting from the specific date when our dataset begins, we sample training and

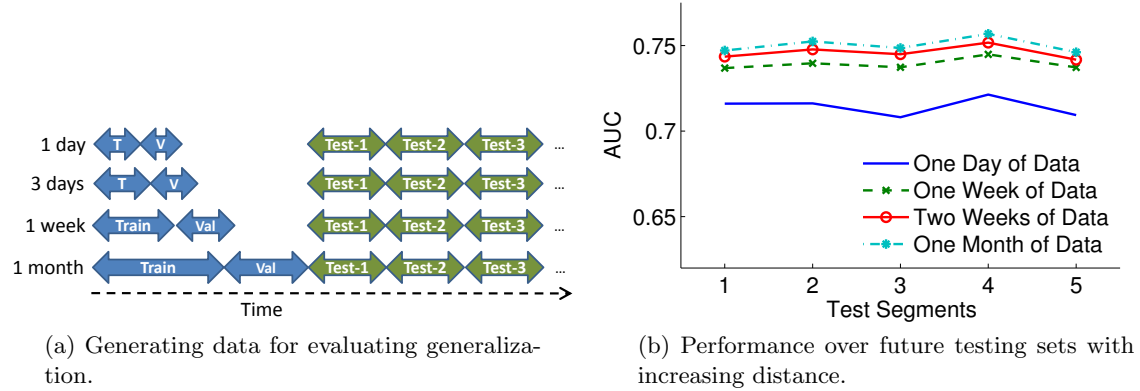


Figure 3.11: Testing generalization performance.

validation instances from the first day, first two days, first three days, first week, first two weeks, and first month of the system’s records. Except for the first day, the amount of training data sampled stays constant between experiments; however, the data set becomes more heterogeneous and represents more diversity among workers with the expansion of the training period. The test set is always sampled from the one-week period immediately after the training and validation sets. This formulation of sampling mimics the challenge of making predictions about engagement in real time as the test set is exactly the subsequent set of worker sessions appearing in the system.

Figure 3.10b displays the results of this set of experiments. The figure shows that the system needs to collect almost a month of data to reach AUC of 0.75—the accuracy of the model trained with the complete data. The performance of the models improves rapidly as data arrives during the first few days of the system’s life. This suggests that in early stages of deployment, it is generally important to continue training a prediction algorithm when more diverse information is collected about workers over time.

Model Generalization

Finally, we study how well a trained model generalizes for instances encountered at later times and whether the performance may diminish over time. Figure 3.11a shows the process for testing this problem. After training models using data sampled from the first day, first

week, first two weeks and first month of the system’s life, we evaluate the models on test sets for each two-week period following the first month of the system’s use. Figure 3.11b shows the performance of the models when tested on later time segments. The results show that all models generalize well to future instances and that the performance of the models does not diminish over time. They also confirm our earlier result that models trained with data sets (of equal size) containing observations about a more diverse population consistently perform better.

3.3 Comparison of Paid and Unpaid Crowdsourcing

In this section, we adapt an annotation task originally performed by volunteers in the Planet Hunters citizen science project to an experiment with paid crowd workers on MTurk. With this experiment, we aim to answer the following questions:

- *How does the performance of workers in paid crowdsourcing environments compare to that of volunteers in unpaid crowdsourcing?*
- *What differences are produced in terms of accuracy, types of errors, speed, and engagement by different financial incentives for workers being paid on a task?*

In a set of experiments, we observe workers completing a variable, self-determined number of tasks under one of three different financial payment schemes. While the actual payments in our experiments do not depend on the quality of work produced, we use a gold standard to evaluate the quality and accuracy of work produced. Because workers select the number of tasks to complete and how quickly to work, we can measure the effect of payments on speed and worker engagement in terms of total time spent and the number of tasks completed. Our results do not provide a universal answer to the questions we asked above for tasks of all types. However, we identify trends for the task that we study, and believe that the approach we use can be harnessed in the study of questions about the influence of requests and incentives on other tasks. Specifically, we find that

- With proper incentives, paid crowd workers can achieve comparable accuracy to volunteers working on the same task, and perhaps even work at a faster rate.
- Different payment schemes, while paying workers approximately the same amount, lead to significant differences in the quality of work produced and amount of time spent. Our results suggest that financial incentives can be used to control tradeoffs among accuracy, speed, and total effort within a fixed budget.

In addition to observations on worker accuracy and speed, the experiments provide insights about workers’ cognitive investment on paid crowdsourcing tasks. In particular, workers’ self-reports on reasons for quitting tasks bring into view aspects of the meta-environment of MTurk. We find via self-reports that a significant percentage of workers stop because they are concerned about the quality of their work—a notable contrast to the belief that workers are motivated purely by immediate monetary gains within paid markets. Overall, our results highlight the complex nature of paid crowdsourcing marketplaces and underscore the need for richer models of the relationships between incentives and worker behavior.

Sections 2.1.1 and 2.1.2 reviews prior work on financial incentives and motivation in paid and volunteer crowdsourcing. The motivation of volunteers in citizen science projects is much less studied; see Raddick et al. (2013) for one recent exception. Our work moves beyond the prior literature in several ways. First, we compare volunteer and paid workers on the same task. Second, we focus on the influence of different payment schemes within a comparable budget. Third, we provide evidence of secondary meta-incentives in paid crowdsourcing, using MTurk as an example.

3.3.1 Task Model

We consider a challenging citizen science task that, by its very nature, invites a high degree of variability in annotations. The task is analogous to finding needles in a sequence of haystacks. Each task can viewed as a haystack housing needles of interest. Workers examine the data and can mark needles directly, but the task is ambiguous because workers may miss certain needles or falsely mark other regions depending on varying levels of difficulty. By exerting

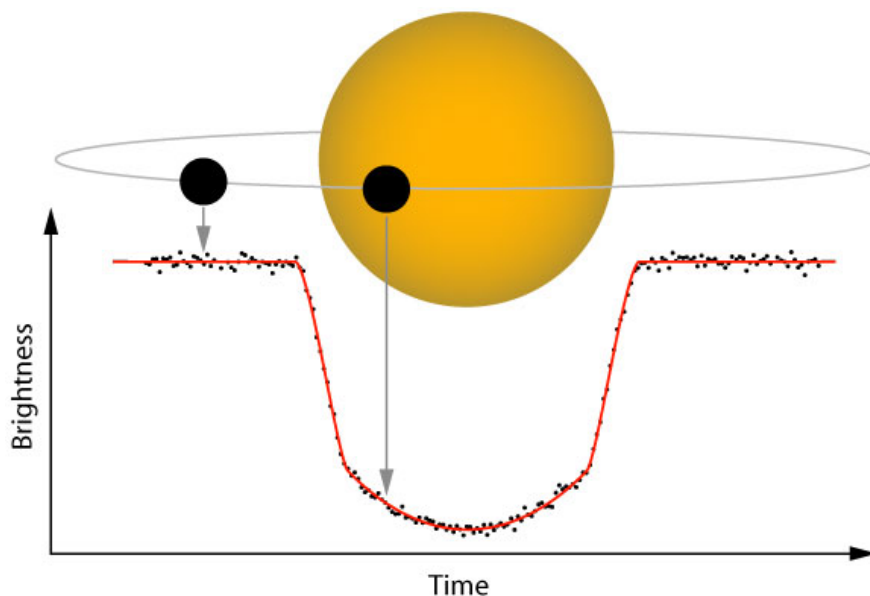


Figure 3.12: A light curve, showing the transit method of detecting exoplanets. Copyright John Johnson, used by permission.

more effort in a more detailed investigation, workers can generally obtain higher accuracy on this task. Workers may complete several tasks in sequence in a continuous session.

Many human computation tasks fall into this category, such as annotating events of interest in images (Salek et al. 2013). We find such task domains particularly interesting because the worker’s perception of the truth can be ambiguous: workers can produce both *false positives* when regions are wrongly marked and *false negatives* when objects of interest are missed, even if they are doing their best. Hence, the particular financial incentives at hand may influence the worker’s contribution and amplify the types of errors the worker makes. We now turn to the specifics of the haystacks and needles that we have studied.

3.3.2 Planet Hunters

Planet Hunters (Schwamb et al. 2012) is a citizen science project started in December 2010 with the goal of finding planets orbiting around distant stars (extrasolar planets or exoplanets), where volunteers search for the signatures of exoplanets in the data from the Kepler spacecraft (Borucki et al. 2010).

The Kepler spacecraft is a space-based telescope that simultaneously monitors the brightness of over 160,000 stars, producing graphs called *light curves* for each star. Kepler generates two data points per hour by measuring the brightness of a star approximately every 30 minutes. A planet that is orbiting the star in a plane aligned with the viewing angle of the telescope will partially obscure the star once per orbit in a *transit*, causing the observed brightness to drop and corresponding dip in the light curve (see Figure 3.12; (Winn 2010 and references within). Typical transits last from two to dozens of hours, so the telescope records multiple data points for a typical transit. The size of the dip in the light curve is proportional to the surface area of the star and the planet; the *relative transit depth*, or percentage decrease in the brightness of a star obscured by a planet during a transit, can be computed from the radius of the planet R_p and the star R_* :

$$\text{relative transit depth} = \frac{R_p^2}{R_*^2}. \quad (3.1)$$

For example, to a distant observer, Jupiter would obscure the sun by around 1%, while the Earth obscures only 0.01%.

Several aspects of the transit detection task affect its difficulty. Telescopes have a natural instrumentation error when measuring the brightness of a star. Moreover, the brightness of stars themselves vary over time, causing fluctuations and changes in the light curve (typically on timescales longer than transits). A transit with a small relative transit depth can be easily seen in a low-variability light curve while a transit with a large relative transit depth may be even hard to see in a highly variable light curve. A planet with short period (orbit time) compared to the span of observation can cause multiple transits to appear at regular intervals, making detection easier. In general, transits by fast-moving planets, by small planets, and in front of large stars are more difficult to detect.

Although the transit method has been in use by astronomers, the orbital telescope technology deployed in Kepler has allowed for searches of planets en masse. Planet Hunters enlists human volunteers to review Kepler data, and has resulted in several planet discoveries that were not detected by automated methods, demonstrating the value of human pattern

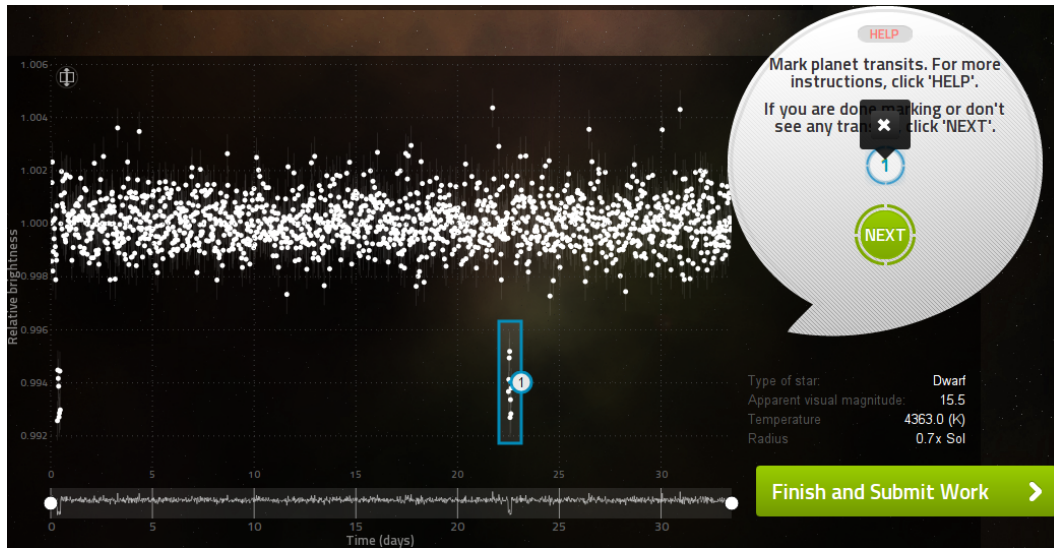


Figure 3.13: The experiment interface, showing a single annotated transit. The **Next** button accesses the next light curve, while the **Finish** button submits the task.

recognition for this task (Fischer et al. 2012, Lintott et al. 2013).

3.3.3 Experiment Design

Interface The interface for Planet Hunters is open-ended, allowing workers to freely examine a light curve of 1,600 data points collected through ~ 35 days with tools for zooming and drawing simple boxes around any potential transits that they see. We designed a similar interface for an MTurk task, shown in Figure 5.5. Workers can mark possible transits on a light curve by drawing a box, resizing and/or deleting them as desired. In this way, workers produce annotations in a continuous space, defined by the coordinates and size of the boxes.

After accepting our HIT and reading a short consent form, workers see an interactive tutorial of the interface, describing what a light curve is and how planet transits are detected. The tutorial demonstrates zoom controls and the annotation process. A help menu is available at any point during the task that provides additional information about identifying planet transits and the interface controls.

A key aspect of this experiment is that workers can annotate multiple light curves (finish multiple tasks), choosing the amount of effort they want to contribute before quitting. This

is similar to the process of performing a number of tasks for a particular requester on systems like MTurk. At each light curve, workers may choose to continue on to the next light curve or to finish their work and submit the task. We place some basic controls on the experiment such as limits on participation, described later in this section.

When workers choose to complete their work, they are required to complete a short survey containing questions about the HIT. We ask workers whether they think the HIT was easy, fun, or well paid, about their strategy on the task, and if they ran into any bugs. Most importantly, we ask workers why they decided to stop working and to submit the task and what, if anything, would have made them work longer.

Simulated Transits In contrast to most real-world crowdsourcing tasks, transit detection has the useful feature that realistic data with a ground truth can be generated easily; simulated transits of different depths and durations can be added to an observed light curve. The Planet Hunters team injected simulated transits into real Kepler data (Schwamb et al. 2012) to estimate the efficiency of detecting different types of planet transits using crowdsourcing.

While planets with multiple visible transits should be easier to detect in a light curve, Schwamb et al. (2012) showed that the difference in behavior in Planet Hunters is insignificant for orbital periods less than 15 days, and that transit depth is the dominant effect. Therefore, we define a difficulty measure for detecting transits for a simulated light curve by comparing the relative transit depth (Equation 3.1) to the noise of the light curve:

$$\text{difficulty} = \frac{\text{stdev}(\text{differences in adjacent points})}{\text{relative transit depth}} \quad (3.2)$$

A light curve simulation with difficulty 0.2 means that the depth of the transit will be 5 times bigger than the typical noise in the graph, and should be relatively easy to spot. A simulation with difficulty 1 means that a transit can easily hide within the noise in the graph, and is more easy to spot. Simulations of difficulties greater than 1 should be very difficult to detect. In the experiment described in the following section, we use simulated light curves that had been annotated earlier by volunteers on the Planet Hunters project.

Payment Schemes The primary goal of the experiment is to compare the effects of different payment schemes on workers’ performance on an ambiguous task at various levels of difficulty, and to performance on the same task by volunteers. We consider three different non-performance-contingent payment schemes:

- **Pay per task.** Workers are paid for each task that they complete; in our case, this is per light curve. This is a typical payment scheme for paid microtasks.
- **Pay for time.** Workers are paid for each unit time that they spend working, regardless of their actual output. This payment scheme is employed commonly in traditional employment.
- **Pay per annotation.** Workers are paid for each object that they annotate; in our case, this is per marked transit.

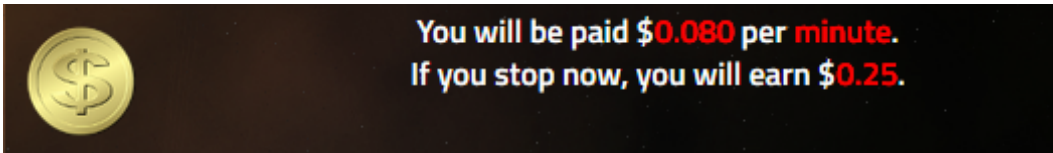


Figure 3.14: An sample payment message displayed to workers.

We focus on payment schemes that do not depend on the accuracy or quality of workers’ output. These payment schemes are simple to explain to workers and do not require the implementation of a quality control mechanism. To ensure that workers are fully aware of how they are getting paid, we show a continually updated banner at the top of the task (Figure 3.14) which displays the method of payment and how much they have earned so far.

Data Selection and Treatments To allow for comparison between unpaid volunteers and paid workers, we selected light curves for our experiment from the set of light curves that had been already reviewed by numerous volunteer citizen scientists contributing to Planet Hunters. All of the light curves were collected during Quarter 1 of the Kepler Mission. As transits are rarely seen overall, our dataset must include many light curves with no transits so that the task is realistic and workers don’t expect to see them in every light curve. However,

we also need to have sufficient simulations to obtain data about the accuracy of workers. Based on the original annotation results, we removed pathological light curves from the data that were particularly confusing, including planets periods shorter than 5 days. We ultimately selected a set of 250 light curves with up to 6 simulated transits and an additional 750 light curves without simulated transits (devoid of planet transits to the best of our knowledge). The simulated light curves are distributed approximately uniformly in the difficulty measure described in Equation 3.2, from values ranging from 0.2 to 1.0. We chose this range after visually inspecting many light curves, as the range included examples that were neither too obvious nor too difficult to detect.

We adopt notions of *precision* and *recall* from the information retrieval community to measure the accuracy of the annotations. An annotated box is counted as correctly marked if the center of the box is within an actual transit—this simple measure of correctness is convenient because it allows for some latitude in the width of a box drawn, which in turn depends on the zoom level. The precision of a worker’s annotations is the fraction of his annotations that are actual transits, or the ratio of the number of correct annotations to the total number of annotations. The recall of a worker’s annotations is the fraction of transits that are annotated by the worker, or the ratio of the number of transits correctly annotated to the total number of transits.

Controls and Monitoring Paying workers without regard to quality can lead to low quality output should workers behave as purely economic agents and expect no negative consequences for errors. As a result, we create controls that would be expected in a practical implementation.

- **Minimum of 5 seconds per light curve.** Without this control, a worker being paid by task can potentially click through light curves very quickly and be fully paid for almost no work.
- **Maximum of 8 annotations per light curve.** In absence of this control, a worker being paid per annotation may mark a potentially infinite number of false positives and

be paid for each one. We restrict the minimum orbital period in our data to be > 5 days, so at most 6 transits will appear.

- **Maximum of 3 minutes of inactivity.** Without this control, a worker being paid by time can potentially do nothing while earning wages. An inactivity warning is shown when a worker has done nothing for 2 minutes. If the worker continues to do nothing for a total of 3 minutes, the task ends and automatically redirects the worker to the exit survey.

We record all of the above events during a session. By monitoring inactivity and enforcing a timeout on the task, we are able to detect when a worker is no longer paying attention or has become distracted. As workers must complete the exit survey to submit the HIT, we can learn why they stopped working. We also restrict all worker sessions to a maximum of one hour or 200 light curves, to limit the amount of data from any one particular worker.

In addition to the detection of timeout, we track the amount of inactivity for each worker during their session, defined by the total amount of time that they were inactive for 30 seconds or more.

Hypotheses When worker payments do not depend on performance, workers would theoretically behave in extreme ways to maximize short-term payment. In theory, workers being paid by annotation would mark as many transits as possible (mostly incorrectly), earning the fixed amount for each. Workers being paid by task would click through the light curves very quickly, paying minimal attention to each one. And workers being paid by time might be expected to simply sit through a task and do barely anything, earning their hourly wage without spending much effort. However, we would not expect to see these extreme behaviors in practice. Workers typically expect that they will be evaluated in some way for their work, and many MTurk workers are keenly aware that rejected work will prevent them from doing lucrative tasks in the future. Aside from spammers, most workers will try to follow the instructions and do the task as well as they can understand it. Yet, the ambiguous nature of the task of identifying planets means that workers cannot be completely sure about the

‘wrong’ or ‘right’ answers; a worker being paid by annotation may subconsciously “see” more transits than a worker being paid by task, without being overtly dishonest. How strong might this psychological bias be?

The difficulty level of the task may also affect workers’ accuracy. When transits are plainly obvious in a light curve, we might expect all but the laziest workers to mark them. However, when transits are more ambiguous, we might expect workers who are paid per light curve or by time to more likely overlook them.

Most interestingly, the demographics of volunteer and paid workers are very different. Workers on Planet Hunters consist of many one-time users, but also include a dedicated community of users with an active discussion forum and many very motivated amateur astronomers, combing the data for transits and even writing their own analysis code. On the other hand, MTurk workers in our experiment do this task with nothing but a short tutorial, and are given a payment in return for their efforts. Given the differences in background and motivation, which group will do better?

Limitations of Comparison The focus of our experiment is to compare payment schemes, but we also give a comparison to volunteer work. There are some notable differences between our experiment and the original interface used by volunteers (see Schwamb et al. 2012 for a full description), which presents a series of additional questions to registered users. Our interface focuses only on transit annotation, and uses a free-drawing interaction and a different tutorial geared toward MTurk workers. Schwamb et al. show consistent behavior between the original box-placement annotation method used in Planet Hunters and a free-drawing method using a similar performance metric, but our measure of accuracy is more strict.

3.3.4 Results

We conducted our experiment as a between-subject study where all workers were allowed to do the HIT exactly once, to reduce the effect of noise in the results from worker experience over repeated tasks. In each set of experiments, workers were randomly assigned to one of the payment treatments. Workers were assigned a new, randomly selected light-curve each

Treatment	volunteer	\$0.0453/annotation	\$0.0557/task	\$0.08/minute
No. sessions	*	71	74	71
Mean hourly wage	*	\$10.993	\$8.056	\$4.800
Mean seconds/task	50*	29.13	24.89	27.45
Mean annotations/task	1.250	1.964	1.435	1.454

Table 3.1: Volunteer and worker behavior in the pilot. Annotations/task refers to the average number of annotations labeled per light curve. *In this table and Table 3.2, volunteers may work for longer due to possible additional questions in the task; we also omit statistics that would be misleading given the differences described previously.

time they continued in the task.

Initial Observations

To make meaningful comparisons among the treatments the wage across treatments must be comparable. Identifying comparable wages across schemes is tricky as we do not *a priori* know how workers would behave. Hence, we conducted a pilot experiment to observe the behavior of workers and obtain a better idea of what comparable wages would be.

Through our experience with tasks on MTurk and guidelines posted on various discussion forums, we observed that most experienced workers aimed at a target of \$0.10/minute or \$6.00/hr as a fairly reimbursed task for which they would continue to work indefinitely. We picked a lower wage of \$4.80, which is close to a fair payment for worker time but low enough that we could expect workers to quit our task (before the time limit) and thus obtain information about why they left.

To set wages for the various treatments, we examined the behavior of unpaid citizen scientists on the corresponding subset of the existing Planet Hunters data, obtaining a baseline of how many annotations volunteers would mark and the rate at which they completed the light curves. Using this data, we computed a wage of \$0.0557 per task and \$0.0453 per annotation, which would all pay the same wage of \$4.80 if the paid workers behaved similarly as the volunteers.

Table 3.1 shows a summary of observations from the pilot experiment. In the treatments shown, over 200 unique workers annotated about 14,000 light curves. Notably, paid workers

completed tasks significantly more quickly than the volunteer workers, resulting in a much higher wage for both the task and annotation treatments. Moreover, workers in the annotation treatment were much more eager about marking transits than the other workers, showing a clear bias. This further boosted their wage to an average close to \$11/hour; some workers were able to earn over \$30/hour, and we observed many comments on various worker forums that our task paid extremely well.

The non-uniform effective hourly wage earned across the treatments confirms that paid workers behave significantly differently from volunteers, both working faster and being influenced by their financial incentives significantly. However, the large discrepancy between wages makes it difficult to compare the payment methods, as some workers are earning more than twice as much as others. We also observed some notable meta-effects during this experiment. As we monitored worker discussion forums over the course of several days, we noticed that workers had begun to discuss our task and compared their payments with each other, being especially curious as to why some thought the task was particularly well-paid compared to others. On the site where the discussion was most lively (<http://www.mturkforum.com>), we talked to workers and discovered that while there was actually a policy against discussing research studies, our task actually appeared to be a normal MTurk task (as we had intended apart from the consent process), and the normal appearance had prompted the discussion. We were pleasantly surprised to learn that the Turker community had self-imposed rules to protect the integrity of research, and were advised to include an explicit statement not to discuss the task with others so as to be covered by this policy.

Balanced Payments

The observations on the pilot study prompted us to design a second round of experiments where workers are paid more equally, and to eliminate biases caused by external discussion. For example, workers might produce worse quality work if they expected a certain level of payment in the task from discussion but received a much lower amount.

We made the assumption, based on aforementioned work in financial incentives, that the

Treatment	volunteer	\$0.0197/annotation	\$0.0331/task	\$0.08/minute
No. sessions	*	118	121	117
Total tasks completed	*	4629	7334	5365
Mean hourly wage	*	\$4.802	\$5.580	\$4.800
Mean tasks/session	*	39.22	60.61	45.85
Mean seconds/task	50*	28.02	21.35	34.65
Median session time	*	9:04	15:08	18:05
Annotations/task	1.250	1.897	1.384	1.348
Precision	0.656	0.635	0.660	0.713
Recall	0.518	0.485	0.454	0.497
Time inactive	*	0.101	0.097	0.149

Table 3.2: Volunteer and worker behavior in the second experiment. Time inactive refers to the average fraction of time that a worker is detected inactive.

per-task behavior of workers would not change much compared to their payment level. Hence, we could scale the piece-rate wages for the annotation and task treatments accordingly, and obtain data where the effective hourly wage is closer to the target of \$4.80. While we could not enforce this in advance, the treatments would be comparable as long as they resulted in similar levels of payment. This resulted in piece-rate payments of \$0.0331 per light curve and \$0.0197 per annotation.

Second, we took several precautions to minimize external discussion about our task. We followed the advice of showing an explicit message not to discuss their work with others during the exit survey. We also posted on several discussion forums that participants should not discuss the task; we noticed that workers indeed passed on this message when others asked about this task. Moreover, since we required all workers to be unique, workers from the first set of experiments were not able to do the task, and this caused the amount of discussion to die down significantly. When closely monitoring discussions, we saw very few posts about our task during the experiment; workers also trickled in at a much slower rate compared to a veritable ‘flood’ of workers looking for a well-paid HIT in the first experiment.

Table 3.2 shows a summary of the second experiment. A total of 356 workers annotated over 17,000 light curves. Of particular note is that the effective hourly wage earned by workers was much closer together than in the previous treatment; the workers in the pay by annotation and by time treatments earned almost exactly the same amount, and the workers

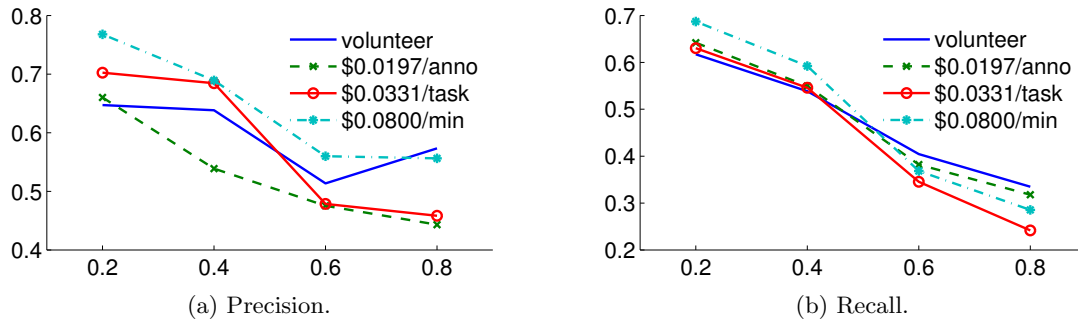


Figure 3.15: Accuracy by difficulty.

in the pay by task treatment, earned only slightly more.

Accuracy by Difficulty. We split the set of simulated light curves into four buckets determined by the difficulty measure in Equation 3.2, and computed precision and recall for each bucket, displayed in Figure 3.15. As expected, both precision and recall drop at higher levels of difficulty, with the only exception being the volunteer group at the hardest difficulty bucket. To test the significance of differences between each bucket, we used a two-sided paired t -test between the aggregate false positive rate and false negative rate among the light curves in each bucket.

We make several notable observations from the second experiment. With regard to precision, paying by time leads to significantly higher performance than paying by annotation at all levels of difficulty (for all but the last bucket, $p < 0.005$). Paying by annotation shows by far the worst precision across the board, with many differences being highly significant. We note that the precision across the volunteer population decreases more slowly as difficulty increases: at the easiest difficulty, they show significantly worse precision than the task and time treatments. However, for the most difficult tasks, they show significantly better precision than for the task and annotation treatments. We discuss possible reasons for this below.

For recall, workers paid by time show by far the best recall for easy tasks. However, the volunteers and workers paid by annotation show best recall at high levels of difficulty. Workers paid by task generally perform poorly, and in the most difficult bucket, they show the

worst recall by far ($p < 0.002$ compared to the unpaid and annotation treatments). Similar to the observation made in the precision analysis, overall, we observe that the recall scores of the volunteers are less sensitive to the difficulty level than the paid workers.

Worker Attention. We can measure the attention or interest of workers in two ways: by the amount of time they are spending on each task, a measure we believe roughly corresponds to effort; and the total amount of time in the session. This comparison is particularly interesting because workers are being paid roughly the same amount for their time, with the wage being almost identical for the annotation and time treatments. Table 3.2 shows that the financial incentive scheme implemented has significant influences on the speed of workers for completing tasks. When being paid by time, workers spend over 60% more time on each task than when being paid for each task, and this is accompanied by a corresponding increase in accuracy. These findings suggest that payment methods can be used to trade off speed and accuracy in worker output. In addition, workers spend less time on the task and show significantly worse precision when paid by annotation rather than by time, in spite of earning almost the same hourly wage. Figure 3.16 shows the distribution of statistics for sessions. The difference in number of tasks per session is significant for workers paid by task compared to the other two treatments at the 0.05 level. The difference in the total time spent in a session is also significant at the 0.05 level for the time versus annotation treatments. The differences in seconds per task is highly significant ($p < 0.0001$) for all treatments.

We also examine the reasons that workers gave for ending the task in the exit survey. There are many explanations for exiting a task. Horton and Chilton (2010) suggested that workers may set a target earnings level when deciding when to stop. In our experiment, workers could be interrupted or time out. Using the experiment controls as well as workers' stated reasons for stopping, we classified reasons for stopping into different categories, described with examples as follows:

- **quality** – concerned about submitting bad work or being rejected: *“I decided to stop because I wasn’t sure if I was doing a good job or not. I would have continued, but I*

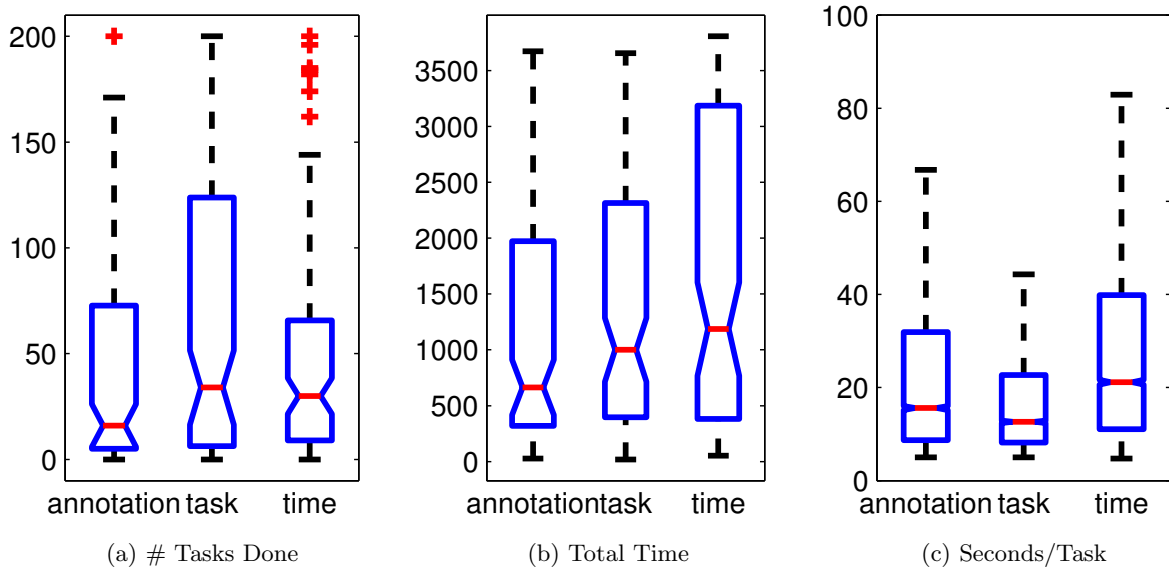


Figure 3.16: Distribution of session statistics. Boxplots show top and bottom quartiles and median as red line.

did not want my HIT to be rejected because I misunderstood or provided bad data. ”

- **limit** – reached a limit of 200 tasks or one hour.
- **exogenous** – had another prior commitment that had to be completed. Surprisingly, some employees Turk during their regular jobs: *“I had to go back to work...I would have worked longer if my lunch break was longer. ”*
- **interruption** – temporarily interrupted during the task, but intended to return to it. This included many bathroom breaks, phone calls, and pizza deliverymen arriving.
- **pay** – The pay for the task was too low.
- **bored / tired** – bored or tired of the task.
- **technical** – didn’t seem to understand the task or had a technical problem.
- **target** – reached a target self-imposed time or monetary amount: *“I decided that \$2.00 was enough for a single task and the amount of time spent on it. If I was paid much*

better I would have continued a bit longer; but I don't like doing a single task/hit for too long. ”

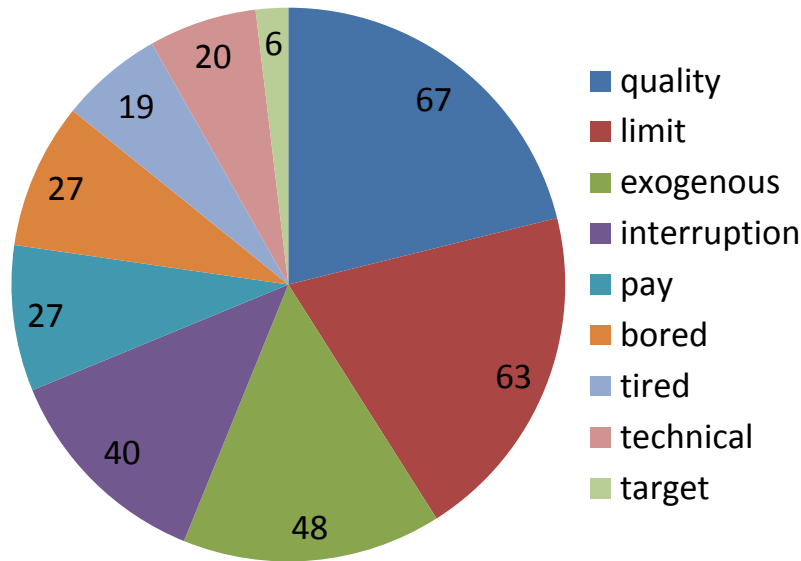


Figure 3.17: Classification of reasons for quitting.

Figure 3.17 shows that many workers were interrupted by distractions or outside commitments or reached our limit. Surprisingly, a significant proportion of workers chose to stop because they were unsure of the quality of their work. This runs counter to characterizations of Turkers as greedy workers who maximize their short-term rewards. To understand this phenomenon further, we analyzed workers’ comments carefully and communicated with them on discussion forums. It became clear that this behavior was founded in two goals. First, workers did not want to have their work rejected, which would waste their effort and lower their HIT approval rate (used as a filter on many tasks). Therefore, if workers are more uncertain about a requester’s approval policy, they would do less work to ‘test the water’. Second, some workers were actually concerned about providing good quality work to requesters and submitted our hit early or even returned it when they were unsure about their work. Very few workers explicitly mentioned a payment or time goal as a reason for stopping the task. As mentioned in Chandler et al. (2013), it is very important for researchers to be aware of

these meta-incentives when designing tasks and especially experiments for paid workers.

3.4 Discussion

In this chapter, we begin to close the gap in the understanding of motivation and engagement between paid and volunteer crowdsourcing. Section 3.2 presented the construction of predictive models of engagement in volunteer crowdsourcing, using data logged on the activity of citizen scientists using Galaxy Zoo. We performed several different experiments to probe characteristics of the prediction challenge. Our results demonstrate the performance of predictive models of engagement on a large-scale citizen-science platform. The trained models reached desirable performance with predicting forthcoming disengagement in different experimental conditions. Finally, we provide insights about the quantity of data needed to train models to perform well and how well the models generalize to making predictions about future instances.

We see numerous practical applications of predictive models of engagement. We expect that inferences about workers nearing disengagement can be employed in designs that use well-timed interventions to extend the engagement of workers. For example, it may be useful to target new workers who are about to leave a system by presenting a tutorial or a link to a discussion forum. Similarly, interventions may target workers who are struggling or losing interest by presenting more interesting tasks or by encouraging them with merit programs such as a badge programs (Anderson et al. 2013). If even only a small fraction of these workers respond to the interventions by staying and continuing to work, or returning to the platform with higher likelihood, then the platform can gain a significant benefit from predictions about disengagement.

We foresee opportunities for developing a variety of predictive models about engagement. For example, we may wish to predict if and when a worker will return after one or more sessions, based on multiple features, including traces of the worker’s history of experiences with the platform. Models of engagement can be expanded to make predictions about more general notions of worker engagement, attention and effort, and they can be applied to tasks

that go beyond of labeling. Beyond use on volunteer-centric tasks, we envision applications of models of engagement in paid systems. Such models may include distinctions and inferences about the joy or excitement associated with tasks, the link between intrinsic reward, payments, and effort, and leveraging of more detailed worker profiles, including demographic information and long-term histories of engagement. We hope that this work will stimulate further research on user attention, effort, and engagement in crowd work.

In Section 3.3, our experiments centering on challenging, ambiguous annotation tasks of varying levels of difficulty, provide a novel comparison of volunteer workers to workers paid by different financial schemes in an online task market. Under the tasks we studied, we find comparable performances between volunteers and appropriately paid workers. We note that the results obtained via experiments with the planet discovery task may not generalize to other tasks. However, the overall approach and methodology can provide the basis for analogous studies. Also, the results have general implications on strategies for compensating workers in online task markets. We found that worker behavior is sensitive to variation of methods of payment. We believe that such influences of payment scheme on worker behavior is a feature rather than a drawback: paying workers the same effective wage, but with different piece-rate methods, can be used to trade off precision, recall, speed, and total attention on tasks. In our case, the canonical per-task payment used on MTurk and many other task markets results in the fastest task completion, but lowest recall. Other methods of payment, such as paying a wage, caused workers to work more slowly, but with better results. Being able to selectively control the output of human workers is desirable for many algorithms that use human computation, and the use of financial incentives in this way is an effective lever that warrants further careful study.

We also observed that the payment methods vary in their sensitivity to difficulty level, and this finding suggests that the performance of volunteers and workers paid using different methods may vary in sensitivity to the hardness of the task. For the planet discovery task, workers being paid in the canonical per-task scheme showed the greatest drop in precision as difficulty increased. The findings suggest that the design of financial incentives is important in

achieving a desired level of performance from crowd workers for a heterogeneous set of tasks. We believe that we have only scratched the surface in exploring the differences in incentives between unpaid citizen science projects and paid crowdsourcing platforms. Comparing the motivations of workers in each of these settings is an important problem that warrants further study.

Our experiments indicate that, even in paid task markets, indirect or secondary incentives may influence the behavior of workers. When examining the reasons that microtask workers may get distracted or leave, we find that many workers report being concerned about the quality of their work. On the other hand, it is likely that some of these workers may behave differently and provide low-quality work in response to a task with loose controls or to a requester with low standards. However, this also suggests that, over the short term and with the right controls, one can indeed use different non performance-contingent payment schemes to collect high quality data from workers. Overall, our collective observations highlight multiple opportunities and directions with pursuing deeper understanding of how incentives influence the behavior and output of crowd workers.

3.5 Acknowledgments

This chapter involved collaboration from Ece Kamar, Eric Horvitz, Yiling Chen, Chris Lintott, Arfon Smith, and Megan E. Schwamb. Portions of this chapter previously appeared in the HCOMP conference publications *Why Stop Now? Predicting Worker Engagement in Online Crowdsourcing* (Mao et al. 2013b) and *Volunteering vs. Work for Pay: Incentives and Tradeoffs in Crowdsourcing* (Mao et al. 2013a).

This work presented in this chapter was commenced during an internship at Microsoft Research and was later partially supported by the NSF under grant CCF-1301976. We thank Chris Lintott for sharing the Galaxy Zoo data, Paul Koch for assistance processing the data, and Rich Caruana for valuable discussions and feedback. We thank Matt Giguere and Debra Fischer for creating the simulated planet transit data, and John Johnson for providing Figure 3.12.

Chapter 4

Performance and Team Size on a Crisis Mapping Task

4.1 Preliminaries

Teams are fundamental to a wide range of production and problem solving tasks (Guimera et al. 2005, Lazer and Friedman 2007, Wuchty et al. 2007, Ungar et al. 2012), many of which are “complex” in the sense that they comprise sub-tasks that are to some extent interdependent (Bettencourt 2009, Shore et al. 2015). In turn, task complexity implies that team performance should increase with size, in part because division of labor allows for greater specialization (Becker and Murphy 1992), and in part because workers in teams can learn from others (Ungar et al. 2012, Mason and Watts 2012b). Other factors, however, suggest that increasing team size can hurt productivity, either because workers find it increasingly tempting to free ride on the efforts of others (Holmstrom 1982), or because the overhead associated with communication increases with the number of individuals whose efforts must be coordinated (Brooks 1975, Malone and Crowston 1994). In the presence of multiple, conflicting factors, the relationship between team size and performance is necessarily an empirical matter. Yet empirical studies (Gooding and Wagner III 1985, Wheelan 2009, Guimera et al. 2005) have had difficulty identifying the casual effect of size in the presence of potential

confounds such as task type, environment, or management style. Meanwhile, controlled laboratory experiments have focused on activities—whether physical (e.g. rope pulling, shouting, clapping (Ingham et al. 1974, Latane et al. 1979)), or mental (e.g. word puzzles (Littlepage 1991, Laughlin et al. 2006), brainstorming or ranking lists (Karau and Williams 1993, Woolley et al. 2010))—that lack the complexity of most real-world settings.

In this chapter, we report results from an experimental study of *crisis mapping*, an activity in which groups of volunteers collaborate online to monitor, classify, and then map real-time crisis-related information, often in the form of social media reports posted by individuals in the midst of the crisis (typically a natural disaster such as an earthquake or hurricane, but also potentially a political conflict) for the purpose of informing decisions about resource allocation, inter-agency coordination, or some other type of humanitarian assistance (Meier 2015). As a model task for studying team performance, crisis mapping has a number of advantages. First, it is simple enough that participants do not require any specialized skills *ex-ante*, yet complex enough to benefit from division of labor, specialization, and coordination among team members. Second, it is a task that is natively online, and thus experiments conducted in an online environment with subjects recruited from online crowdsourcing sites bear a close resemblance to the real-world activity. Third, by conducting our experiments online we can fully exploit the advantages of “virtual lab” environments, both in terms of the range of n that we consider and the granularity of the data that we collect. And finally, although crisis mapping is a relatively recent phenomenon, originating in 2010 during the Haiti earthquake, it has attracted considerable attention in the humanitarian affairs community (Meier 2015), hence better understanding of how teams of workers solve this problem is of practical as well as scientific importance.

4.1.1 Crisis Mapping and the Standby Task Force

Crisis Mapping is an umbrella term referring to informatics for natural and man-made disasters, usually conducted in a decentralized manner by volunteer organizations. The *Digital Humanitarian Network* (Meier 2015) is the largest network of crisis mapping organizations,

specializing in aspects such as aerial imagery, mapping technology, and digital volunteerism. Crowdsourced, volunteer crisis mapping originated with the 2010 Haiti Earthquake, where a real-time collaboration between Internet volunteers effectively monitored emerging news and social media data and categorized and verified reports of damage and casualties. The success of this effort led to the formation of the *Standby Task Force* (SBTF), a digital volunteer organization for collaborative mapping. In contrast to other efforts leveraging technology or computation, the SBTF focuses on human intelligence through the coordination and organization of volunteer teams. A *deployment* of the SBTF is a call for volunteers to work together online as a disaster unfolds, often alongside other digital humanitarian organizations.

During a typical deployment, the SBTF focuses on three main tasks as part of different teams: *media monitoring* (searching Twitter, news feeds, and other social media for relevant crisis events and filtering irrelevant reports), *geolocation* (geographical familiarity with the target area and finding physical locations of reported crisis events), and *verification* (checking the quality and correctness of mapped events). Several leaders and coordinators oversee the effort and communicate with other organizations. This particular organizational structure emerged after earlier deployments when participants realized that specializing on a specific type of task would increase overall effectiveness. Yet, the tools used by the SBTF are relatively ad hoc, with Google documents for collected data, Skype for group chat, and a Ning website for persistent information about the organization—typically requiring members to use multiple applications simultaneously.

4.1.2 Typhoon Pablo Deployment

Typhoon Pablo (also known as Typhoon Bopha) was a category 5 tropical cyclone that made landfall in the Philippines on Dec 4 2012 and caused widespread damage and loss of life over the ensuing 24 hrs¹. On the evening of Dec 5, in response to a request from the UN’s OCHA (Office for the Coordination of Humanitarian Affairs), 32 SBTF volunteers deployed for a 12 hour period with the objective “To collect all relevant tweets about Typhoon Pablo posted

¹See http://en.wikipedia.org/wiki/Typhoon_Bopha for details.

on December 4th and 5th; identify pictures and videos of damage/flooding shared in those tweets; geo-locate, time-stamp and categorize this content.”². To achieve this goal, they were given a collection of approximately 1600 tweets that had been pre-processed to eliminate re-tweets and to preserve tweets with links.

Although 32 volunteers registered to work on the task, the user names tagged to tweets suggest that only 18 produced actual classifications. Over the course of the 12 hour deployment these 18 active workers tagged 93 tweets; however, some of these tweets reported similar damage in similar locations, hence the number of distinct events was less than 93. During this same period *Humanity Road*, a different crisis mapping organization, tagged an additional 40 tweets from the same set. The combined set of 133 tagged tweets was then mapped, and the resulting crisis map submitted to the UN’s OCHA (Figure 4.1). More details of the deployment are documented³ and a copy of the spreadsheet from the deployment is also publicly available⁴.

4.2 Experiment Design

In this work, we simulated the deployment of the Standby Task Force (SBTF) during Typhoon Pablo. To replicate the general work flow of the SBTF while also maintaining a high level of experimental control, we designed and built a collaborative, real-time web application that allows many groups of users to work simultaneously on the same task (see Figure 4.2). Workers recruited from Amazon’s Mechanical Turk, a crowd-sourcing site that is commonly used by researchers to recruit and pay subjects for behavioral experiments (Mason and Suri 2012b), first completed a tutorial explaining the functionality of the platform and were then randomly assigned to teams of sizes $n = 1, 2, 4, 8, 16$, and 32. We conducted a total of 50 experiments comprising 258 unique individuals, where no individual participated more than once.

²<http://irevolution.net/2012/12/06/digital-disaster-response-typhoon/>

³<http://blog.standbytaskforce.com/2012/12/09/how-the-un-used-social-media-in-response-to-typhoon-pablo-updated/>

⁴<http://bit.ly/1PjIAdV>

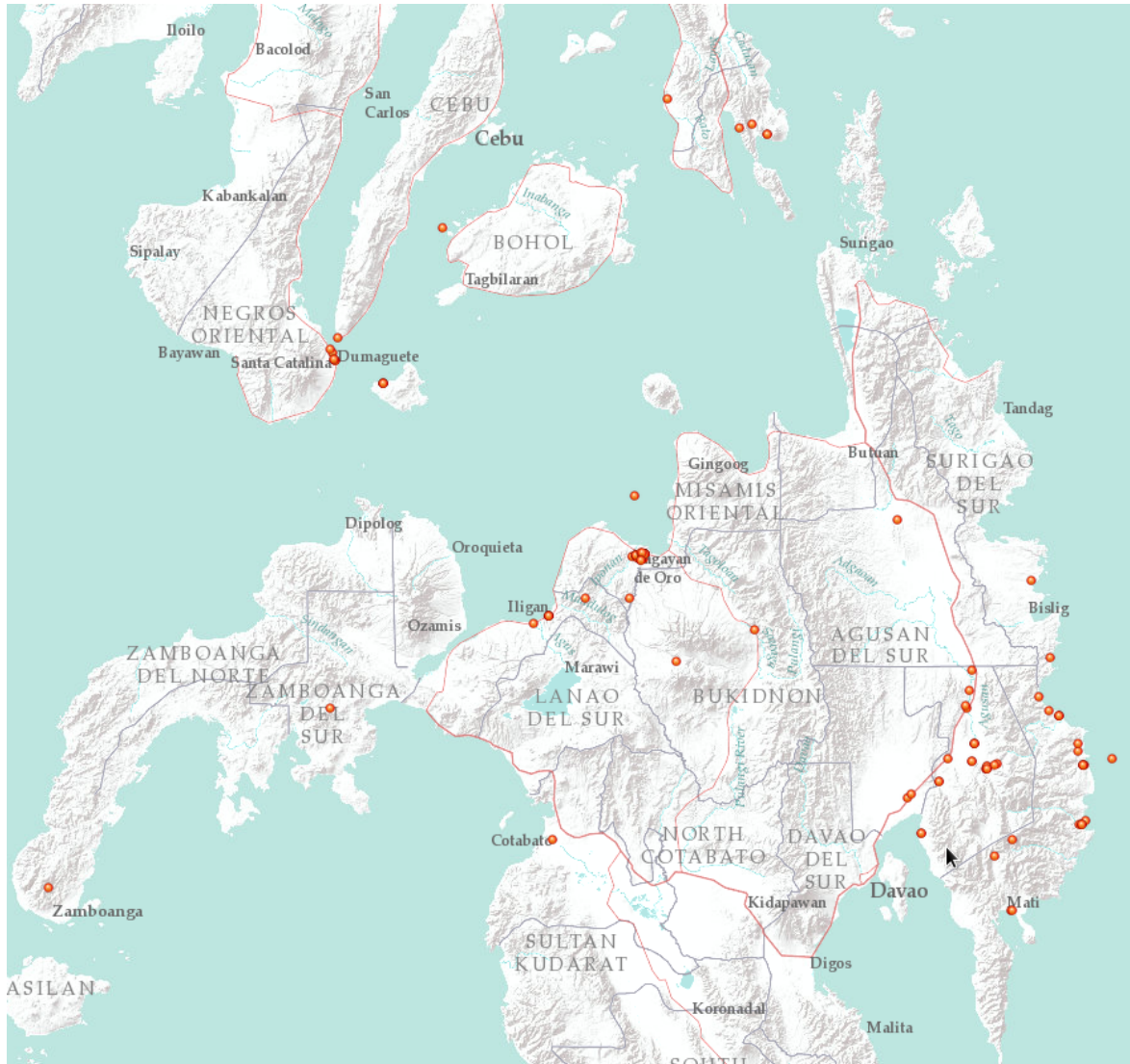


Figure 4.1: Crisis map produced by the Standby Task Force and Humanity Road in response to Typhoon Pablo, Dec 5 2012.

4.2.1 Collaborative Real-time Mapping Application

To capture the essential features of a real-world SBTF deployment while also adding a high degree of experimental control and data logging, we designed and built a customized real-time web application called *CrowdMapper*⁵ that allows a group of participants to self-organize and work on a crisis mapping task within a single software platform. The application, shown in Figure 4.2, consists of the following components:

- **Events.** An editable table contains all crisis events recorded by the group. Columns of the table are configurable aspects of the event such as its category, a textual description, and location information. Each entry in the events table can be edited simultaneously by different workers. A map view of the events displays any event geographically with a recorded longitude/latitude. Workers can also vote up or down each event entry as a way to vouch for its correctness to others.
- **Tweet stream.** A common feed of Twitter-style messages (tweets) is shown to all workers in the group. Each message contains potentially relevant crisis data, and may contain links to outside websites. Workers are encouraged to click links to view their content, and can hide (filter) an irrelevant tweet by clicking on a red **X** button, or drag a relevant tweet to the event table to attach it as a source for an existing recorded event. Multiple workers can filter different segments of the tweet stream simultaneously without disturbing one another.
- **Chat rooms.** Workers can create any number of chat rooms with different names. Each chat room displays the entire history of messages in that room, and workers can switch freely between chat rooms. Workers can also send invitations to other workers inviting them to join a particular chat room. When sending a message, a special tagging syntax allows referring to other workers, or tweets and events.

⁵The application and our experiment code is open source and can be found at <http://github.com/mizzao/CrowdMapper>.

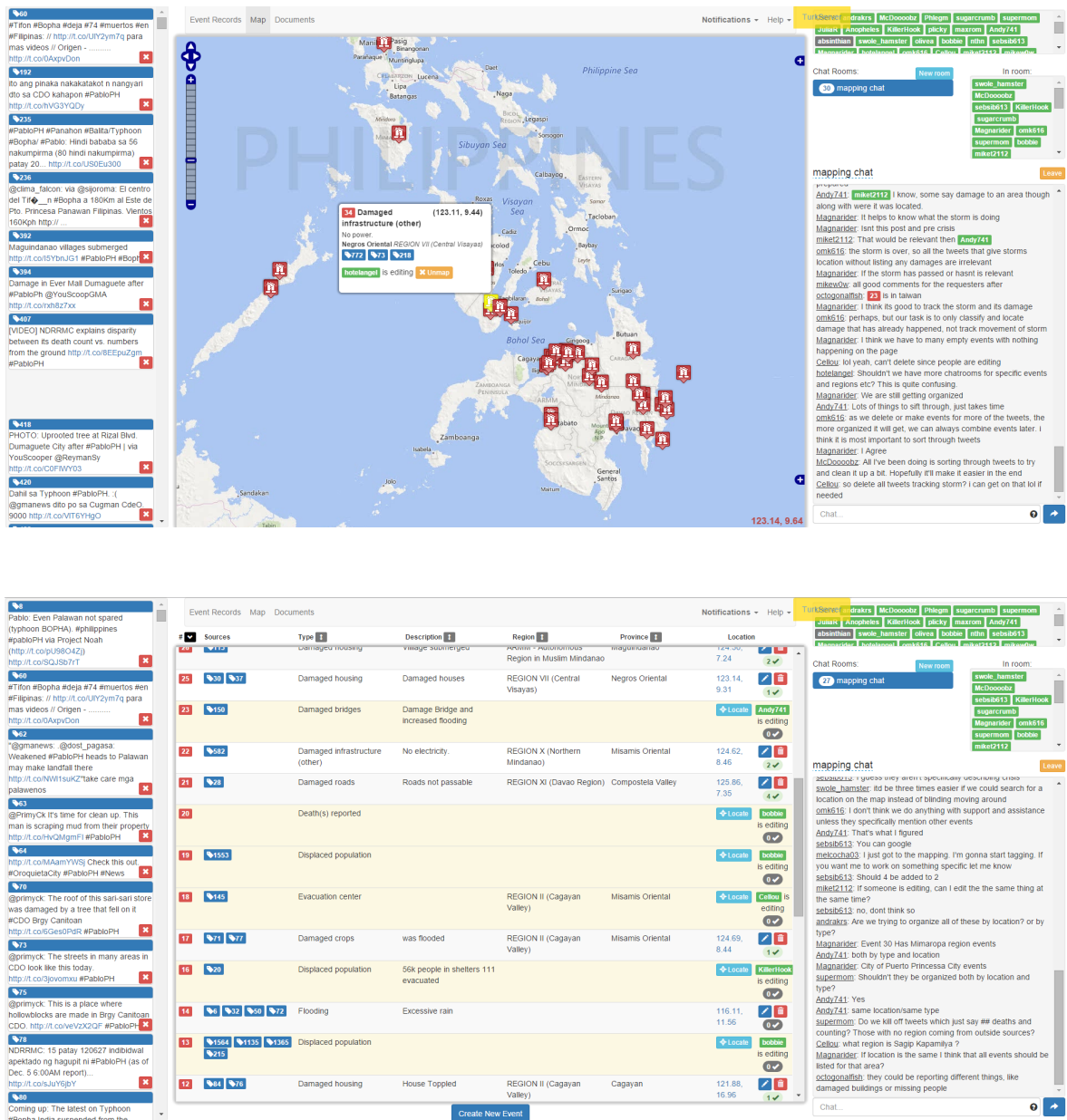


Figure 4.2: Map (top) and table (bottom) views of the CrowdMapper platform

- **Documents.** Workers can create any number of documents with the ability to simultaneously edit, and use them to store persistent data.

In particular, the interface does not prescribe any particular task to participants. Each worker is free to use any part of the interface, or even do nothing at all. CrowdMapper was refined through several tests and a pilot experiment, and we designed an interactive tutorial to train new workers. A time-lapse video showing the progress of one of the $n = 32$ groups working on CrowdMapper is available at https://youtu.be/xJYq_kh6N1I.

4.2.2 Input Data

Our experiment uses a set of 1,567 pre-filtered tweets from the December 2012 deployment of the SBTF with the goal of conducting a damage assessment of Typhoon Pablo impacting the Philippines. This dataset is particularly useful for evaluating the performance of a group on the mapping task because there are several annotated maps of the typhoon’s damage from this set of Twitter data, allowing us to construct a relatively accurate gold standard of aggregated crisis damage. Some links among the tweets in this dataset may have been broken at the time of our experiment; however, as all experiment sessions took place over a short period of time, all groups experienced the same broken links.

4.2.3 Subject Recruitment and Training

We recruited almost 1,300 workers from Amazon’s Mechanical Turk to build a panel of workers for the experiment. During the recruitment task, each worker completed a short, interactive tutorial that explained each part of the interface and required them to map a few example crisis events. The full text of the tutorial is presented in Appendix A.

The workers were instructed that they had to complete the following tasks: (1) identify tweets that referred specifically to instances of typhoon-related damage (e.g. washed-out bridges, flooded roads, damaged crops or buildings, displaced population); (2) create event records for every such instance, describing the event in words, attaching the relevant tweets, classifying it as one of several predefined types, and establishing its geographical location

(by latitude and longitude as well as province and region); and (3) verify and if necessary correct the information in existing event reports. After completing the tutorial, we collected feedback about any part of the interface that was confusing, as well as workers' availability by timezone to schedule the simultaneous part of the experiment. Finally, we asked if workers would agree to be contacted for scheduled experiment sessions.

4.2.4 Informed Consent

Before proceeding with recruitment our project was subjected to ethical, privacy, and legal review. Because it qualified as human subjects research under the HHS Common Rule, we followed standard procedures for obtaining informed consent from each of our participants prior to their participation. In addition, before they could participate in the experiment, participants were required to complete the tutorial twice—once to join the panel and a second time immediately before joining the experiment. Thus all participants had a clear idea of what they were being asked to do, and how much they were likely to be compensated for their time. In addition, all participants were offered the opportunity to complete an exit survey in which they could register any complaints about the experiment or the instructions. We did not receive any serious complaints about either.

4.2.5 Group Assignment

Over a period of two weeks during August 2014, we conducted our experiments in scheduled sessions with simultaneously participating users. Each day, we selected a random subset of workers in our panel who had not yet participated in the task, chose a time of day that showed the most availability by workers' preferences, and sent an e-mail in advance asking workers to arrive at that particular time. All workers who arrived again completed the same tutorial as they had during recruitment, and again consented to participate in the experiment.

Because participants did not arrive at precisely the same time and because they also took varying lengths of time to complete the tutorial, they were assigned to a virtual “lobby” until sufficiently many participants were available to begin the experiment. At this point, all

Team Size	Completed Teams	Number of Individuals	Dropouts (rate)
1	18	21	3 (14.2%)
2	11	22	1 (4.5%)
4	6	24	3 (12.5%)
8	4	31	2 (6.5%)
16	4	59	3 (5.1%)
32	4	123	10 (8.1%)
Total	47	280	22

Table 4.1: Assignment of individual teams, showing number of teams that completed the task, number of teams that did not complete the task, total number of individuals in each condition, and number of dropouts.

participants in the lobby were released simultaneously and randomly assigned to a predetermined set of teams of varying sizes. Every group was then presented with the same set of 1,567 tweets; i.e. groups only differed in the number of participants assigned to the task.

Workers assigned to the $n = 1$ condition were told that they would be working alone; in all other conditions they were told their number of coworkers as well as their user names and were instructed on how to create chat rooms for the purpose of communicating; however, they were not given any instructions in how to coordinate. All teams were given one hour to identify as many crisis-related events as possible from the time of arrival of the first participants.

Although ideally our design required all teams to be fully populated at the same time, the variance in arrival and tutorial-completion times would have required long waits for some participants, thereby increasing the likelihood of attrition. To address this concern, we released the waiting room no later than 10 minutes after the first arrival, and then continued to assign subsequent arrivals to groups for several minutes after the task started⁶. As a consequence not all teams had a full complement of workers for the entire duration of the experiment. Table 4.1 shows the resulting number of individuals assigned to teams in each condition. A Fisher exact test on the dropout rate is insignificant at the 10% level across all pairs of conditions, suggesting that the attrition rate did not significantly change based on the treatment.

⁶The slowest arrivals were all routed to a single ‘buffer’ group and this data was excluded from our analysis.

4.2.6 Worker Incentives and Monitoring

Since our experiment was conducted on Amazon Mechanical Turk, we paid workers for their participation. We designed an incentive structure, based on the positive results from using an hourly wage in Section 3.3, to encourage workers to participate and collaborate with their team while preventing obvious strategic behavior. Based on performance, each team was assigned an hourly wage between \$6 and \$15, computed as $\text{Team Wage} = \$6 + \$9 \times \frac{\text{Team Performance}}{\text{Best Team Performance}}$, where the minimum of \$6 per hour is considered the norm for minimum acceptable wage on Mechanical Turk.⁷ Each participant was then paid an amount corresponding to their participation time scaled by team wage: $\text{Individual Payment} = \text{Team Wage} \times \text{Active Time}$. Workers were informed that their individual compensation would vary between a minimum of \$6 and a maximum of \$15 and would be computed as a joint function of their own time spent working and the overall performance of their team relative to other teams.

We used a software-based inactivity monitor to detect when participants were inactive for an extended period of time, and this time was not counted toward their payment. Individuals could realize the maximum possible payment only by both participating throughout the task and also ensuring that their team performed well, a point that we emphasized during the tutorial. Moreover, because our compensation scheme was based on a combination of individual time worked and collective performance relative to (unobserved) other teams, it was extremely hard to game (i.e. it did not reward any specific type of activity over others), and also penalized both explicit loafing as well as more indirect forms of free riding (e.g. performing “busy work” that didn’t contribute to the stated goals). Workers therefore had real and meaningful monetary incentives both to contribute individually and also to cooperate with their coworkers.

4.3 Evaluation Methods

Our web application collected a comprehensive log of activity by all workers over the course of each experiment. This log allows us to reproduce the state of each experiment group at

⁷http://wiki.wearedynamo.org/index.php?title=Fair_payment

any point in time, and compute several measure of output and performance and described below.

4.3.1 Computation of “Person-Hours”

Obviously the disparity between nominal group size and the actual number of active workers shown in Table 4.1 renders comparisons in terms of nominal team size problematic. A team of nominal size $n = 32$, for example, would almost certainly be missing at least some of its workers for at least some of the time (as described in Section 4.2.5), and because this amount could vary from day to day or even across groups within the same session, two groups of the same nominal size might have substantially different amounts of actual labor available to them. For this reason, we generally avoid comparison in terms of nominal group size, instead comparing team performance as a function of “person-hours” τ_p defined as the total time worked by active team members:

$$\tau_p(t) = \sum_{i=1}^n t - t_0^i \quad (4.1)$$

where t is the clock time for the experiment, and t_0^i is the start time of the i^{th} worker. In this way we can compare groups of all sizes in a consistent and meaningful manner.

4.3.2 Constructing the Gold Standard

We measure group performance relative to a “gold standard” crisis map that we constructed in two stages. First, we aggregated all of the events generated by all groups in the experiment, and removed all duplicates, ensuring that each potential event was included only once. Second, we then manually checked all the de-duped events, and deleted any that could not be independently verified either by cross-referencing them with the SBTF map or through some other means. The resulting map is shown in Figure 4.3 and is available in our software repository⁸. Following this procedure, our gold standard map comprised 49 distinct events, which were referred to by 245 distinct tweets.

⁸<http://github.com/mizzao/CrowdMapper>

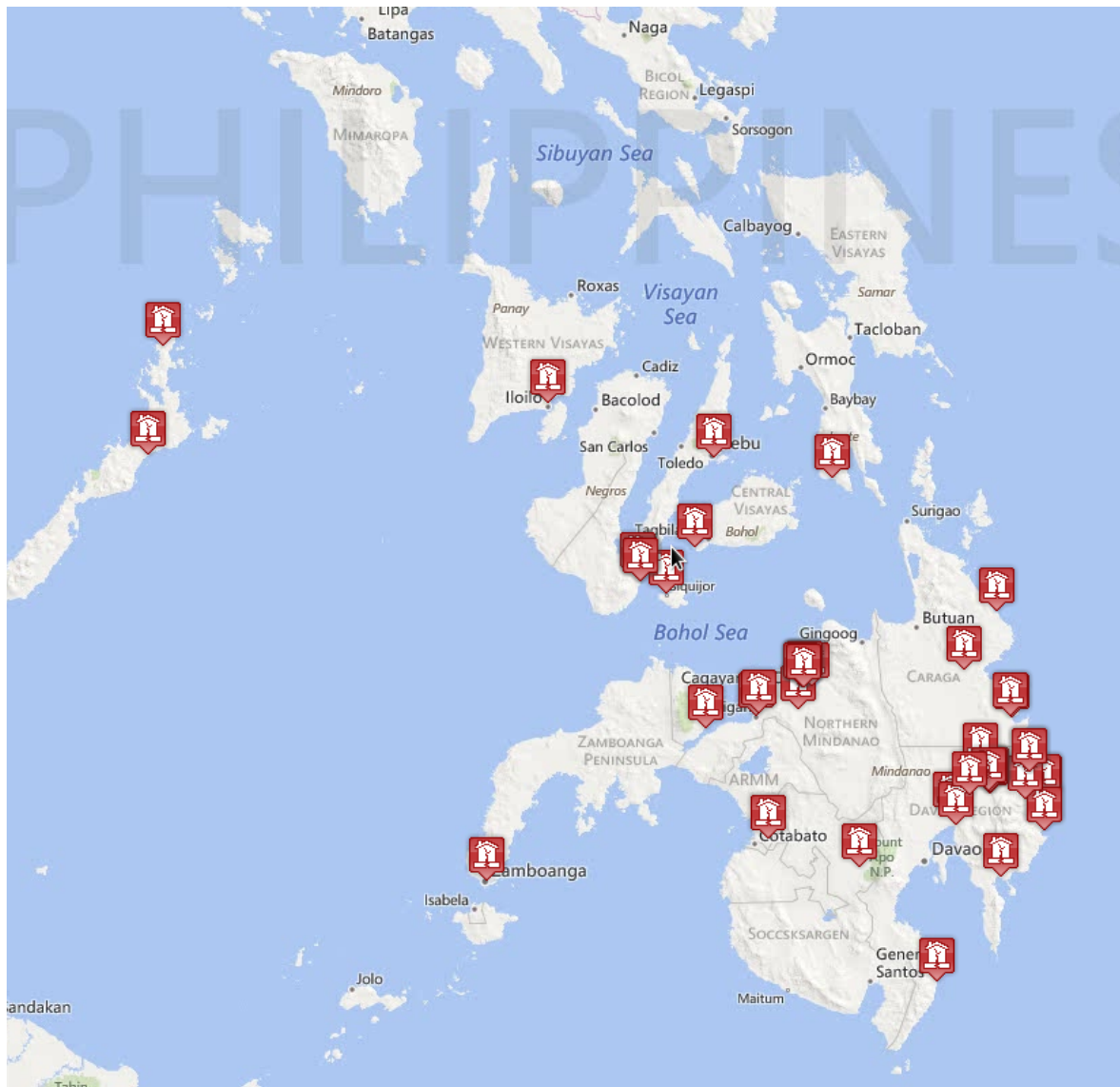


Figure 4.3: The gold standard crisis map.

We note that our gold standard could be missing real events that were not detected by any of our experimental groups, hence we do not refer to it as a “ground truth” map; however, it is by definition impossible for any group to perform better than the gold standard.

4.3.3 Performance Measures

By comparing groups’ results against the gold standard, we can compute performance for each group as follows. First we compute a score for each tagged event relative to the gold standard⁹. That is, for each event e_{ij} in group i and each event e_k in the gold standard, we define the *fractional score*

$$s(e_{ij}, e_k) = \frac{1}{4}I(\text{correct type}) + \frac{1}{4}I(\text{correct province}) + \frac{1}{4}I(\text{correct region}) + \frac{1}{4}\log_{10}(\max(10 \text{ km}, \min(100 \text{ km}, \text{distance in km})) - 1) \quad (4.2)$$

That is, $s(\cdot) \in [0, 1]$ for any pair of events, with each of the categorical type, province, and region fields, and latitude/longitude distance contributing 1/4 of a point. The distance is clamped to a minimum of 10 km (worth 1/4 point) and a maximum of 100 km (worth 0). We ignore the contents of the free-text description field to avoid any dependence on text processing. We then define the *binary score* as the fractional score rounded at 2/3:

$$b(e_{ij}, e_k) = \begin{cases} 1, & \text{if } s(e_{ij}, e_k) \geq 2/3 \\ 0, & \text{otherwise} \end{cases} \quad (4.3)$$

The threshold of 2/3 was chosen so that an event would need to have all three categorical fields correct, or two fields correct and be within ≈ 20 km, to match an event in the gold standard. We can then define the number of identified events by a group as the size of a *maximum matching*¹⁰ between e_{ij} and e_k using the scores $b(\cdot)$. This measure allows for groups

⁹Our scoring method focuses on events (as opposed to tweets or some other unit of analysis) on the grounds that the goal of crisis mapping is to correctly identify as many relevant events as possible for consumption by a client agency such as UN’s OCHA.

¹⁰A maximum matching of a bipartite graph, also known as the *assignment problem*, is computable in polynomial time using the Hungarian algorithm.

to obtain credit for an event if it is “close enough” to the original, while preventing duplicate events from counting twice.

Because false positives, false negatives, and duplicate crisis events all impair the quality of a crisis map, we desire a scoring metric that takes these factors into account. Building on these event-based scoring procedures, therefore, we can now compute the following performance metrics:

- Recall measures the performance of a group of workers to find unique event records within the set of data, defined as for group i as

$$r_i = \frac{|\{\text{gold standard event records}\} \cap \{\text{identified event records}\}|}{|\{\text{gold standard records}\}|}. \quad (4.4)$$

where the numerator is the size of the maximum matching of the bipartite graph defined by Equation 4.3.

- Precision measures the accuracy of groups in the events that they generate, defined as

$$p_i = \frac{|\{\text{gold standard event records}\} \cap \{\text{identified event records}\}|}{|\{\text{identified event records}\}|}. \quad (4.5)$$

- F_1 score is the harmonic mean of precision and recall

$$F_{1,i} = 2 \frac{r_i \cdot p_i}{r_i + p_i} \quad (4.6)$$

4.3.4 Intermediate Group Performance

Because we know the exact state of any group at any point in time, we can follow the above procedure for constructing a gold standard and comparing a group’s map at any point in time. In this manner, we can compare groups at specific stages of task completion as well as at the end of the experiment. Specifically, we divided the total amount of person-time spent on the task into quadrants and computed statistics after the completion of each quadrant. For example, due to the simultaneous recruitment process, a group of 8 users may have spent

only 7 person-hours on the task. The performance of the group at 50% of the task would be computed at 3.5 person-hours from the start of the task. In this way, we can adjust for differences in how quickly the tasks filled (as noted earlier), as well as for users that became inactive or did not contribute. Moreover, we used a similar approach to compare different groups at the same amount of person-time, e.g. computing group performance for all groups at the time they reached 2 person-hours of progress.

4.3.5 Comparison with SBTF Deployment

We also score the original crisis map produced by the Standby Task Force (SBTF) to the gold standard map used to evaluate teams of workers in our experiment. Although we believe this comparison to be a reasonable demonstration of external validity for our experiment, we note a few important differences.

Because of the lengthy delay between the original deployment (Dec 2012) and our experiments (Aug 2014), however, some of the links that were used by the SBTF were no longer usable by our teams. As a consequence it is conceivable that the SBTF could have correctly reported events that could not have appeared in our gold standard, and that these events would be registered by our scoring methods as errors, thereby artificially reducing the precision of the SBTF. Fortunately manual inspection of the SBTF data reveals that these events were extremely rare in practice, hence their presence did not overly penalize the SBTF relative to our groups.

In addition to degradation in the data affecting our scoring procedure, the SBTF differed from our experiment in a number of other respects. First, one might expect that volunteers in a crisis situation would differ considerably in disposition and motivation from paid participants in a simulated exercise; moreover, in contrast with our participants, SBTF volunteers all received some degree of mapping training, and were organized by experienced leaders who had worked on previous deployments. Second, our experimental platform integrated all the mapping and communication functions into a single application greatly reduced the overhead compared to working in several applications (e.g. Skype, Google Docs, Google Spreadsheets,

and other websites) simultaneously. Our mapping software also made the identification and aggregation of duplicate tweets significantly easier than the Google Spreadsheet format used by the SBTF. And finally, our experiments required participants to be online at the same time and thus improved the possibilities of coordination between multiple people.

For all these reasons the performance comparison between the SBTF and our experimental groups should be regarded as approximate, not exact. Nonetheless, the performance of the best performing experimental groups is remarkably similar to that of the SBTF, all the more so in light of the difference in time-scales (twelve hours vs. one hour). Thus we believe the comparison is a persuasive indicator of external validity.

4.3.6 Synthetic Groups

We also sought to evaluate the effectiveness of groups relative to individuals working independently. Thus, we compare the performance of synthetic groups of individuals with actual groups that collaborated during the task. To make this comparison, we combined the output of random subsets of individuals who had worked alone, and used the combined set of events to compute the matching as described previously.

For example, to compute the performance of synthetic groups with 8 person-hours of work, we randomly sampled, without replacement, 100 sets of 8 individuals from the total of 18 participants who had worked independently for one hour. Each such sample defines a set of events generated by all individuals in that sample, for which we can compute precision, recall, and F1 score. We repeat this process for all numbers of person hours for $n = 2$ to 16, and additionally for the 18 groups of $\binom{18}{17}$ individuals as well as the combined group of all 18 individuals. Figure 4.5 shows the results of this sampling as box plots displaying the median and interquartile range for each value of person-hours.

4.3.7 Measuring Effort

We also measured individual and group effort as follows. First, all worker actions were logged and coded according to one of four high-level action types, as described in Table 4.2:

Category	Action Type
Filtering	Hiding irrelevant tweets, attaching tweets to events
Classification	Filling out fields in event records, geolocation
Verification	Moving and removing tweets from events, editing existing records, voting
Chat	Messages to other workers

Table 4.2: Categories of worker actions in the crisis mapping experiment.

“filtering” (deleting irrelevant tweets or adding relevant tweets to event reports); “classifying” (adding new information to event reports); “verifying” (editing existing information, deleting reports, moving tweets between reports, or clicking a report’s “thumbs up” button); and “communicating” (participating in one or more chat rooms). Second, averaging over all instances of a given action type across all workers in all conditions, we computed an “action time” for each action type. Finally, summing these action times over all actions of a given worker yields the total of “effort hours” contributed by that worker. By expressing work in units of effort hours, we can compare individuals and groups alike in a consistent way.

4.3.8 Measuring Collaboration

For each event, we measure collaboration by weighing participants’ contributions by their effort. For example, suppose that a total of m workers each contributed a fraction p_k of the total effort on event e_{ij} . The *effort entropy* is therefore

$$\eta_{ij} = \sum_{k=1}^m -p_k \lg p_k \quad (4.7)$$

This is the standard information-theoretic measure of entropy, measured in bits. We define the collaboration on event e_{ij} as $c(e_{ij}) = 2^{\eta_{ij}}$. This has the property that if m people contribute equal effort to recording an event, the collaboration also has value m . On the other hand, this value decreases with unequal contribution, such that if a single person contributed almost all of the effort to an event, the collaboration value is very close to 1. The collaboration of a particular group is then the mean collaboration value over all events in the group.

4.4 Results

Given the gold standard as constructed in Section 4.3.2, we then computed, for every team i , the measures of performance in Section 4.3.3: *precision*, defined as the fraction of events identified by team i that matched the gold standard; *recall*, defined as the fraction of all events in the gold standard that team i correctly identified; and F_1 score, defined as the harmonic of mean of precision and recall. Because both precision and recall are important for our application—i.e. both false positives and false negatives impair the utility of a crisis map—we focus on F_1 as our preferred measure of performance (although for explanatory purposes we also include results for precision and recall). Finally, in order to test the external validity of our results we also scored the performance of the SBTF deployment as described in Section 4.3.5.

Fig. 4.4A shows performance (F_1 score) over the course of the experiment for teams of all sizes (computed by following the above procedure at intermediate time points). Performance increased monotonically with time for all team sizes, suggesting that teams were actively engaged in the task, continually adding new event reports (thereby improving recall) and editing existing reports (thereby improving precision). Fig. 4.4A also shows that relative performance was roughly consistent over the course of the experiment, hence we can focus on the end of the task with little loss of generality. Second, Fig. 4.4B shows F_1 score as a function of person-hours, defined as the sum of time worked by all team members, at the completion of the experiment (as explained in Section 4.3.1 and Table 4.1, we use person-hours rather than nominal team size n because the larger teams were not completely populated with active workers for the whole time). Average performance, indicated by a least squares fit (solid line), increased monotonically with person-hours ($r = 0.689, p < 10^{-10}$) eventually achieving comparable performance to the SBTF deployment (dashed line), a notable accomplishment given that the latter comprised between 18 and 32 workers and lasted for 12 hours. Third, however, Fig. 4.4B also shows that performance exhibited diminishing marginal returns: per-capita productivity was highest for individuals and decreased with team size. Fig. 4.4C and D decompose this effect into precision and recall respectively, showing that performance

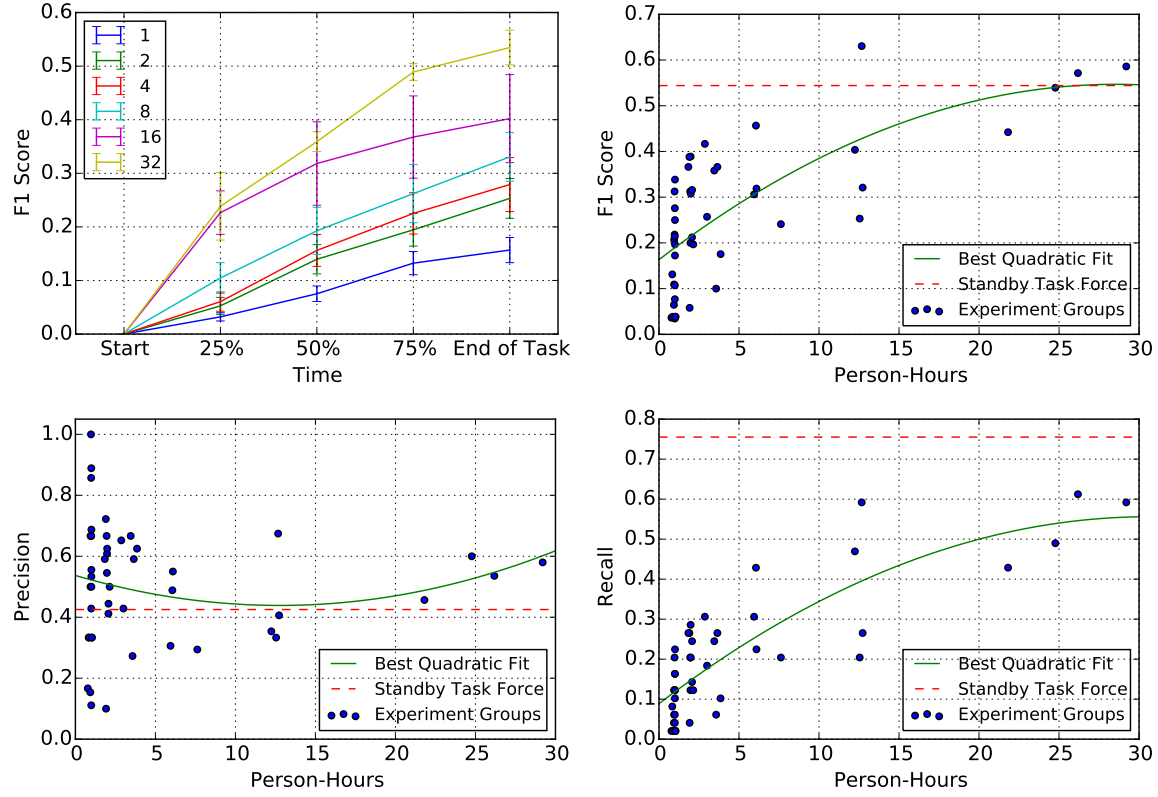


Figure 4.4: **A.** Performance (F_1 score) vs time for teams of size $n = 1, 2, 4, 8, 16, 32$. Error bars show standard errors. **B.** F_1 score vs. actively worked person hours. **C.** Precision vs. actively worked person hours. **D.** Recall vs. actively worked person hours. In **B-D** each dot represents one team, and solid lines indicate least-squares best fit. The dashed lines indicate the performance of the SBTF.

differences were dominated by improvements in recall ($r = 0.807, p < 10^{-10}$), which also increased with diminishing marginal returns, whereas precision neither consistently increased nor decreased with team size ($r = -0.004, ns$).

Fig. 4.4 suggests that in spite of their favorable comparison to the SBTF, large teams under-performed small teams and individuals on a per-capita basis. However, this comparison ignores possible redundancies between the reports of smaller teams, and hence likely overstates the accuracy of their combined output. Fig. 4.5 addresses this possibility by comparing the performance of our experimental teams with “synthetic teams” constructed by combining the output of $n^* \leq 18$ independent workers, randomly selected without replacement (only 18 participants were assigned to the $n = 1$ condition). Fig. 4.5A shows that for

smaller values of person-hours synthetic teams generally outperformed real teams, consistent with the diminishing marginal returns argument. Interestingly, however, the performance of synthetic teams peaks around eight person hours and then subsequently decreases, with the result that the largest teams performed at least as well as the optimal-sized synthetic teams (solid line), and the best-performing large team (one of the $n = 16$ teams) outperformed all synthetic teams. Fig. 4.5B and C shed further light on this result, showing that although recall for synthetic teams increases monotonically, precision is monotonically decreasing, hence the harmonic mean of precision and recall (i.e. F_1 score) is non-monotonic.

What accounts for the superior per-capita performance of individuals visible in Fig. 4.4 but the superior collective performance of large teams in Fig. 4.5? Fig. 4.6 suggests that the answer is a combination of individual effort and collective coordination. First, we computed each individual worker’s effort as the weighted sum of that worker’s actions, where the weights represent the time, averaged over all workers, taken to complete an action of a given type (Section 4.3.7). Fig. 4.6A shows that, consistent with previous work on “social loafing” (Karau and Williams 1993), average individual effort decreased by more than 30% from $n = 1$ to $n = 32$ ($F(1, 45) = 1122.3, p < 10^{-10}$). Second, however, Fig. 4.6B shows that *collaboration*, defined as the effort-weighted number of participants who contributed to each event (Section 4.3.8), more than doubled over the same interval ($F(1, 45) = 98.77, p < 10^{-10}$). Fig. 4.6, in other words, shows that team performance was hurt relative to synthetic teams by team members exerting less effort than independent participants; however, the ability of team members to coordinate their efforts more than compensated for their reduced effort. An intuitive explanation for the coordination benefit experienced by teams in our task is that for any given event there is only one possible correct report whereas there are many possible incorrect reports. Combining the output of many independently generated reports, as we do for the synthetic teams, therefore increases the probability that at least one report will be correct, (improving recall), but also increases the probability that at least one will be wrong (hurting precision). By allowing individuals to coordinate, as they do in teams, the differences between multiple conflicting reports can be reconciled and precision improved

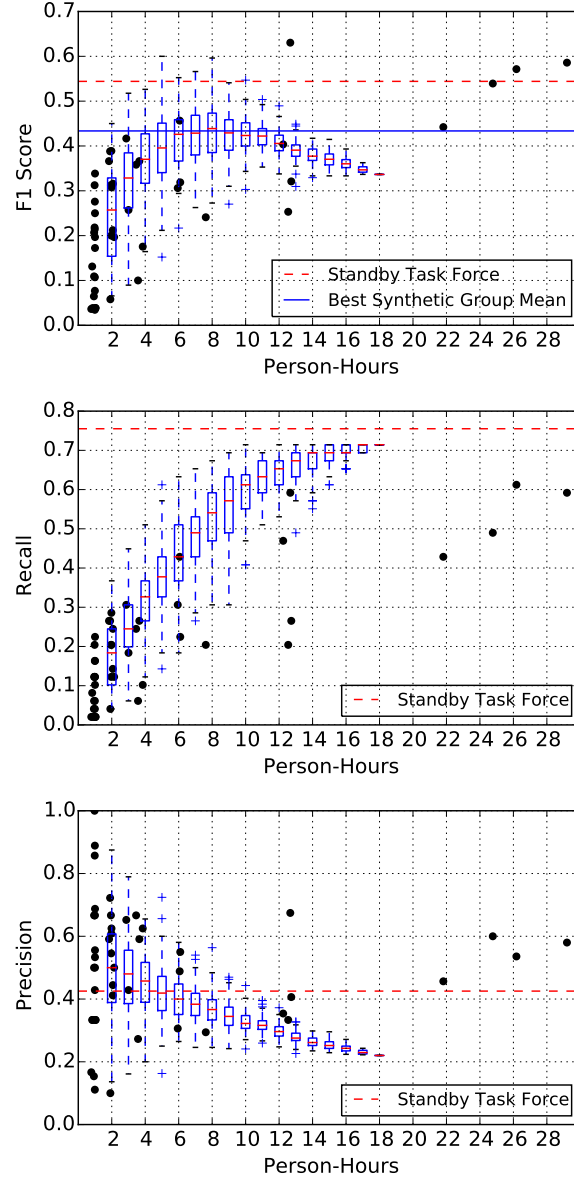


Figure 4.5: Performance (F1 score (top), recall (middle), and precision (bottom)) of synthetic teams relative to actual teams. Red bars indicate median performance of synthetic teams, boxes indicate interquartile range, and error bars full range. Dots correspond to the performance of experimental teams. Solid line indicates the average performance of the synthetic groups of size 8, the best performing synthetic group size. Dashed lines are the SBTF performance.

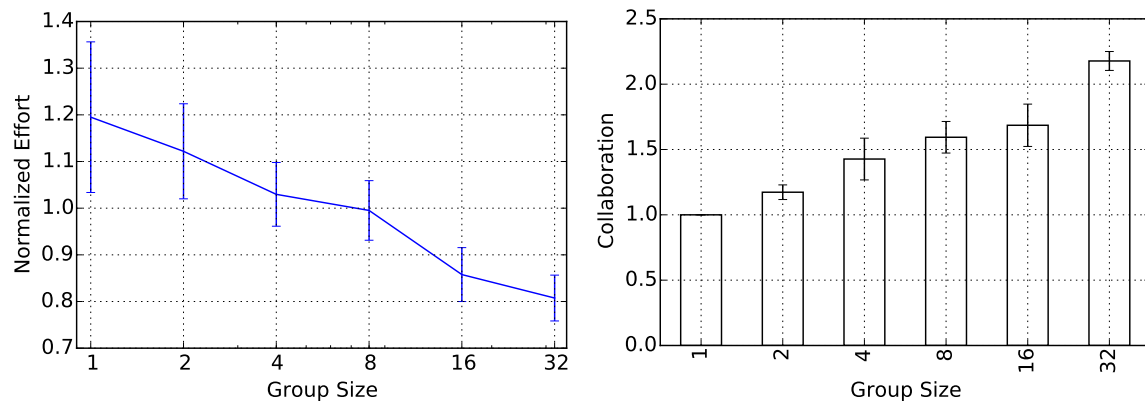


Figure 4.6: **A.** Individual effort per unit time according to team assignment (a value of 1.0 represents an average across all experimental conditions). **B.** Collaboration activity as a function of team size, defined as the mean effort-weighted number of contributors per event.

without much loss in recall.

4.5 Discussion

In addition to showing that the benefits of coordination allowed large teams to outperform individuals in spite of lower per-capita effort by their members, our results also open up a number of directions for future research. First, although social loafing has been studied extensively (Latane et al. 1979, Karau and Williams 1993), previous work has focused on relatively simple tasks for which effort is a uni-dimensional quantity. In contrast, for complex tasks such as ours effort is multidimensional, hence the same overall quantity of effort can be allocated across many different subtask types, potentially with important consequences for performance. The social loafing result is also interesting because individual compensation was computed as a function of *individual* effort, and team performance was only compared with teams of the same size. As both these features of the compensation scheme were designed to eliminate free riding (Holmstrom 1982), the observed reduction in effort is likely due to more subtle psychological effects (Karau and Williams 1993). The relationship between incentives, effort, and performance in complex collective tasks is therefore an open question that appears ripe for experimental study.

Second, it is also interesting and somewhat puzzling that in our experiment we did not observe more of a positive effect of the division of labor. One possible explanation is that although we trained all of our participants on the user interface prior to the experiment, we deliberately did not offer them guidance on how to organize themselves, and that the one hour allocated to the experiment did not allow sufficient time for a useful division of labor to emerge fully. Moreover, because we recruited inexperienced participants, any specialization must necessarily have emerged during the experiment itself, hence again one hour may have been insufficient for individuals to learn particular roles and hence fully realize the potential returns to specialization. For both these reasons, we anticipate that future experiments, harnessing some combination of more experienced participants and more tailored instructions will yield even larger benefits for teams vis-a-vis independent participants.

Third, the surprisingly strong performance of our experimental teams relative to the actual Standby Task Force deployment for Typhoon Pablo allow us to make some tentative predictions about real world applications. To reiterate, the core SBTF deployment comprised between 18 and 30 volunteers with varying degrees of experience who produced a crisis map in roughly 12 hours using the same 1600 tweets as our subjects. Although there were some differences between the 2012 deployment and our experiment that make an exact performance comparison problematic, our experiments nonetheless demonstrated that a similar number of inexperienced crowd-workers using our platform could construct a map of comparable coverage in just one hour. These results suggest that with further optimization in the lab and possibly also field deployments of our platform in real crises, “always-on,” real-time crowdsourced crisis mapping may be possible, thereby significantly expanding the capacity of existing digital humanitarian organizations (Meier 2015).

Finally, our mapping application and real-time experimental platform, comprising more than 30,000 lines of code, and which is publicly available, demonstrates that the high degree of interactivity, realism, and data instrumentation available in “virtual lab” settings is promising for understanding collective performance of large teams working on complex, real-world problems, especially relative to what is possible in traditional physical labs (Zelditch Jr 1969).

We hope that future research will leverage this platform to study both crisis mapping and also other types of collaborative tasks, thereby addressing a broad range of questions about team performance including the role of experience (Guimera et al. 2005), diversity (Page 2008), social sensitivity (Woolley et al. 2010), network structure (Shore et al. 2015), and leadership (Brooks 1975).

4.6 Acknowledgments

The work in this chapter was produced in collaboration with Winter A. Mason, Siddharth Suri, and Duncan J. Watts, and is under review at the time of this writing. The authors are grateful to Patrick Meier for providing the SBTF data, and to Meier and Matthew Salganik for helpful conversations.

Chapter 5

Voting and Probabilistic Ranking Models for Social Computing

5.1 Preliminaries

The problem of creating a ranking from noisy ranking or pairwise comparison data is widely studied across statistics, economics and machine learning. Statistical rank aggregation has been used to rank sports teams and racing drivers (Stern 1990), for information retrieval (Liu 2009), for collective ideation (Salganik and Levy 2012), in preference learning (Kamishima 2003), and for social choice (Conitzer et al. 2009).

Common to many current applications is that the input data comes from people, such as preferences in collaborative filtering or crowdsourcing and quality judgments in human computation. In crowdsourcing, a common problem is that of choosing the best or most desirable alternative from a large set of possibilities using ranking or voting (Chen et al. 2013). For example, Little et al. (2010a) give an example of ranking a list of suggestions for things to do in New York.

Yet, human-generated ranking data can be fickle, encompassing both varying preferences between users and imperfect perception or judgment. In this chapter, we are interested in viewing human-generated rankings under one or both of the following settings:

- **Imperfect perception:** The variability in rankings arise from errors in the perception of an underlying truth. By learning this distribution, we can recover the underlying truth, and discover which comparisons are noisy or cognitively difficult for users.
- **Population preference:** Different rankings arise from a distribution over the population. By learning this distribution, we discover the most common preferences in the population as well as how uniform or varied preferences are across users.

In this chapter, Sections 5.2 through 5.4 examine the practical effectiveness of different voting mechanisms as aggregation for imperfect perception in a human computation setting. Sections 5.5 and 5.6 look beyond simple rank aggregation to also understand the patterns of decision making that emerge. In contrast to the significant body of theoretical work in social choice and ranking models, we focus on three *human-generated* data sets demonstrating these two settings: preferences over types of sushi, decisions about ranking 8-puzzles by distance to the solution state, and decisions about ranking pictures of dots by number.

Section 5.3 describes a novel technique to collect realistic voting data with varying amounts of noise. We chose two domains representing human computation tasks with different properties in regard to voter noise, and our core design insight is the ability to reliably adjust the amount of implicit noise with which users perceive a known underlying ground truth. However, as the noise itself is still generated by the voters, we can compare the performance of several methods in realistic conditions.

Based on thousands of empirical rankings from workers on MTurk, we find that human agents produce very different noise than we would expect from theoretical noise models. In particular, we find that ideal ranking methods under common noise models can fare badly with real human voters, while the commonly used and easily implemented plurality rule compares favorably to other more involved methods.

We also explore the relationships between voting and general probabilistic ranking models. Using several probabilistic models of rankings, we show that a generalized random utility model (RUM) based on the normal distribution (Azari Soufiani et al. 2012), is able to better explain the variability in the data than the classical and often used Mallows 1957 and Plackett-

Luce 1959, 1975 models.

In particular, we show that the Normal RUM is significantly better in matching the empirical pairwise comparison probabilities—the marginal probability that one alternative is ranked ahead of another—in the data, and reveals interesting patterns as a result. In data over preferences, we discover users’ ubiquitous affinity or dislike for certain alternatives as well as distinguishing between conventional and more controversial items. In data about decision making, we reveal the comparisons are that harder or easier to make, and how the difficulty of a ranking task affects these comparisons. In contrast, we also demonstrate why the Mallows and Plackett-Luce models have inherent limitations in capturing heterogeneity in human-generated data. These insights, derived from more flexible models, can prompt the use of new techniques for describing human preferences and perception beyond simple rank aggregation.

5.1.1 Voting in Human Computation

Human computation is a fast-growing field that seeks to harness the relative strengths of humans to solve problems that are difficult for computers to solve alone. The field has recently been gaining traction in the AI community as interesting, deep connections between AI and human computation are uncovered (Dai et al. 2010a, Shahaf and Horvitz 2010, Kamar et al. 2012).

Reliable output from human computation generally requires an efficient and accurate way to combine inputs from multiple human agents. For example, *games with a purpose* (von Ahn and Dabbish 2008) produce useful data from many users as they play an enjoyable game, and the advent of *scientific discovery games* (Cooper et al. 2010a;b) harnessed the power of the crowd for scientific research. In EteRNA¹, players collaborate in folding RNA into its stable shape by submitting different proposals for stable designs. A subset are then synthesized in a laboratory to learn which design is truly the most stable (and to score the players). Human computation has also expanded into more general tasks with the use of *online labor markets*

¹<http://eterna.cmu.edu>

such as Mechanical Turk (MTurk; described in Section 2.1.1).

The input provided by humans via human computation systems is typically quite noisy, and beyond the setting of very simple tasks there is often a need to aggregate information into a collective choice. Naturally, this stage is often crowdsourced as well, often by letting people *vote* over different proposals that were submitted by their peers. For example, in EteRNA thousands of designs are submitted each month, but only a small number of them can be synthesized in the lab. To single out designs for the lab, players vote for their favorites, and the most popular designs are synthesized.

Voting is also frequently used on MTurk. The popular TurKit toolkit (Little et al. 2010a) is essentially a programming language for creating and managing tasks, and in particular provides an implementation of a voting function. This function receives two alternatives and a threshold as input, and posts HITs asking workers to single out their preferred alternative, until the number of votes for one of the alternatives is greater than the given threshold. To implement the common best-3-out-of-5 vote, it is sufficient to elicit three votes, and elicit more only if the first three agents do not all favor the same alternative. The authors give an example where several suggestions for things to do in New York, themselves generated by workers, are sorted using such pairwise comparisons. Little et al. (2010b) also demonstrate how human computation workflows can solve more complex problems using many iterations of voting.

However, combining many comparisons from different voters does not yield a straightforward ranking of the alternatives. So what is the best method to construct a such a ranking? Our goal is to give the first principled answer to the question:

How do the prominent vote aggregation methods compare in human computation settings?

Two research areas are well-equipped for solving this problem. First, mathematicians and economists have for centuries been studying *social choice theory*, the aggregation of individual preferences into a collective decision. In the last two decades, a field which marries computer science and social choice theory—*computational social choice*—has emerged. Most

of the work in computational social choice focuses on applying computational paradigms to social choice theory, for example, by studying the computational complexity of winner determination (Hemaspaandra et al. 1997, Conitzer 2006, Brandt et al. 2008) and manipulation (Faliszewski and Procaccia 2010, Conitzer et al. 2007, Procaccia and Rosenschein 2007) in elections. Second, the field of *utility theory* in economics has produced *discrete choice* or more general *random utility* models (McFadden 1974) that predict choices when agents are presented with different alternatives. Recent AI research has developed deeper connections between utility modeling, social choice, and machine learning (Azari Soufiani et al. 2012).

In previous work, Forsythe et al. (1996) have compared voting rules empirically, but not in the information aggregation setting, and Palfrey (2009) has conducted small-scale experiments on aggregating rank orders from voters. There is little work that applies social choice theory to computer science empirically (see Dwork et al. (2001) for one exception). Our experiments stand out in two ways: in studying human computation, we collect significantly more data; and we are particularly interested in comparisons at different levels of voter noise.

5.1.2 Social Choice Theory

A typical social choice setting has a set of n voters, $\mathcal{N} = \{1, 2, \dots, n\}$ and a set of m alternatives (or candidates), $\mathcal{A} = \{a_1, a_2, \dots, a_m\}$. Each voter i has a preference σ_i , which is a total order over \mathcal{A} . In other words, each voter ranks the alternatives. Let \mathcal{L} denote the set of all total orders over \mathcal{A} . Then, $\sigma_i \in \mathcal{L}$, $\forall i \in \mathcal{N}$. A *preference profile* $\vec{\sigma}$ is a collection of the preferences of the n agents, $\vec{\sigma} \in \mathcal{L}^n$.

Voting Rules. Social choice theorists have developed a large number of voting rules for aggregating individual preferences. Depending on whether the output is a single winning alternative or a preference ranking of all alternatives, a voting rule can correspond to a *social choice function* or a *social welfare function*.² A social choice function is a function $C : \mathcal{L}^n \rightarrow \mathcal{A}$, while a social welfare function is a function $W : \mathcal{L}^n \rightarrow \mathcal{L}$. Note that both functions receive

²In the computational social choice literature, the term “voting rule” sometimes coincides with social choice function, whereas “rank aggregation rule” is equivalent to social welfare function. We do not make this distinction here.

a preference profile as input. Any social welfare function induces a social choice function by selecting the alternative at the first position in the social preference ranking.

In this paper, we consider the following four popular voting rules:

- **Plurality:** Each voter casts a single vote for his most preferred alternative. The alternative that receives the most votes wins. If a ranking is desired, alternatives can be ranked by the number of votes received.
- **Borda:** For each voter who places an alternative at position k in his ranking σ_i , the alternative receives a score of $m - k$. The alternative that receives the highest total score wins. Alternatives are ranked by their total scores.
- **Maximin:** Let $N(a_i, a_j)$ be the number of voters who rank alternative a_i higher than alternative a_j . An alternative i 's maximin score is its worst score in a pairwise election, that is $\min_{j:j \neq i} N(a_i, a_j)$. Alternatives are ranked by their maximin scores.
- **Kemeny:** The Kendall tau distance between two preferences σ and σ' is given by

$$K(\sigma, \sigma') = \frac{1}{2} \sum_{(a, a') \in \mathcal{A}^2: a \neq a'} K_{a, a'}(\sigma, \sigma'), \quad (5.1)$$

where $K_{a, a'}(\sigma, \sigma')$ is 0 if alternatives a and a' are in the same order in σ and σ' and 1 if they are in the opposite order. Kemeny selects the ranking with the smallest total Kendall tau distance summed over all individual preferences. That is, $W(\vec{\sigma}) = \arg \min_{\pi \in \mathcal{L}} \sum_{i \in \mathcal{N}} K(\pi, \sigma_i)$. Although computing a Kemeny ranking is NP-hard, heuristics are available (Conitzer et al. 2006), and it is easily solvable for a few alternatives.

We are interested in settings where there is a true ranking of several alternatives according to quality. Each voter provides us with his own ranking of the alternatives, and our goal is to identify either the entire true ranking or its top alternative.

In the context of social choice, this is a slightly unusual setting because there are no “preferences” over alternatives, only rankings that reflect subjective estimates of the quality of different alternatives. Nevertheless, this setting is a perfect fit with the view of voting rules

as *maximum likelihood estimators*. This view was proposed by the Marquis de Condorcet as early as the 18th Century; it was picked up by Young (1988) two centuries later, and more recently studied by AI researchers (Conitzer and Sandholm 2005, Conitzer et al. 2009, Procaccia et al. 2012). The premise is that the ranking provided by each voter is a noisy estimate of the true ranking, which is generated using a known noise model, and a voting rule aggregates such noisy information. An ideal voting rule then outputs the ranking (resp., alternative) that is most likely to be the true ranking (resp., to be the true top alternative).

5.1.3 Ranking Models and Random Utility

The view of voting rules as maximum likelihood estimators falls into a broader scope of general ranking models, particularly *random utility* models in economics. Let \mathcal{M} with parameters θ denote a ranking model, generating an i.i.d. distribution on rankings: $\sigma_i \sim \mathcal{M}(\theta), \sigma_i \in \mathcal{L}, i \in \{1 \dots n\}$ denotes the i^{th} ranking in the data.

In a general random utility model, each agent obtains utility from an alternative according to an underlying deterministic value, corresponding to the true quality, plus an unobserved stochastic error, which influences the observed quality. Using the above notation, each alternative a_j has a random value (or utility) $x_j = \mu_j + \epsilon_j$, where ϵ_j is a zero-mean noise component, usually independent across alternatives, and $\mu_j \in \mathbb{R}$ is the mean value. The realized values (x_1, \dots, x_m) induce a ranking σ_i for each voter with $a_j \succ a_k \Leftrightarrow x_j > x_k$, such that voters produce preferences based on their probabilistic observed utilities for each of the alternatives. A random utility model can also be used to construct rankings or find a winning alternative. For any given distribution of stochastic errors, one may design an algorithm for inference of the most likely true quality values, which produces a ranking over alternatives in the same way as a voting rule.

Different distributions for ϵ_j correspond to different random utility models (RUMs). In the Normal RUM (Azari Soufiani et al. 2012), $\epsilon_j \sim \mathcal{N}(0, \sigma_j^2)$, where σ_j can vary across alternatives. Although a straightforward model, it has been historically intractable to use; Azari *et al.* addressed this by adopting Monte Carlo Expectation Maximization (MC-EM)

to estimate the parameters.

The classical *Plackett-Luce* model (Luce 1959, Plackett 1975) can be interpreted as a RUM in which the noise terms ϵ_j are independent Gumbel distributions with different means and the same variance (Yellott 1977). The Plackett-Luce model is popular due to its tractability. In particular, the likelihood function has a simple closed form and can be optimized efficiently with algorithms such as *minorization-maximization* (Hunter 2004).

Perhaps the oldest example of a ranking model originated more than two centuries ago, when the Marquis de Condorcet suggested a natural noise model with an intuitive interpretation: given a true ranking, a voter ranks each pair of alternatives correctly with probability $p > 1/2$. Today this model is also known as the *Mallows* model (Mallows 1957) in the statistical literature. Mallows is not a random utility model. Rather, the parameters θ define a *reference ranking* $\sigma^* \in \mathcal{L}$ and a *noise parameter* $p \in (0.5, 1]$. This model generates a random ranking by ordering all ordered pairs (a_j, a_k) in agreement with reference σ^* with probability p , and disagreement otherwise. If the result is an (acyclic) rank order than it is retained; otherwise, the process is repeated.

Voting Rules as Maximum Likelihood Estimators.

Clearly, finding true values for alternatives under a particular random utility model is an example of a maximum likelihood estimation (MLE) problem. However, we can also view voting rules as reconstructing an underlying true ranking of alternatives given noisy information. Using this perspective, there is a body of research seeking to single out voting rules that are MLEs under a model of noisy votes.

Condorcet solved the case of two alternatives, proving that plurality (known in this case as *majority*) is the maximum likelihood estimator. Moreover, as the number of voters n grows, the probability that plurality will elect the correct alternative increases; it approaches 1 as n approaches infinity (while obvious today, probability theory was in its infancy in the 18th century).

Young (1988) extended Condorcet's solution to the case of more than two alternatives.

He showed that, under Condorcet’s natural noise model, the voting rule that is most likely to output the correct *ranking* coincides with the Kemeny rule. As a result, a challenge with the Mallows model is that estimating the maximum likelihood parameters is NP-hard. Young also observed that if p is very close to $1/2$ (i.e., when there is a lot of noise), the voting rule that is most likely to select the correct *winner* is Borda. More formally, for every number of voters n and number of alternatives m there is p sufficiently close to $1/2$ such that Borda is an MLE for the top alternative. Finally, Young observed that when p is very close to 1, there are examples where Maximin is the MLE for the top alternative.

Related Ranking Models

Many extensions have been proposed to the above models in attempts to describe heterogeneity in data, such as by modeling agent-specific features or correlation between alternatives. The Mallows model has been extended to allow for mixtures (Lu and Boutilier 2011), and there are many extensions to the Plackett-Luce model, such as (Xia et al. 2008, Qin et al. 2010), which allow for a mixture of distributions or for the random utility parameters to depend on other features. The goal of these extensions has been largely to increase the models’ descriptiveness. In this chapter, we focus on simple models assuming that people are *ex ante* symmetric, and show that the Normal RUM already reveals interesting new observations about human judgment.

We implement maximum likelihood estimation for all the models previously described using multi-core parallelization, with numerical optimizations where possible. Our code is available as an integrated package³ with the aim of increasing the accessibility of using many different algorithms for modeling rank data, allowing for analysis and visualization using the methods we show in this chapter.

³<https://github.com/mizzao/libmao>

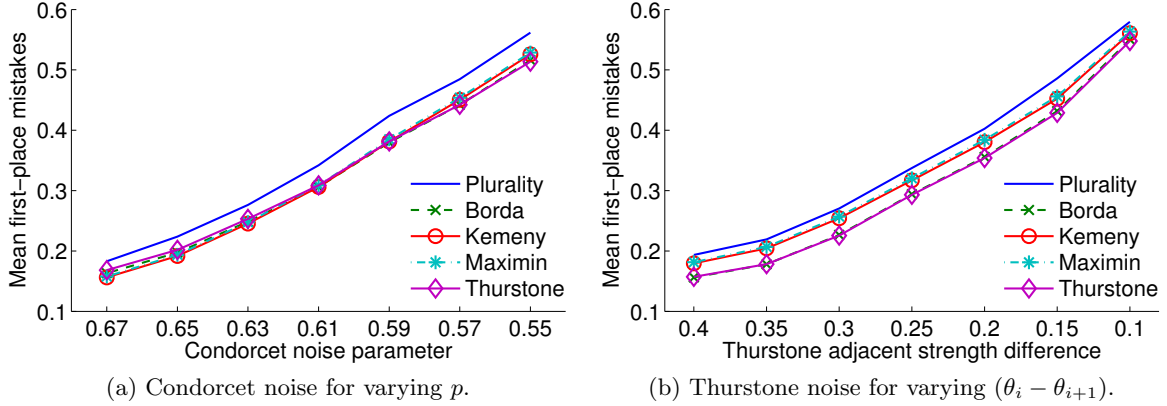


Figure 5.1: First-place mistakes at different noise levels.

5.2 Comparison of Voting Via Synthetic Data

We can confirm the maximum likelihood properties of different ranking methods under theoretical noise models by simulating preference profiles and comparing the performance of voting rules. In addition to simple voting rules, we create a voting rule from the Thurstone-Mosteller model: the model fits all pairwise comparisons over each ranking in a preference profile and estimates the strength parameters using a probit regression. We consider two types of noise: the Condorcet model with noise level p and the Thurstone model with uniformly spaced strength parameters (a fixed $\theta_i - \theta_{i+1}$ for candidates a_i and a_{i+1} in the true ranking). For each model and at each noise level, we generate 100,000 random preference profiles with 4 alternatives and 10 voters, and compute the ranking accuracy as described below.

First-place mistakes. To evaluate the voting rules' performance as a social choice function, which elects a single winner, we use the metric of first-place mistakes—the number of preference profiles where a voting rule fails to rank the best alternative at the first position. When more than one alternative ties for first place, we use the expected number of mistakes in random tie-breaking, averaging the number of mistakes across tied alternatives.

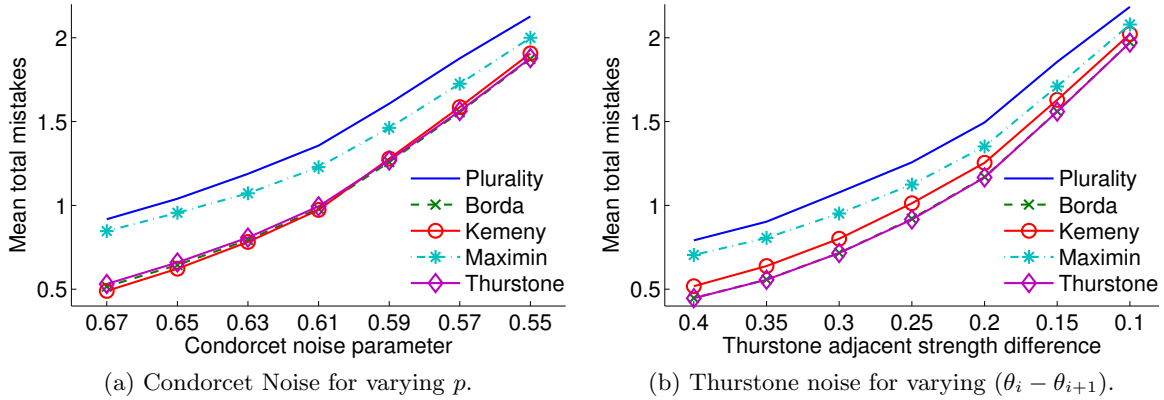


Figure 5.2: Total ranking mistakes (Kendall tau distance).

Total ranking mistakes. To evaluate how well voting rules perform as social welfare functions, we are interested in how the aggregate ranking compares against the ground truth. The standard, most natural way to measure the difference between two rankings is using the Kendall tau distance, given in Equation (5.1). The Kendall tau distance between two rankings is simply the total number of pairs on which the rankings disagree, or the number of pairwise mistakes when compared to the ground truth. Once again, for tied rankings, we average the number of mistakes to compute an expected value for random tie-breaking.

Observations. As seen in Figure 5.1, the plurality rule consistently does worse at winner determination in both types of noise. When comparing overall ranking mistakes in Figure 5.2, we observe that plurality and Maximin both perform poorly. The Thurstone rule does predictably well with normally distributed noise, but interestingly, Borda performs almost identically. Among simple voting rules, Borda is almost always the best rule in all cases, except for the Condorcet model with $p > 0.6$, where Kemeny performs slightly better. We also point out that while Kemeny is an MLE for the true ranking under the Condorcet model, it doesn't produce the fewest expected mistakes at high levels of noise; instead, Borda performs best here and this is consistent with theory when choosing the winner.

Because of the large amount of data we generated, pairwise differences between ranking methods are almost all highly statistically significant (p -values < 0.0001), except for points

1	2	3
4	5	6
7	8	

1	2	3
6		8
5	4	7

Figure 5.3: Two 8-puzzle states: on the left, the goal state. On the right, a state that requires at least 10 moves to reach the goal. The puzzle dataset consists of rankings of sets of four pictures like the one on the right.

that appear to overlap exactly in the figures.

5.3 Design of Experimental Voting Data

Do the theoretical properties observed above hold up in practice? To answer this, we first identified two voting problems with different characteristics that allowed for both a true ordering and control of ranking noise. We then designed an interface to carefully elicit ranking data from workers on MTurk, and applied the ranking methods described above.

5.3.1 Sliding Puzzles

The 8-puzzle (and its larger cousin, the 15-puzzle) has a rich history in AI research, and has often been used as a test environment for various search algorithms. However, the notion of sliding puzzles itself is much older—according to Wikipedia, it was reportedly popularized in as early as 1880 in North America⁴, and has been a well-known game for many generations ever since. Hence, this kind of puzzle game is natural and familiar to many people.

⁴<http://en.wikipedia.org/wiki/8-puzzle>

As shown in Figure 5.3, the 8-puzzle consists of a square 3x3 board with tiles numbered from 1 to 8 and an empty space. Starting from any legal board state, one solves the puzzle by sliding the tiles into the empty space to obtain a board state where the numbers are correctly ordered from top to bottom and left to right. Each movement of a single tile counts as one “move”, and the general goal is to solve the puzzle in as few moves as possible. An optimal solution to the 8-puzzle game using the fewest number of moves can be found using a search algorithm such as A^* . However, when humans play this game, they will rarely be able to find a solution in the fewest number of moves without significant effort.

Using this idea, we ask users to rank four 8-puzzles by the least number of moves the puzzles are from the solution, from closest to furthest. To collect votes at a certain level of noise, we chose a sequence of numbers, such as (7, 10, 13, 16), and generated a set of four random puzzles solvable in a corresponding number of moves as computed by A^* search. For example, for the above sequence, we would generate one puzzle (approximately uniformly over all such puzzles) that is 7 moves away from the goal, one that is 10 moves away, etc. By fixing the difference between the numbers but varying the overall distance to the goal, we make the puzzles harder or easier to rank relative to each other.

5.3.2 Pictures of Dots

The problem of counting pseudo-randomly distributed dots in images has been suggested as a benchmark task for human computation in Horton (2010). Pfeiffer et al. (2012) used the task of comparing such pictures as a proxy for noisy comparisons of items in ranking tasks. We use this latter setting as the basis of voting in our experiments; this task is also easy to explain and requires minimal understanding to complete.

This task of comparing dots is also easy to explain and requires minimal understanding to complete. Each voting task involved sorting four pictures from fewest dots to most dots, There were many more dots than could be easily counted; to control the level of noise, we varied the difference in the number of dots among each set of pictures. For example, Figure 5.4 shows pictures with 200 and 208 dots, respectively. A larger difference is easier to detect,

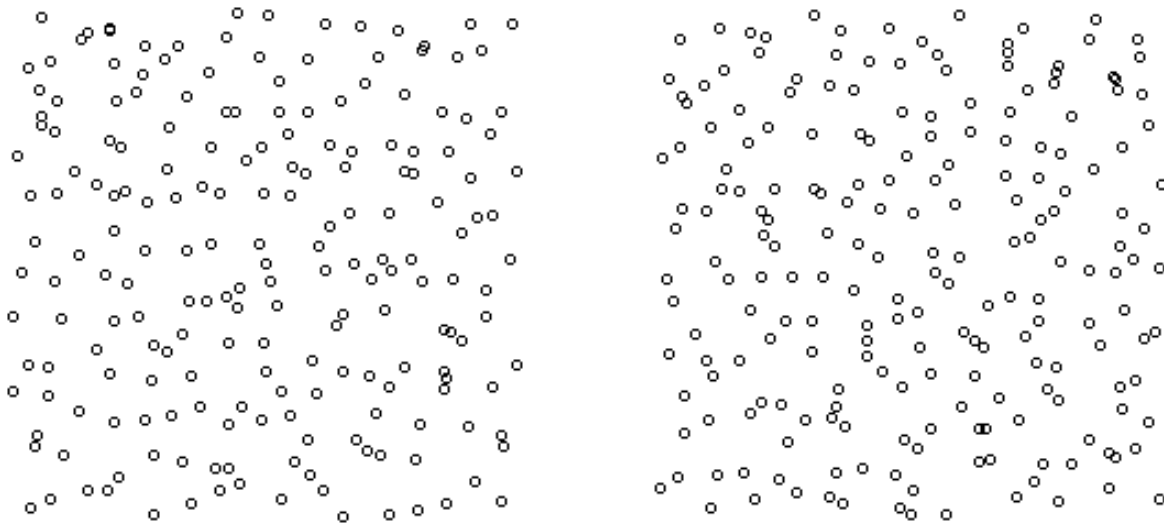


Figure 5.4: Two pictures of dots: on the left, a picture with 200 dots. On the right, a picture with 208 dots. The dots dataset consists of rankings over sets of four pictures such as these.

and therefore less noisy, than a smaller difference.

We took additional precautions to minimize idiosyncratic variance in the task. For example, to generate an image with 207 dots, we divided a square two-dimensional space into 207 identically-sized subregions and positioned a single dot uniformly at random in each region. The result was a picture with the prescribed number of dots arranged randomly but uniformly over the entire area, and with low probabilities of many dots overlapping with each other, which would increase the variance of comparing two pictures. Additionally, the pictures were presented in bitmap format as opposed to the commonly used JPEG or PNG formats, which would allow comparisons by users looking at image file size.

5.3.3 Comparison of Domains

We chose the two domains to represent different types of human computation tasks. When ranking 8-puzzles, there are many ways to approach the problem using heuristics or other solution methods, and we expect (and observe) that workers will expend varying amounts of effort on the task; this is a proxy for tasks where the expertise or quality from workers is very different. However, in the case of counting dots, we took care to ensure that better

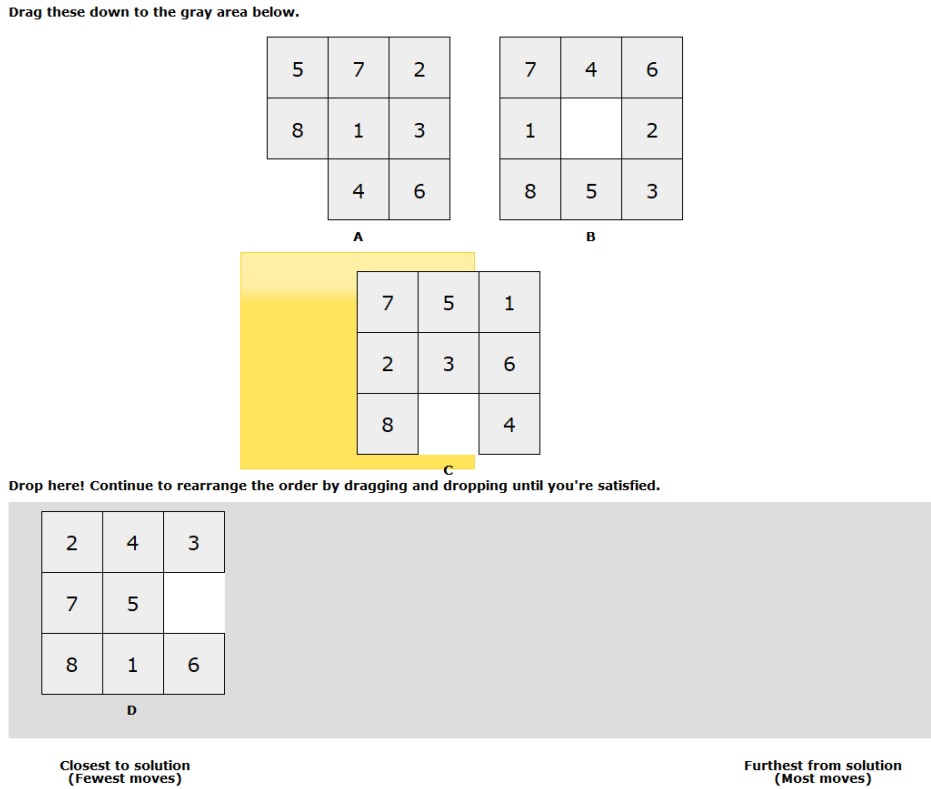


Figure 5.5: The experiment interface. Voters drag and drop the objects from a square arrangement into a sequence of their preferred order.

accuracy is more insensitive to additional effort: the noise from a group of voters will be more homogenous, and this setting gives us an understanding of tasks where input from many voters will be of more equal importance.

5.3.4 Methodology

Our evaluation tests how different ranking methods fare when applied to noisy collective estimates of a ground truth, comparing them across different levels of noise.

Interface. The core of our interface is an elicitation mechanism for voters to indicate their ranking of alternatives. To collect reliable data for each ranking problem and at different noise levels, we designed our experiment interface carefully, using randomization to reduce behavioral artifacts or potential for bias.

Figure 5.5 shows the interface displaying four 8-puzzles. The objects were presented in

randomized order in a square ‘starting’ grid to workers. Below this was a linear ‘target’ area with suggestive text anchors at both ends where workers had to order all the alternatives. Objects could be picked up and inserted at any point using drag-and-drop. Moving alternatives to the target area forced workers to make a decision about each one relative to the others; moreover, by randomizing the initial set of objects and arranging them in a square, we removed any bias suggested to low-effort workers by an initial ordering.

We paid \$0.10 for each HIT, which consisted of ranking one set of four objects. Each task began with a basic description of the task, followed by a short quiz to check that users understood how they were comparing puzzles or pictures. Additionally, we enforced a limit of 5 HITs per user per daily period, and ensured that no user saw the exact same set of objects twice.

Experiment Parameters. After first conducting some initial trials on both data domains, we selected an appropriate level of noise for each set of experiments. For the 8-puzzles, we created puzzles corresponding to the following sets of distances from the goal state:

- (5, 8, 11, 14) — Easiest
- (7, 10, 13, 16)
- (9, 12, 15, 18)
- (11, 14, 17, 20) — Hardest

For the dot comparisons, we generated pictures containing the following numbers of dots:

- (200, 209, 218, 227) — Easiest
- (200, 207, 214, 221)
- (200, 205, 210, 215)
- (200, 203, 206, 209) — Hardest

For each domain and at each level of noise, we generated 40 sets of objects, then collected approximately 20 preference rankings on each set. In other words, we collected 40 *preference profiles* per sequence with 20 voters each, for a total of 3,200 rankings for each type of task.

5.4 Comparison of Voting Via Human Data

We tested the accuracy of the methods described above in teasing out the correct top alternative and ranking, and computed comparable results to the synthetic data in Section 5.2. This data averages the number of mistakes over random subsets of the 40 preference profiles—at each noise level, all voting rules are applied to the same randomly sampled preference profiles consisting of 10 voters. This approach simulates the effect of having fewer voters (and more noise) in aggregation, but also reduces the amount of statistical variance across rules so that they can be compared. Averaging the differences in number of mistakes also creates a normal distribution, allowing for use of a paired t-test for statistical significance.

First-place mistakes. Figure 5.6 shows the average first-place mistakes for all the ranking methods. The mean number of first-place mistakes for all rules increases as the noise level increases for both ranking problems. This confirms our premise that varying the distance to the goal state in the 8-puzzle or the difference in dots across pictures changes the noisiness of the votes collected. We observe that the Borda rule, predicted to do well in theory, consistently has among the highest number of mistakes; at several points the pairwise difference is significant at the 0.01 level. Meanwhile, the commonly used plurality rule performs much better than in theory, never emerging with the worst score.

Total ranking mistakes. Figure 5.7 shows the mean Kendall tau distance from the ground truth to the voting rules we tested. Once again, we see that the total number of mistakes for all voting rules rises according to increasing levels of noise. In this case, we also see a distinct difference between plurality and Maximin versus the other voting rules—both have noticeably higher errors when used to construct a ranking (almost all of the differences are significant

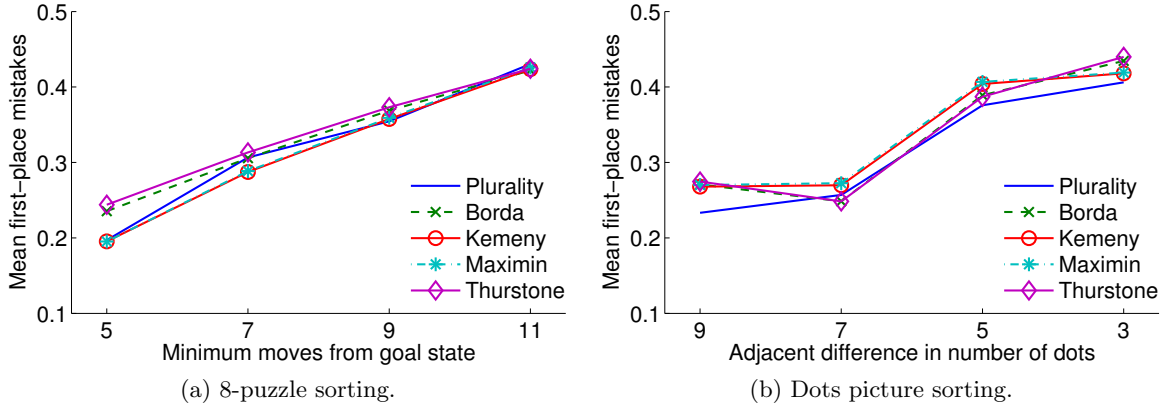


Figure 5.6: First-place mistakes at different noise levels.

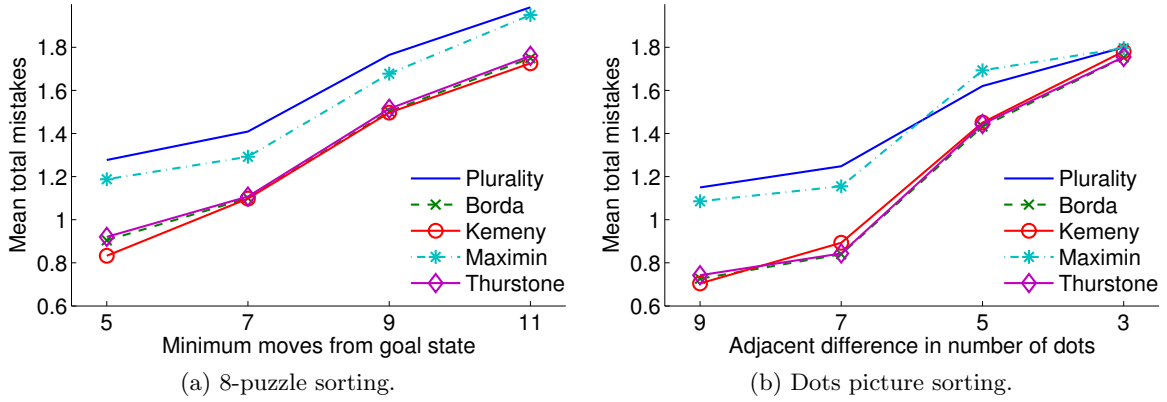


Figure 5.7: Total ranking mistakes (Kendall tau distance).

at the 0.01 level), confirming what we observed in the theoretical models. However, among the other rules, there is no clear winner.

Observations from users. Our entire dataset, including initial exploratory data, consists of 8,529 individual rankings from 1,693 unique voters, including approximately 6,400 rankings from 1,300 unique users in the final evaluation. At \$0.10 per ranking, we collected a large amount of data from a very diverse population in a very economical way.

We also asked users about how they approached comparing puzzles. While the individual heuristics of voters do not affect how the voting rules compare, it was interesting to observe different approaches. As expected, the majority of users compared puzzles mentally and did

the task quickly, but some also tried to solve the puzzle using pencil and paper, or went even further by constructing a physical 8-puzzle and sliding the pieces around. Others computed what was essentially the Manhattan distance heuristic (an admissible heuristic for A^* search) for each puzzle. A few workers went all the way, wrote code to solve each puzzle, and entered in the minimum number of moves in their comments, even though this was far beyond what we requested. These varying user effort levels are only natural when it comes to human computation settings.

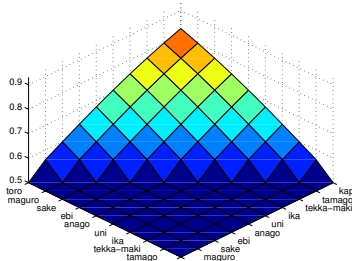
In our dot comparison tasks, we did not observe any user comments alluding to significant differences in strategies. In particular, voters often commented that they were unable to come up with new ways to solve the task, and desired feedback on their performance. This supports the idea that there was not much beyond simply eyeballing the pictures that could differentiate between them.

5.5 Variation and Uncertainty in Ranking Data

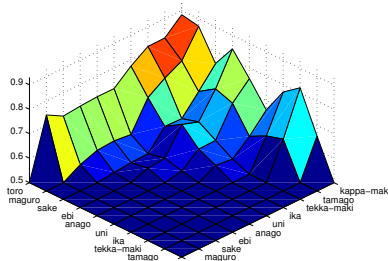
The observations in Section 5.4 suggest that ranking models that accurately represent human behavior can be useful for aggregating noisy information. However, such models can be equally useful for describing variation in preferences across a population, even in the absence of a ground truth. In this section, we explore the distinctions between the two different types of applications.

5.5.1 Sushi Ranking Dataset

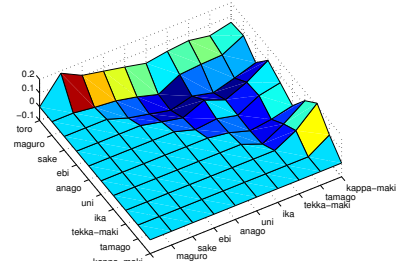
Kamishima (2003) collected data on the rank order preferences of restaurant customers in Japan for different types of sushi. A particularly interesting subset of this data are a collection of 5000 rank orders on the same 10 pieces of sushi: *ebi* (shrimp), *anago* (sea eel), *maguro* (tuna), *ika* (squid), *uni* (sea urchin), *sake* (salmon roe), *tamago* (egg), *toro* (fatty tuna), *tekka-maki* (tuna roll), and *kappa-maki* (cucumber roll). These rankings, each provided by a unique customer, are an example of *preference* data. By accurately modeling the distribution of rankings, we can describe the distribution of preferences over the population, and ultimately



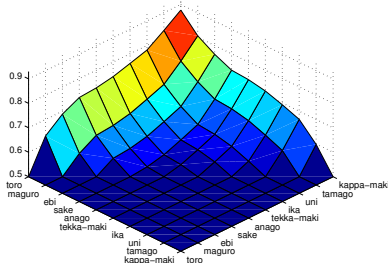
(a) Mallows pairwise probabilities. Negative log likelihood: 71353.



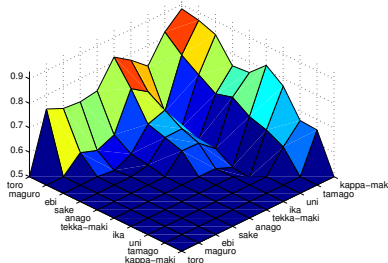
(b) Empirical probabilities in order.



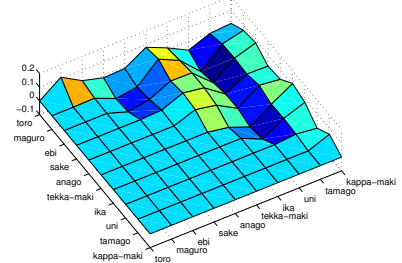
(c) Mallows deviation.



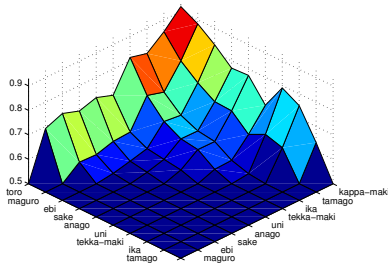
(d) Plackett-Luce pairwise probabilities. Negative log likelihood: 71211.



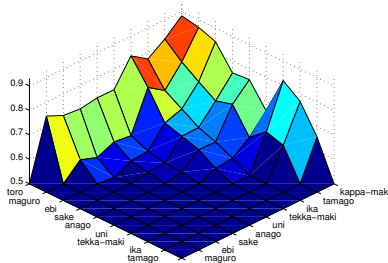
(e) Empirical probabilities in order.



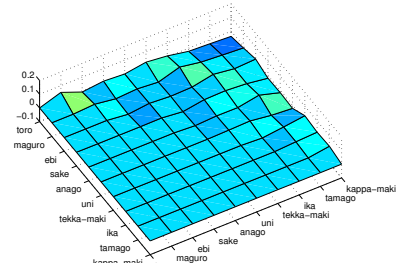
(f) Plackett-Luce deviation.



(g) Normal RUM pairwise probabilities. Negative log likelihood: 69011.



(h) Empirical probabilities in order.



(i) Normal RUM deviation.

Figure 5.8: Comparison of different models for the sushi data: each row shows one model with axes of plots arranged in the model's modal ordering. The first column shows the model's predicted pairwise comparison probabilities (for an item on the left axis to be ranked above an item on the right axis). The second column shows the empirical pairwise comparison probabilities. The last column shows the difference of the two. For clarity and because of symmetry, we plot one probability for each pair of items.

produce a succinct depiction of collective preferences.

Azari Soufiani et al. (2012) originally showed that the Normal RUM has better model fit on this dataset and others; we additionally demonstrate why this model can better describe collective preferences than the Plackett-Luce and Mallows models. We focus on the pairwise comparison probabilities—the marginal probability that one alternative is ranked above another alternative. Pairwise comparisons measure the first-order accuracy of the model and have been used since the earliest statistical ranking models (Mosteller 1951, Bradley and Terry 1952).

Figure 5.8 compares the aggregated pairwise comparison probabilities to the empirical data, with one model in each row, as surface plots of $m \times m$ matrices. The first column shows the model’s prediction according to maximum likelihood parameters. The second column shows the empirical probabilities, and the third column shows the element-wise difference. As each model implies a different modal (most likely) ordering of the alternatives, the plots in each row show the implied modal ordering along the axes. For the sake of clarity, and recognizing symmetry, we show only one comparison between each pair of items in the plots. A continuous color scale indicates the magnitude of each value, and in the third column, a flatter plot with more uniform color indicates a better fit between the model predictions and empirical comparisons.

The Mallows model, shown in the top row, can only fit a surface of probabilities derived from the single parameter p , which monotonically increases for alternatives that are further separated in its modal ranking (Figure 5.8a). Compared to empirically observed comparisons, the Mallows model suffers from both systematic over- and under-estimates of pairwise probabilities (Figure 5.8c). In the second row, the Plackett-Luce model, while more flexible than the Mallows model, predicts comparison probabilities that are constrained by monotone increases along its modal ordering (the axes of Figure 5.8d), and still shows significant deviations from the data (Figure 5.8f).

The third row illustrates the predictions of the Normal RUM. Due to flexible variance parameters, one per alternative, the model is no longer subject to the monotonic behavior we

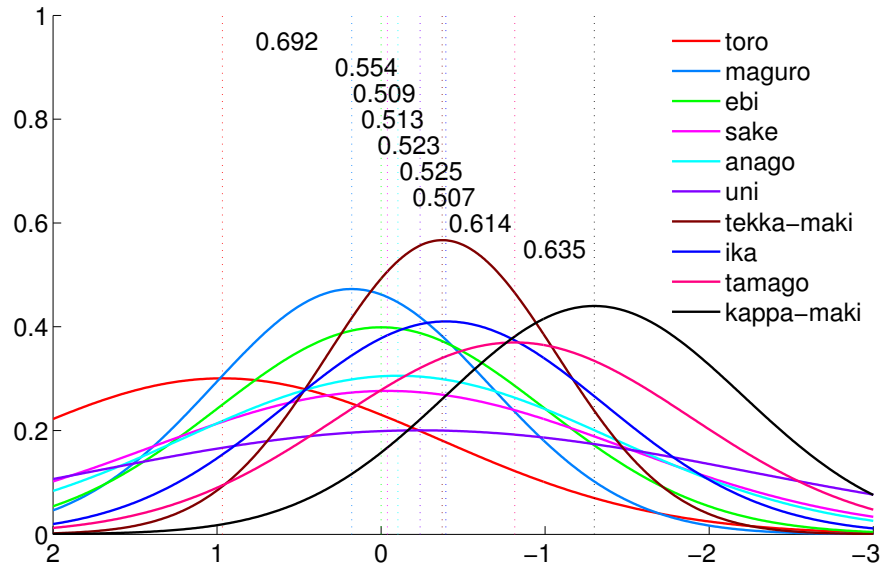


Figure 5.9: Distribution of random utilities for the different types of sushi in the estimated Normal RUM. Values show the probabilities of two adjacent sushi in the modal ordering to be ranked in that order. For example, 0.692 is the predicted probability that *toro* is ranked ahead of *maguro*, and 0.554 is the predicted probability that *maguro* is ranked ahead of *ebi*.

observed previously (Figure 5.8g), and bears a much closer resemblance to the empirical probabilities (Figure 5.8h). The Normal RUM is able to fit the pairwise, empirical probabilities very closely compared to the other models (Figure 5.8i is much flatter).

Since these models attempt to capture a symmetric distribution over the population and not predictions for each person, the increased explanatory power of the Normal RUM is not due to overfitting: the model consists of only 20 parameters, while the data contains 5000 permutations of 10 alternatives.

Model Interpretation

As the Normal RUM achieves a much more accurate estimate of marginal pairwise probabilities in the data, we now turn to interpretation of its distribution. Figure 5.9 plots the estimated, random utility distributions for each alternative. For identifiability in estimation of the model, an arbitrary distribution (*maguro*) is fixed to the standard normal distribution. Note that the x-axis is reversed, with larger values to the left. Under the model, each

consumer’s preferences are represented by an independent draw of random values from each of the the distributions, and ranked according to the realized values. The plot also shows the predicted, marginal pairwise probabilities that adjacent pairs in the modal ordering are ranked according to that ordering.

This interpretation yields a great deal of information about the preferences of sushi consumers. First, the most preferred (*toro*) and least preferred (*kappa-maki*) show a clear separation from the rest of the alternatives, and are separated from their immediate neighbors with a much greater probability than other adjacent pairs in the ranking. This shows greater universality in the like or dislike of these alternatives.⁵ In contrast, preferences over adjacent items in the middle of the ranking are more noisy and comparison probabilities are close to 0.5 for many pairs.

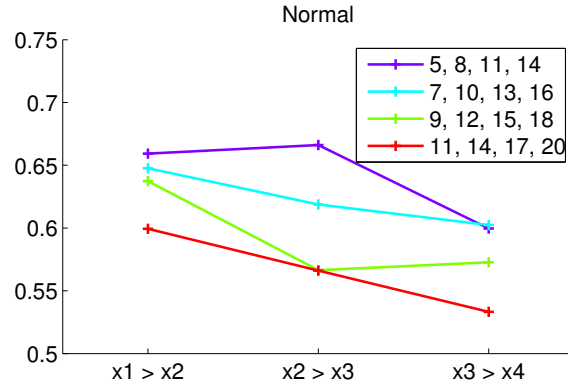
The variance of the distribution for each type of sushi is also informative. The greatest variance is for *uni*, while *tekka-maki* (tuna roll) has the lowest variance, revealing a dichotomy of preference for the former and a more consistent but average support for the latter.⁶ This illustrates how the Normal RUM allows a very large set of ranking data to be distilled into an intuitive explanation of the population. Notably, this interpretation would not be possible with the widely-used Mallows and Plackett-Luce models, whose parametrizations cannot capture the differing variance across items and are thus more inaccurate in representing the diversity of preferences. We discuss these model deficiencies in further detail in Section 5.6.

5.5.2 Voting Decision Data

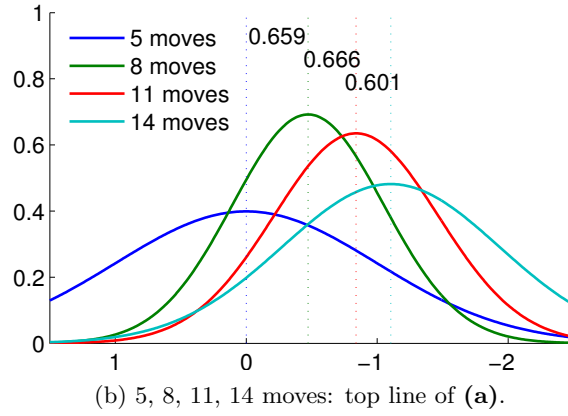
Model Interpretation In contrast to the sushi data described in Section 5.5.1, the voting datasets described in Section 5.3 do not capture different preferences across individuals. Instead, every user has the same preferred ranking over the data, but variance arises from *imperfect or noisy perception*. Thus, by learning a distribution of rankings over the data, we can describe decision-making ability and cognitively difficult comparisons across the collective

⁵ *Toro* is a fatty cut from the belly of the bluefin tuna that is especially highly regarded, and invariably commands a premium price in sushi bars. *Kappa-maki* is a perhaps unremarkable sushi of cucumber and rice.

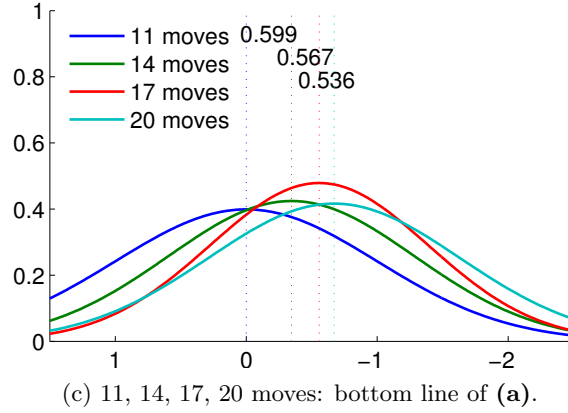
⁶ In contrast to the conventional *tekka-maki* (tuna roll), *uni* is a rather unique type of sushi made from the gonads of the sea urchin, known to elicit delight or disgust depending on a person’s taste.



(a) Adjacent pairwise probabilities.



(b) 5, 8, 11, 14 moves: top line of (a).

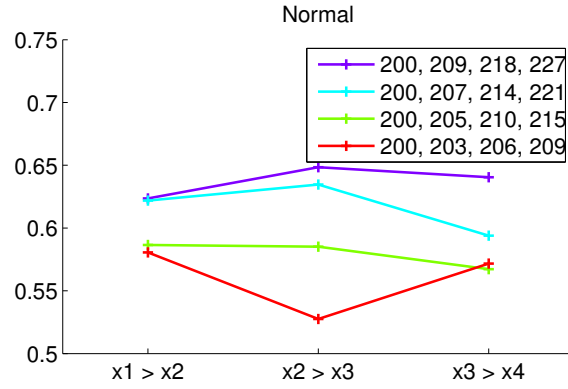


(c) 11, 14, 17, 20 moves: bottom line of (a).

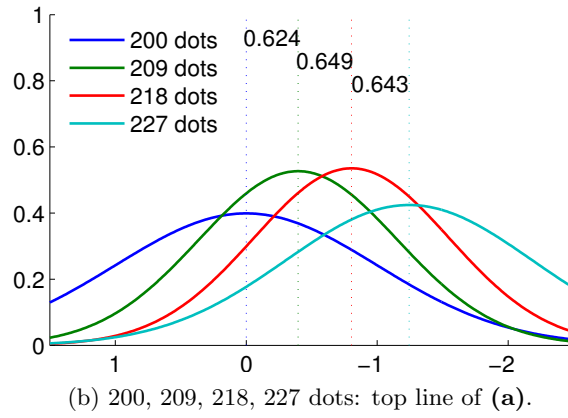
Figure 5.10: Aggregate fitted results for the normal model on the 8-puzzle rankings. Dashed lines indicate mean strength. Three numbers in each plot show probabilities of adjacent pairwise probability implied by the model.

population.

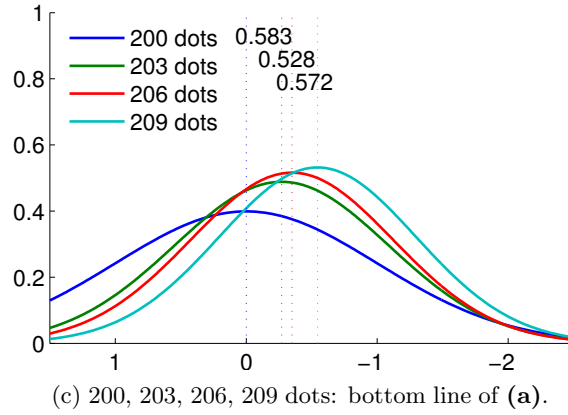
As in the sushi dataset, the Normal RUM provides the best approximation of empirical comparison probabilities on this data. We focus on the estimated parameters to gain an



(a) Adjacent pairwise probabilities.



(b) 200, 209, 218, 227 dots: top line of (a).



(c) 200, 203, 206, 209 dots: bottom line of (a).

Figure 5.11: Aggregate fitted results for the normal model on the dots image rankings. Dashed lines indicate mean strength. Three numbers in each plot show probabilities of adjacent pairwise probability implied by the model.

understanding of human perceptive judgment in these two domains.

Figures 5.10 and 5.11 show the properties of the estimated random utility distributions for each problem. The first graph shows the pairwise comparison probabilities for adjacent

alternatives at each level of difficulty. The second and third plots show the estimated random utility distributions for the easiest and hardest ranking problems, respectively. In each case, the distributions are scaled so that the mean and variance of the highest-ranked item (closest puzzle to the goal or lowest number of dots) has the standard normal distribution.

In both problems, adjacent pairs of items become harder to rank as difficulty increases, with comparison probability decreasing toward to 0.5—shown by the progression of successively lower line segments in Figures 5.10a and 5.11a. Yet, it is particularly interesting how the varying difficulty of the two different problems affects users’ judgments, as revealed from the estimated model. For the 8-puzzle, the probability of correctly ordering adjacent puzzles slopes downward from left to right, showing that is harder to compare two puzzles that are further from the goal state than two that are closer. This is naturally explained by observing that it is easier, for example, to tell which of two puzzles that are 7 and 10 moves from the solution is closer, than two puzzles that are 13 and 16 moves from the solution. All levels of difficulty in the problem exhibit this property.

For the fields of dots, there is a different behavior as difficulty increases. At the easier levels of difficulty, the probability of ranking adjacent alternatives in the correct order does not significantly slope from left to right for the three adjacent pairs. This is quite different from the 8-puzzle setting, but is naturally explained considering that dot fields do not become harder to rank when the difference of dots is a similar percentage of the total (around 200 for all pictures). However, in the most difficult setting, with a difference of only 3 dots between pictures, there is a marked drop in accuracy for the intermediate two pairs of pictures.

In the distribution over rankings, we see a similar pattern to the sushi dataset: namely, there is more certainty about the pictures with the least and most number of dots (resp. stronger preference of the favorite and least favorite sushi) than among the intermediate choices. In the dots data, the lower certainty arises from increased perceptive error between the extremes, while in the sushi data this manifests from more varied preferences apart from the favorite and least favorite. The Normal RUM captures both of these cases, allowing it to describe both rankings of preference and rankings of imperfect perception.

As each set of parameters is generated from 800 rankings and the patterns we observe are consistent across the difficulty levels for both problems, we believe these results are robust. Compared with the parameters estimated for sushi data, the variance of the noise distributions for alternatives is more uniform. However, the expressiveness of the Normal RUM allows similar variances across alternatives to be interpreted as more or less uniform perceptive error in ranking, rather than a necessary limitation of the model (as is the case with Plackett-Luce: see the next section).

5.6 Properties of Ranking Models

Sections 5.5.1 and 5.5.2 showed that the Normal RUM allows a better interpretation of several data sets than the classical Mallows and Plackett-Luce models, because of its advantages in correctly capturing comparison probabilities over all pairs of alternatives. Here, we show analytically why this generally can be true for any data set, due to inherent restrictions in the classical models.

By estimating a ranking model \mathcal{M} (i.e., by using maximum likelihood), we obtain parameters $\hat{\theta}$ implying a distribution on rank orders as well as a marginal distribution over pairwise comparisons. We generally expect that the probability of users ranking one particular item above another will be only marginally affected by the presence of other choices, so we can evaluate the suitability of a model by how well it approximates all marginal pairwise probabilities. More generally, pairwise comparison data has a long history of use in learning rankings (Mosteller 1951, Bradley and Terry 1952) and is generally viewed as more robust than cardinal data (Ammar and Shah 2011).

Under the Normal RUM, the probability that a particular alternative a_j is preferred to another item a_k is

$$\Pr_{\text{Normal}}(a_j \succ a_k \mid \boldsymbol{\mu}, \boldsymbol{\sigma}) = \Phi\left((\mu_j - \mu_k) / \sqrt{\sigma_j^2 + \sigma_k^2}\right) \quad (5.2)$$

where Φ is the CDF of the standard normal distribution. Figure 5.8, showed that this

two-parameter model closely approximated all pairwise comparisons in a large set of empirical preference data.

Limitations of the Mallows Model Under the Mallows model, the noise parameter p is also the probability that adjacent pairs in the reference ranking σ^* are ranked in the same order. More generally, any two items separated by a fixed distance in σ^* are ranked correctly with the same probability $\Pr(a_{\sigma^*(k)} \succ a_{\sigma^*(l)})$ for positive $z = l - k$, which can be shown to be

$$\Pr_{\text{Mallows}}(a_{\sigma^*(k)} \succ a_{\sigma^*(l)} \mid \phi, \sigma^*) = \frac{\sum_{z=1}^c z \phi^{z-1}}{(\sum_{z=1}^c \phi^{z-1})(\sum_{z=0}^c \phi^z)} \quad (5.3)$$

with $\phi = (1 - p)/p$. As shown in Figure 5.8a, this is monotone increasing in c for a given ϕ . In the context of rankings from human input, this rather strong assumption is easily violated, such as when agents have more certain comparisons over adjacent pairs in one part of the ranking than the other. In the case of the sushi data and the most difficult dots ranking problem, this occurred at endpoints of the ranking with more uncertainty in the middle. Hence, while popular and extended in many ways as outlined in Section 5.1.3, the Mallows model is inherently rather restrictive.

Limitations of The Plackett-Luce Model Under the Plackett-Luce model, a rather severe restriction is the fixed variance of the Gumbel distribution for the random utility across alternatives. The marginal probability that alternative a_j ranked is higher than alternative a_k is

$$\Pr_{\text{PL}}(a_j \succ a_k \mid \boldsymbol{\mu}) = 1 / \left(1 + e^{-(\mu_j - \mu_k)} \right) \quad (5.4)$$

Since the logistic sigmoid $g(x) = 1/(1 + e^{-x})$ is strictly monotonically increasing, there is a strict limitation in which pairwise probabilities for ordering items in a particular way are determined only by the difference in their strength values. Specifically, for any strictly monotone increasing function $f(x)$, we can show that for any fixed μ_j ,

$$\mu_k > \mu'_k \iff f(\mu_j - \mu_k) < f(\mu_j - \mu'_k) \quad (5.5)$$

This behavior is exemplified in Figure 5.8d, where the comparison probabilities monotonically increase along the axes, and emerges in all fixed-variance (one-parameter) random utility models.

As we have seen, this assumption is rather strong, and any RUM with this property cannot capture the notion that it may be less certain to compare some particular alternative (such as *uni*) versus others, and hence it would be sensible for the variance of utility for that alternative to change rather than the mean value.

5.7 Discussion

Our results indicate that in realistic human computation settings, there can be significant differences between various methods of aggregating votes. As our results are consistent across two different domains, we believe that they have robust implications for voting in noisy settings. For choosing a ranking, the Borda rule stands out as both simple and accurate: in both theory and experiment, Borda performed as well as Thurstone model—a surprising result given that the latter is a numerically complex probit regression. Our results from real data also support the common use of plurality for the selection of a single alternative, especially since it requires eliciting only one vote instead of a ranking.

Our empirical results (Section 5.4) stand in contrast to simulations based on the Condorcet and Thurstone noise models (Section 5.2), where plurality consistently performs poorly. We conclude that the most prominent theoretical noise models are not necessarily good predictors for the performance of different methods on human computation data. It is to be expected that the 8-puzzle data differs from the simulated data, because there the voters are far from being i.i.d. (see Section 5.4). However, the difference is more surprising when considering the dots data, where we expect workers to be similar in terms of their ability and time investment.

Nevertheless, we believe that theory and models can play an important role. One of our main contributions is our experimental methodology, which allowed us to collect a massive number of high-quality variable-noise votes. This unprecedented dataset⁷ facilitates an easy

⁷The dataset consists of voter rankings as well as the puzzle sequences and dot images used to generate

comparison of newly suggested, perhaps tailor-made, voting rules with existing techniques.

In particular, our results also illustrate the importance of flexible and expressive models for fitting human ranking data, encompassing both variation across *population preference* as well as errors arising from *imperfect perception*. Commonly used but restrictive models such as Mallows and Plackett-Luce are insufficient to capture the various patterns encountered in human perception. However, an appropriately descriptive model such as the Normal RUM allows for intuitive interpretation of both types of data in a more nuanced way.

Such models allow us to gain new understanding and intuition about real data. We are able to see clear preferences for certain alternatives by users, as well as which choices present more contention or uncertainty and which are uncontroversial. We can also see how users' perception and decision making is affected by harder and easier tasks, and for which alternatives they can make judgments that are either more certain or more ambiguous. Our approach also highlights the importance of detailed evaluation techniques that reveal the quality of a model's fit to data in a natural way, such as looking at pairwise comparison probabilities. Our results clearly show how classical models fall short in representing human ranking data.

Our results also motivate several areas of future work. The Normal RUM and other flexible random utility models are promising for discovering interesting patterns in preferences across a collective group of people. In our case, estimating such a model distilled several thousand rankings into a much more concise representation. We believe that this type of analysis facilitates a more natural understanding of collective preferences over simple rank aggregation. At the same time, we foresee that a better understanding of the difficulty and variance in human judgment problems can motivate the design of better user interfaces and crowdsourcing systems. Using any ranking data, one can explore errors and variance when human users are asked to make comparisons that are noisy or uncertain.

In the long run, we hope that our work will spark an interaction between researchers in human computation, computational social choice, and machine learning that will lead them.

to the design of better human computation systems via more principled voting and ranking techniques, and more insightful descriptions of collective human preferences as well as a better understanding of decision making in the design of systems involving human agents.

5.8 Acknowledgments

The research in this section was produced through thoughtful collaboration with Hossein Azari Soufiani, Yiling Chen, David Parkes, Ariel Procaccia. Portions of this chapter previously appeared in the AAAI publication *Better Human Computation Through Principled Voting* (Mao et al. 2013c).

Chapter 6

Design and Implementation of a Web-Based Experimental System

The use of laboratory experiments has led to procedures and guidelines for conducting experiments and ensuring the quality of experimental data. Such standards are also developing for web-based experiments. While lab experiments are often conducted with software such as the popular zTree (Fischbacher 2007), web-based frameworks present both new challenges and new opportunities for experiment designers. This chapter details various components of the TurkServer¹ experimental platform and how they enabled the collection of experimental data described in earlier chapters. We hope that the features of TurkServer will become standard techniques for future software-based experiments.

6.1 Deploying Web-based Experiments

While web-based experiments offer many opportunities to conduct experiments that are impossible in a physical lab or even in field settings, they also present a set of challenges and common pitfalls. It is important for the experiment designer to control for as many extraneous variables as possible, since a cleaner experiment design allows for better quality data and stronger measurements of the effect of experimental treatments.

¹<https://github.com/HarvardEconCS/turkserver-meteor>

6.1.1 Advantages of Online Experiments

There are several strong advantages that online lab-style experiments have over physical labs (Reips 2000; 2002). These are described in more detail in Section 2.2. In summary, these include the following.

Larger subject pool. Physical lab experiments are constrained by local availability of participants, typically undergraduates at a university. This limits both the number of unique participants available over a long period of time.

Simultaneous participation. Physical lab constraints also affect the number of people that can participate simultaneously in a particular experiment, limiting the ability to study interaction within a large group of people.

Recruitment and on-demand participation. Moreover, physical lab constraints also affect the ability to recruit many participants without planning ahead. Constraints imposed by other simultaneous experiments limit the number of subjects available, and lab resources require many sessions to reach a large sample size. This imposes an implicit limitation that the cycle of experiment design, hypothesis testing, and refinement is relatively slow. In particular, this can be quite painful if a particular experiment reveals new insights that require more studies to test. In contrast, Gao et al. (2014) show that the cycle of iteration can be quite fast for online studies.

More diverse subject pool. Physical lab studies typically make strong use of university undergraduates, leading to the observation that much of human behavior is being extrapolated from WEIRD (Western, Educated, Industrialized, Rich, Democratic) participants (Henrich et al. 2010). On the other hand, recruiting participants from the Internet allows for a much broader sampling of culture and ethnicity, allowing for cross-cultural studies that would typically require much more effort as a traditional field experiment, e.g. in Suri et al. (2011).

Better balance of internal/external validity. While physical experiments often face a clear choice between a lab and field setting, the online environment reaches all users through a similar interface—their computer. Web software can be designed to both provide procedural control of an experiment while allowing for a natural environment and hence generalizability. As a result, it is possible for the experimenter to choose among multiple desirable options for an experiment design that straddle the traditional field versus lab distinction.

6.1.2 Challenges for System Implementation

While physical lab experiments and field experiments typically have standardized equipment in the form of computers and other procedures, online experiments are deployed to a variety of bespoke hardware and network connections of varying capabilities. Typically, this means that experiment software must support a range of operating systems and web browsers while minimizing the effects of network latency and even disconnection during a particular study. These issues are particularly acute for experiments allowing for real-time interaction between users, as non-responsive participants can greatly affect the resulting data collected.

In general, research studies that deploy software to a broad audience must face similar technical problems to those of web software and mobile app developers in general, but often with fewer resources. Section 6.2.1 provides some suggestions for minimizing potential issues.

6.1.3 The Experiment Design Triangle

The advances in scale offered by online experiments, despite significant are not unlimited. There are three main goals that are desirable in experimental design.

- **Simultaneous recruitment.** An obvious attribute of having many users available at the same time is the ability to study interaction between users. However, being able to schedule a large “session” of users at once helps to achieve proper randomization over many experimental treatments, especially when each treatment itself has potentially many participants as in Chapter 4. Additionally, it also helps to avoid time-of-day

effects, e.g. when mid-day users are demographically different from evening users and thus display different behavior.

- **Participant Uniqueness.** By preventing repeat participation, we can avoid priming effects or other behaviors where the participants bring some expectation of behavior into the study different from someone who has just read the instructions.
- **Large sample size.** A desirable feature to increase the power of the experiment by averaging over unobserved variation between experimental units.

On typical crowdsourcing systems where only a few thousand workers are available at a given time, it is typically impossible to achieve all three objectives simultaneously, resulting in an *experiment design triangle*.² Any one of these restrictions can be relaxed to make an experiment design significantly easier.

By not requiring a large sample size, we can still recruit several very large sessions with unique participants as in Chapter 4, but very quickly exhaust the pool of active users. This design is more acceptable if the treatment effects are hypothesized to be large.

By relaxing the constraint for users to be unique, we can allow repeat participation and pump many (of the same) users through the experiment. This design requires an experiment that is insensitive to learning effects, such that new users and those who have participated many times are not expected to behave significantly differently. Alternatively, the analysis can be modified to account for learning in users, although this can complicate the observation of causality in experiment design.

By not requiring simultaneous recruitment, an experiment can still reach a large number of unique users simply by interacting with them over a longer span of time. This is the typical arrangement of most experiments on MTurk, conducted with single users. However, this design can still be done with small groups of users if the randomization is sufficiently robust: the design of Gao et al. (2014) is a good example.

²As in the *project management triangle*— “fast, good, or cheap: pick two”.

Ultimately, these issues can only be resolved with a very large participant pool, possibly at the regional, national, or international level, or with more nontraditional participant recruitment methods as discussed in Halberda et al. (2012) or Reinecke and Gajos (2014).

6.1.4 Participant Comprehension and Attention

Participants in web-based experimental studies typically may often be splitting their attention with other demands. This affects two aspects of experiment design: *comprehension* of instructions and *attention* during the task itself.

In traditional lab studies, instructions are usually given in a standardized procedure, either for participants to read by themselves, or during a presented briefing by an experimenter. Because the participants are in an enclosed environment with no distractions, they are generally compelled to understand the instructions in the absence of any other demands for attention. On the other hand, online participants may be at home or at work and dividing their attention with other activities. As a result, poorly designed instructions may be read less carefully or ignored, with the result that the participant simply muddles through the task.

Another concern is attention during the task itself. Participants at home or work may have other errands or responsibilities, and the likelihood of interruptions increases with the length of the task. Experiment designs that require real-time interaction can produce very different results when some participants are inactive for a long period of time. As a result, it is prudent to explicitly consider the trade-off between level of attention required and the length of a task, and design user interfaces to maximize intrinsic interest and improve both characteristics.

6.1.5 Limitations of Amazon Mechanical Turk

As Amazon Mechanical Turk (MTurk) is currently the most prominent platform for designing many types of web-based experiments, it is worth discussing its intricacies in particular.

While most experimenters treat the population of MTurk workers (Turkers) as a homo-

geneous population, the actual population is very diverse and heterogeneous (Chandler et al. 2013). Even considering only the US population, there is a small core group of *super-Turkers* that are very active, depend on MTurk to make a living, and are very likely to be available over a long period of time. Other users on MTurk are more casual and use MTurk as a source of side income or out of curiosity, vary demographically by the time of day, and are much less likely to return if contacted weeks after a task.

The most active workers on MTurk communicate constantly outside of the system itself to share profitable HITs (tasks) and socialize on various online forums³. As a result, lucrative tasks posted on MTurk will be quickly shared and taken by the most active users. Moreover, users will often talk about the details of the task itself, especially when it is not apparent that it is a research study. Hence, all experimental designs on MTurk should include an explicit and obvious note not to discuss the task externally, due to the risk of cross-treatment contamination—especially for field experiment designs in crowdsourcing such as discussed in Chapter 3.

As MTurk does not have an explicit reputation system, the third-party TurkOpticon⁴ has emerged as an integrated website and browser plugin for workers to rate and review requesters. A requester’s TurkOpticon rating is very important for recruiting and retaining good workers, as poorly rated requesters will only see traffic from new or uninformed MTurk workers.

It is beneficial to explicitly engage in communication with Turkers in online forums for many other reasons. Because forums are the most active means of Turker communication, they are also the most direct line to participants for diagnosing software or comprehension problems. Because many requesters do not engage with their participants online, those that do participate in forums are more respected, creating a reciprocity effect where participants are more willing to help diagnose issues with the task, read the instructions more carefully, and avoid spamming. Finally, this communication results in better requester reputation, both

³The most popular sites change over time, but as of this writing, they include www.mturkforum.com, www.mturkgrind.com, <http://www.cloudmebaby.com/>, www.reddit.com/r/mturk, and www.turkernation.com/.

⁴turkopticon.ucsd.edu

by the ability to fix and improve problems with a HIT as well as giving Turkers a positive communication experience.

Finally, the heterogeneous nature of Turkers means that it is beneficial to log data about participation and check for understanding as much as possible. This includes asking comprehension questions, monitoring active participation and inactivity, and qualitative feedback both within and outside of a HIT itself. Learning from this cycle yields a better designed task and ultimately better experimental data.

6.2 Designing and Conducting Experiments using TurkServer

In this section, I review a series of contributions to deploying and conducting software-controlled experiments allowing for real-time interaction, qualitative participant observation, and other procedural improvements. This includes the *TurkServer*⁵ open source software package, as well as several other software packages, and a discussion of methods for facilitating experiments with many users and/or real-time interaction.

6.2.1 Web Technology and Software

The original design of the web, was designed for purely *static* pages: a client (browser) would issue a request and the server would return a fixed response, usually the contents of a file. As websites changed to allow *dynamic* content, scripts or programs on the server would now run to generate the contents of a page on the fly, depending on the state of the user and their actions. These scripts could be run in many different programming languages, from the earliest such pages in Perl with CGI to more recent frameworks such as Rails (Ruby) and Django (Python).

However, because loading a new webpage is a resource-intensive operation that usually halts user interaction for a few seconds or more, much of the generation of dynamic content now occurs in the browser, using code written for the ECMAScript standard (Javascript). As a result, building real-time web applications often requires programming in two languages—

⁵<https://github.com/HarvardEconCS/turkserver-meteor>

Javascript on the client and a second language on the server—along with logic to deal with asynchronous client-server communication between the two.

This situation makes it very difficult to apply good programming practice to web development, but is still the model underlying many web frameworks (Rails, ASP.NET, Django) despite its difficulty of use. However, the development of server-side Javascript in the form of Node.js and a new way of thinking about primitives for real-time web applications has resulted in some exciting and revolutionary ideas, exemplified by web frameworks such as Meteor.⁶

Meteor innovates in several ways that are a marked improvement for developing real-time web applications, such as experimental studies:

- **One language.** Instead of writing in two different languages on the server and client, Meteor allows for the use of a single language—Javascript—for both environments. This allows for ease of code organization and reuse while reducing the overhead needed to maintain a web application.
- **Database caching and live queries.** Instead of requiring round-trip AJAX requests to the server to fetch data, Meteor maintains a local cache of the server’s database in client memory, allowing many operations to be done instantly without waiting for the server to respond. This local database is updated using *live queries*—queries whose results are updated using a diffing protocol⁷ instantly as the database changes. As a result, client-side data is continually kept up to date with the server without requiring additional logic from the programmer beyond the query specification.
- **Reactive user interfaces.** Meteor constructs user interfaces using a client-side library that ties UI elements to the local database cache, so that if the data changes, the UI elements automatically update as well. As a result, building a user interface in Meteor is a *declarative* process of describing *what* will be rendered as opposed to an *imperative* process of *how* to change elements on the page. This model results in significantly less

⁶<https://www.meteor.com>

⁷Distributed Data Protocol (DDP): <https://www.meteor.com/ddp>

error-prone programming and is also used in similar frameworks such as Facebook’s React.⁸

Taken together, Meteor makes prototyping and designing real-time web applications significantly easier than traditional techniques, and it is a perfect fit for the rapid iteration and design needed to deploy web-based experiments. While Meteor is arguably facing some non-traditional challenges in areas such as search engine optimization, these are typically irrelevant for experimental studies and will likely be addressed over time.

6.2.2 Software Abstractions

TurkServer and other software discussed here build on top of Meteor with the aim of providing a comprehensive set of modules to design, deploy, monitor, and collect data from web-based experimental studies. From a software perspective, TurkServer innovates in the following ways:

“Multiple worlds” abstraction. The main software abstraction in TurkServer is the observation that the smallest unit of an experimental session is a grouping of participants (possibly even a single one) that have the ability to interact with one another. This type of paradigm was perhaps first used on a large scale by Salganik et al. (2006), and also central to the crisis mapping experiment described in Chapter 4. The `partitioner`⁹ package for Meteor implements this abstraction, allowing experiment software to reason about the interaction between participants in a particular group without having to consider other users connected to the server. This also improves the simplicity of setting up tutorials and practice sessions with full data logging. TurkServer builds upon the multiple worlds abstraction to allow re-assignment of participants to different new worlds.

Integration with MTurk API. The MTurk API lacks features such as being able to prevent repeat participation in a straightforward way, and only recently gained other features

⁸<https://facebook.github.io/react/>

⁹<https://atmospherejs.com/mizzao/partitioner>

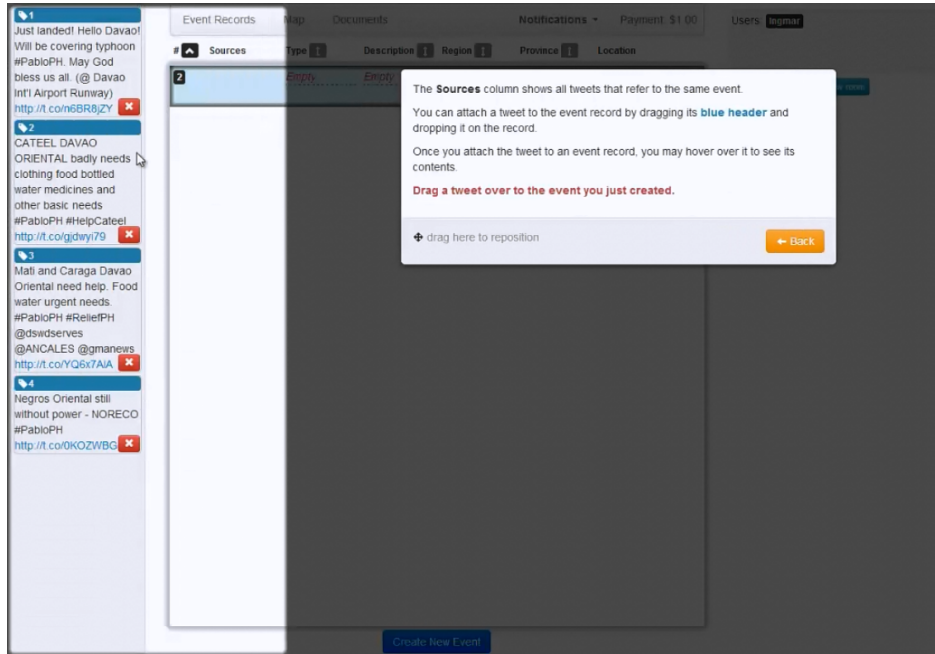


Figure 6.1: One step of the interactive tutorial for participants in the experiment described in Chapter 4. A shadow focuses the interface on the desired area for the user, along with step-by-step instructions. The user must complete the action in the step to proceed, ensuring that they learn how to use the interface.

such as a HIT that accepts participants from more than one country. TurkServer aims to abstract the intricacies of the MTurk API from the experimenter, implementing features such as tracking and limiting participation of a unique participant over time, with the goal of being able to integrate other subject recruitment pools in the future.

6.2.3 Interactive Tutorials

Alongside the main TurkServer package is a tutorials package for Meteor.¹⁰ The package facilitates the design of interactive step-by-step tutorials that clearly and concisely explain an interface to participants (Figure 6.1 shows a step of the tutorial for the experiment described in Chapter 4). Each step of the tutorial supports the following:

- A custom HTML template to display instructions for the current step.
- A *highlight* area to be shown on the page, with all other areas dimmed by a shadow.

¹⁰<https://atmospherejs.com/mizzao/tutorials>

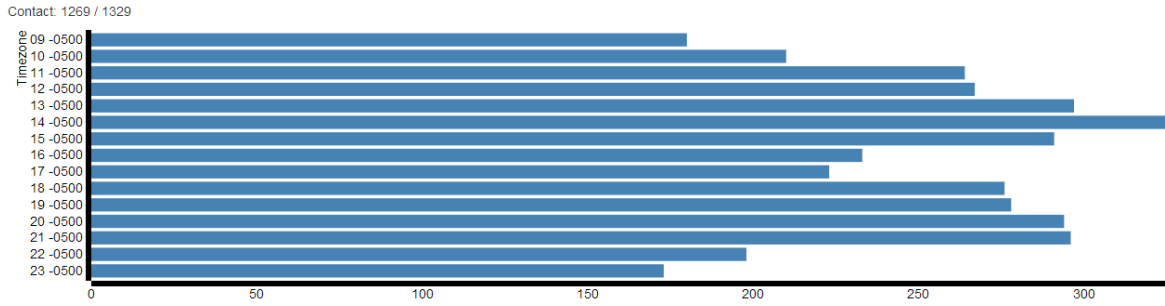


Figure 6.2: Initial availability of about 1,300 recruited users, available at 2 PM.

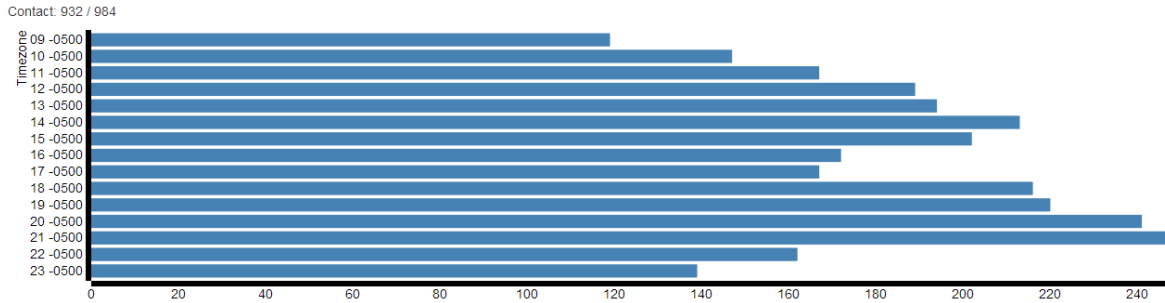


Figure 6.3: Remaining users available after several experiment sessions at 2 PM Eastern Time.

The highlight is calculated as the convex hull of one or more specified DOM selectors, such that the tutorial will automatically update if the interface updates without the need for pre-specified positioning.

- An optional *mandatory action* that the user must complete to continue. This helps to ensure that users learn how to interact with the interface, and limits spamming actions without reading the instructions.

Put together, the tutorials package allows for interactive tutorials that are programmatically specified and automatically update with user interfaces, with little boilerplate code required. This significantly eases the delivery of instructions while also being flexible enough to accommodate changes in interface design.

6.2.4 Improved Recruitment and Scheduling

Mason and Suri (2012a) described a mechanism for building a virtual “panel” of users for scheduling simultaneous experiment sessions with interaction. Due to the heterogeneity of Turkers as described in Section 6.1.5, not all workers are equivalent, nor will they even be available over the same period of time. Scheduling users for online experimental sessions requires relatively accurate forecasting of users’ availability.

It is difficult to enforce a specific number of participants because there is no cost to users to not showing up. Hence, the general approach is to contact slightly more users than necessary to fill all treatment conditions. This is a delicate balance: if too many users are invited, some may be turned away or directed to discarded data; while if too few users are invited, not all treatment conditions may fill properly—especially a problem for group interaction studies. TurkServer improves the scheduling of experimental sessions by collecting availability data by asking for several preferred times for each user and normalizing the aggregate data by time zone.

Figure 6.2 shows the initial availability of about 1,300 users recruited from the US population. The highest number of users available is at 2 PM Eastern time. We scheduled several experiment sessions at this time, and invited a random subset of users from the panel.

Figure 6.3 shows the remaining availability after our experimental sessions in Chapter 4, with less than 1,000 new users on the panel. Note that the 2 PM mode is significantly diminished compared to the new mode at 9 PM Eastern Time. However, even users who had specified that they were available at other times still participated in our experiment sessions scheduled at 2 PM.

Future improvements can include better targeting of panel users who are available at a particular time, weighting of availability by panel age (as older users are less likely to return) and estimates of arrival probability by time bucket. These features will be much easier to implement with sufficient data about online participant pools.

Start Time	Duration	Treatments	Size	Users	
8/13/2014 2:21:50 PM	0:05:50	parallel_worlds	8	nlv53tan DawnBaker tengford kira432 Tenkei nudicorn DCKst1 techboyof	Watch Logs Stop
8/13/2014 2:20:33 PM	0:07:07	group_1 parallel_worlds	1	eddie	Watch Logs Stop
8/13/2014 2:17:35 PM	0:10:05	group_1 parallel_worlds	1	gms5002	Watch Logs Stop
8/13/2014 2:16:58 PM	0:10:41	group_2 parallel_worlds	2	Jennifer Spylet7	Watch Logs Stop
8/13/2014 2:12:05 PM	0:15:35	group_4 parallel_worlds	4	bbs444 hicks7 mcwilliams mkejamo	Watch Logs Stop
8/13/2014 2:11:35 PM	0:16:04	group_2 parallel_worlds	2	plrs199 bponerflec	Watch Logs Stop
8/13/2014 2:11:35 PM	0:16:05	group_1 parallel_worlds	1	atsera	Watch Logs Stop
8/13/2014 2:11:35 PM	0:16:05	group_1 parallel_worlds	1	CatsMeow	Watch Logs Stop
8/13/2014 2:11:32 PM	0:16:07	group_16 parallel_worlds	15	shrimp Mphira Yalorio brynnaroo Adak17 marko42 CAD0Y5214344 letauoZahina Jayden gogogogogo Xylozcent robert rhobar92 Geronimo Jomy462	Watch Logs Stop
8/13/2014 2:11:32 PM	0:16:07	group_8 parallel_worlds	8	Andrewmatt Eric535 JohnHocker Presto Nathan Kelly19 keezay ryanawall	Watch Logs Stop
8/13/2014 2:11:32 PM	0:16:08	group_1 parallel_worlds	1	edfnewg46	Watch Logs Stop
8/13/2014 2:11:32 PM	0:16:08	group_1 parallel_worlds	1	Klasens	Watch Logs Stop
8/13/2014 2:11:32 PM	0:16:08	group_4 parallel_worlds	4	Dandiata Rheilont17 ccarran codebanker	Watch Logs Stop
8/13/2014 2:11:32 PM	0:16:08	group_2 parallel_worlds	2	mrts123 gavidae	Watch Logs Stop
8/13/2014 2:11:31 PM	0:16:08	group_2 parallel_worlds	2	ars141 lencid	Watch Logs Stop
8/13/2014 2:11:31 PM	0:16:08	group_32 parallel_worlds	32	andakrs McBoocods sugarcrumb rthbgn supermoon Juliet Amphiboles Kellertook plucky naaron Andy241 abanethian karde_hamster olivia hobbie nthen scobak612 Magnarider hotblangeit ssak916 Cefkoi mike2112 mikewhe octogonalfish Bruce43 kadath stephanee1065 Wolf359 Kenzie1 manderwitz meekocho83 stephanie	Watch Logs Stop
8/13/2014 2:11:31 PM	0:16:09	group_1 parallel_worlds	1	mmogegner	Watch Logs Stop

Figure 6.4: Simultaneous assignment of about 100 users to different-sized groups.

6.2.5 Qualitative Observation

Compared to physical lab studies where participants are easily observed by experimenters, it is difficult to observe the activity and participation of users connected to a web application. TurkServer aims to improve the process of this observation in a few key ways.

First, TurkServer allows for real-time viewing of connected users as well as the groups they are assigned to. An inactivity monitor logs when users are idle (not having performed an action for a while) or has switched to another window. This “virtual console”, shown in Figure 6.4, gives a quick overview of all participants and their state.

Furthermore, building experiment software on top of real-time web frameworks as described in Section 6.2.1 allows for live viewing of participants in a similar way to one-way windows in physical labs, but with potentially far more flexibility. The crisis mapping experiment described in Chapter 4, for example, is particularly suitable for this because all users see an identical interface showing the data in the experiment, and thus the experimenter can use the same view as well.

Figure 6.5 shows a screenshot of several randomized groups of different sizes participating



during an experiment session. This ability to view the activity of participants is not only helpful for understanding behavior qualitatively, but it additionally allows very fast diagnosis of user confusion or other problems that can aid in rapid iteration of experiment design.

Finally, the live data captured during real-time experiments can be integrated with web-based visualization software using libraries such as D3 (Bostock et al. 2011) to provide even more informative summaries of user behavior beyond what is possible through simple qualitative observation. For example, Figures 6.6 and 6.7 allow for a concise summary of the division of labor in a team and the chat network, respectively. We foresee that live data visualization of the progress during an experiment will be a powerful tool in the future to understand the processes underlying human behavior.

6.2.6 Monitoring and Data Logging

TurkServer uses computer systems principles to enable efficient and accurate data logging on client and server software. An implementation of NTP (network time protocol)-style time synchronization¹¹ allows server time to be used down to millisecond accuracy on clients, enabling features such as inactivity monitoring without continual round-trip messages to the server. Additionally, this synchronization allows for actions on any client to be accurately timestamped before sending to the server, such that logs can be constructed as a total ordering across the entire system regardless of any client to server latency.

One particularly important use of this type of highly accurate logging is exemplified by the instrumentation described in Chapter 4. Logging actions by all users with accurate timestamping allows us to reconstruct the state of data in the application at any point in time, allowing for metrics of individual performance not just in summary at the end of the task, but at any point between the start and the end. Moreover, this allows for a live, interactive “replay” of any experiment session¹², serving the same purpose as audio and video recording in a physical lab but with much less overhead. With the ability to qualitatively observe an experiment at any time, as many times as needed, experimenters now have significantly more

¹¹<https://github.com/mizzao/meteor-timesync>

¹²One such video, showing a group of 32 users, can be viewed at https://youtu.be/xJYq_kh6N1I

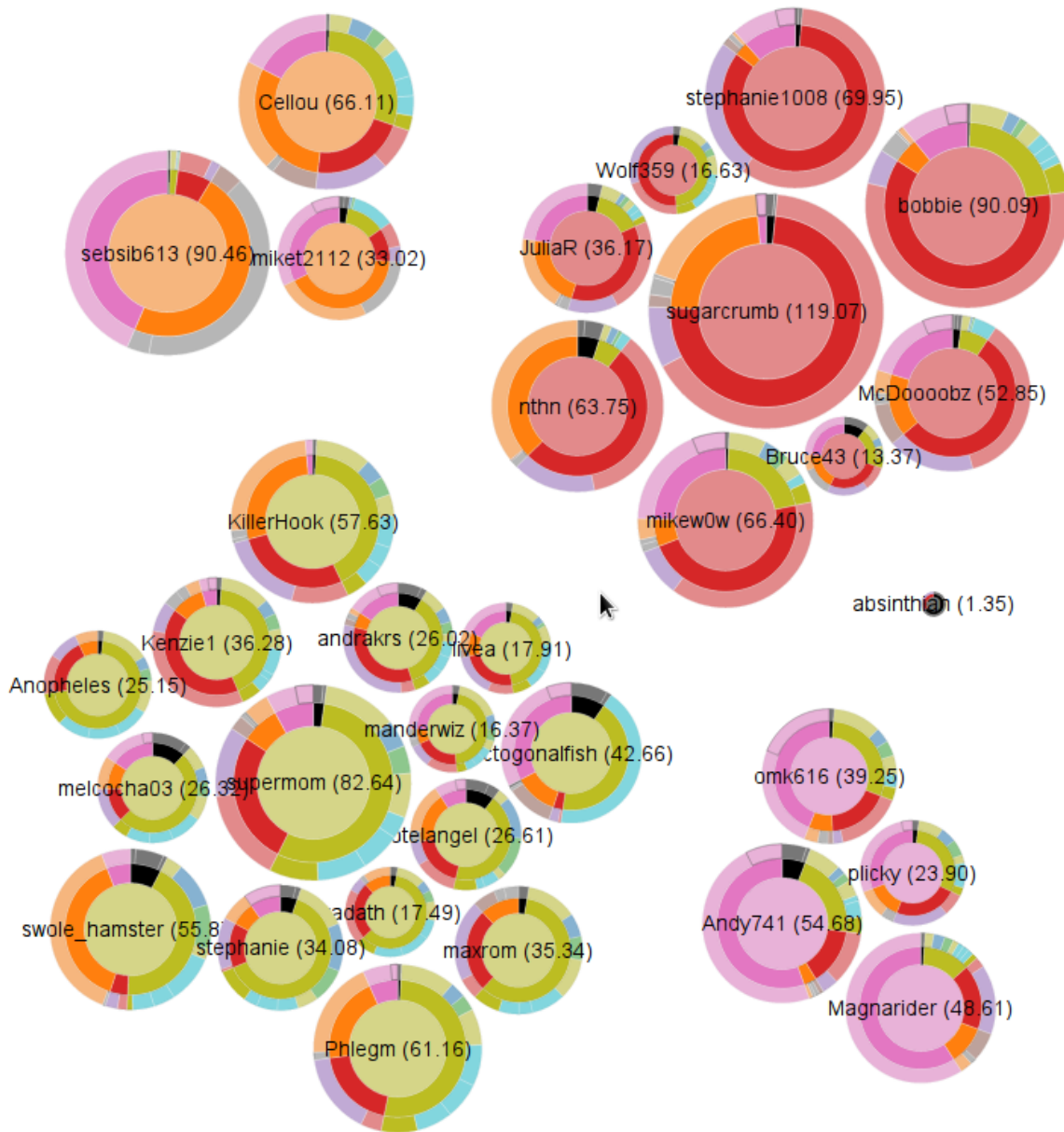


Figure 6.6: Clustering by main subtask for a group of 32 workers from the experiment in Chapter 4, showing different groups doing predominantly filtering, classification, verification, and communication.

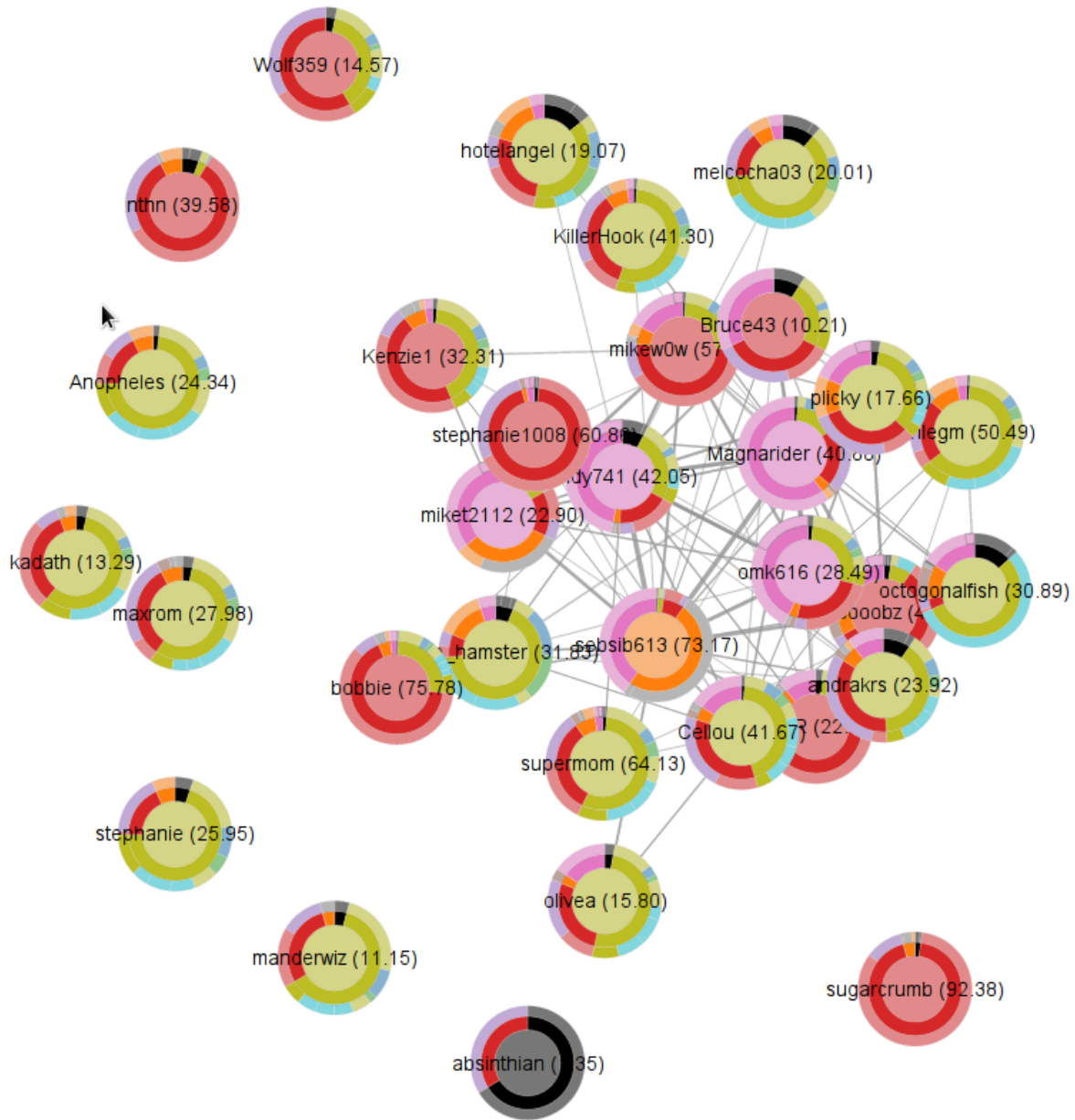


Figure 6.7: Empirical chat network for the group of workers shown in Figure 6.6. A few core workers who coordinated the group emerge at the center of the visualization, while others have little or no communication working independently.

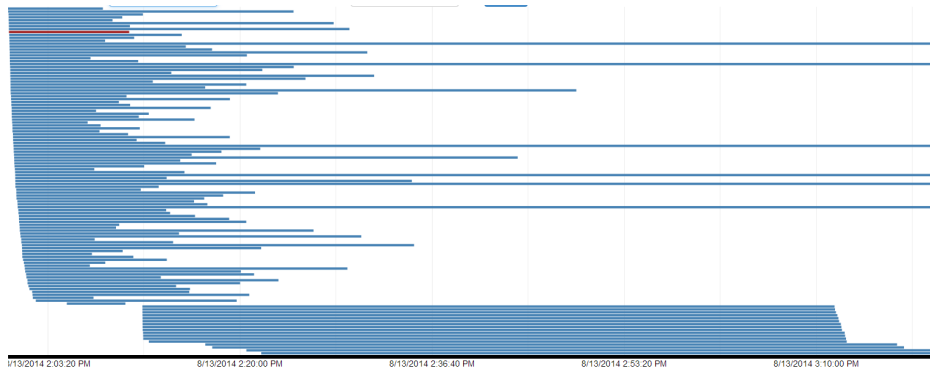


Figure 6.8: Assignment of users to tutorial instances, followed by randomization into groups of different sizes.

latitude to make qualitative observations.

6.2.7 Randomization Methodology

Taken together, the innovations described in this section foreshadow significant advances in methods for online experiments. Real-time software allows for qualitative observation, data visualization, and experiment replay capabilities that match and potentially eclipse the observation tools in any physical lab. Software control also allows for online experiments to scale far beyond what is possible with subjects that may fit in a physical room.

In accommodating the increased number of simultaneous users that may be recruited in web-based experiments, we developed a novel method for randomizing users into groups of different sizes while allowing for each user to complete a procedural tutorial that refreshed their familiarity with the task.

Figure 6.8 shows the mechanics of this process. In this case, users were e-mailed to arrive at 2 PM Eastern time. Each arriving user immediately begins a tutorial task that reviews the instructions and interface for the experiment. As users complete the tutorial, they wait in a lobby area with other users in order for a critical mass of users to complete the tutorial. When enough users are in the lobby, but without necessarily all users completing the tutorial, the randomization process starts. Each individual spot in a treatment condition is assigned a “ticket”, and these tickets are randomly shuffled and given out to users. The

assignment process continues as later users complete the tutorial, allowing most groups to be filled within a few minutes of the first and last arrival while minimizing boredom or distraction from the earliest users waiting too long. Finally, a “buffer” group is used to hold all of the slowest arrivals, which are commonly indicative of users having technical problems or misunderstanding; this data is discarded.

6.3 Imagining the Future of Virtual Experiment Labs

Arguably the most important feature of web-based experiments is the ability to recruit large numbers of users on short notice, without the scheduling constraints of physical labs. This allows for rapid iteration and testing of experimental hypotheses, and for participant feedback to inform the design of future experiments. Reducing the cost of deploying experiments will improve their viability as a research method in the future. However, there are other features that we imagine will define an effective virtual experiment lab in the future.

The current state of experimental software consists of separate packages in various forms of stability across a wide variety of disciplines. This is inefficient, because it is not only the result of much repeated work to achieve the same end goal, but because it will significantly inhibit the ability to replicate and modify existing experiments due to incompatible software and methodology. In the future, designing a centralized software system for deploying experiments, especially using the same subject pool, will allow the research community to focus their efforts on one set of standards while facilitating replication and repeatability.

Better tracking and consistency of the participant user pool is important for many experiment designs and currently difficult on systems such as Mechanical Turk. Chandler et al. (2013) showed that a non-negligible fraction of participants have become “professional participants” and know almost all of the expected answers to priming questions presented in psychology studies. Additionally, it is impossible to consistently bring back users who have participated in the past due to the high rate of turnover. Building a more consistent pool of participants with known experience in different types of research will lower the idiosyncratic noise of experimentation and also allow for novel approaches to studying people over longer

periods of time.

Chapter 4 suggests that there is much to learn about the social behavior and interaction of users in a real-time setting, which is much more natural than the prescribed communication in most experiments. However, real-time applications are still difficult to deploy and typically require computer systems experience to implement and debug. We foresee that wider adoption of next-generation web technologies such as those described in this chapter will facilitate more realistic experiments allowing for the study of interaction and cooperation between users in more natural settings.

Chapter 7

Conclusion

The ubiquity of social computing systems in all aspects of human lives and the resulting torrent of behavioral data simultaneously creates an opportunity and challenge for computing system designers to understand and reason about human behavior. Although we only scratch the surface of possibilities in this work, experimental studies have great potential to reach far beyond their traditional applications and become a core tool for computer scientists, both for designing desirable interventions for real systems and learning about individual and collective behavior at scale. As techniques for conducting online experiments are only in their infancy, we are excited to see novel research findings emerge in tandem with new innovations in methodology. In this chapter, we summarize the work presented in this dissertation, some of the common challenges faced in experimental studies and exciting directions that lie ahead.

7.1 Summary of Contributions

Chapter 3 ties together performance on a similar task from a volunteer to a paid setting, and suggests that there is much additional work to be done in understanding how to combine both economic and non-economic incentives with the goal of designing more efficient social computing systems. First, as motivated by Section 3.2, we must create better models of the attention and engagement of users, understanding why they spend time and effort in contribution and the reasons for which they return. Second, Section 3.3 demonstrates that

financial incentives can have subtle, nuanced effects beyond rational agent models, and that we must take these into account when designing payments. An important challenge for designing social computing systems in the future is to effectively use of a combination of social and monetary incentives to engage users and encourage their participation, and this is a ripe area for further study.

Chapter 4 presents an in-depth study of how groups of different sizes perform on an identical task. By designing an application that simulates the environment of a real crisis mapping deployment, we captured interaction between individuals working on a real-world task of practical importance while also deploying fine-grained instrumentation of user activity that allowed for a better understanding of how teams worked together. Our study suggests that although there are many effects that decrease raw productivity as people work together, including social loafing, reduced individual effort, and coordination costs; there is significant value in the collective intelligence of teams and that this could be further improved with better organization. We also expect that this study will set a future standard for more realistic experiments with better external validity.

Chapter 5 highlights the importance of testing the theoretical properties of information aggregation mechanisms in realistic settings. While plurality voting performs worse in many models, our experiments demonstrated that it can be an effective method in many voting settings while being significantly simpler to implement. Moreover, our work also shows that better probabilistic ranking models can not only be used for more accurate information aggregation, but also to succinctly describe the varied preferences across a population at large. Designing more accurate probabilistic ranking models will allow for the implementation of more principled systems for eliciting ranking information in the future.

Taken together, the work described in this dissertation points to a future where the human aspect of computational systems is better understood and used to make principled design decisions rather than tested through trial-and-error. Expanding the reach of this vision requires more fundamental research through more extensive deployment of experimental studies in social computing, as discussed in the next section.

7.2 Challenges for Experimental Design

Despite the synergy between behavioral experiments and social computing systems, **experimental studies are still relatively uncommon in computer science**. Gerber and Green (2012) argue that “developing expertise as an experimental researcher is part technical training and part apprenticeship”, and as there are few such apprenticeships in much of computer science, the methodology of experimental design is not well established, and many first-time experimenters experience the same pitfalls. As an example, consider the side effect of the proliferation of similar research into a single pool of participants: Chandler et al. (2013) showed that workers are much less naïve than many researchers imagine, and that “professional participants” already knew the answers to all of the priming questions in common psychology studies.

The academic community also faces challenges in both **interdisciplinary research** and **collaboration with industry and other organizations**. While there are many disciplines effectively studying human behavior, there is a distinct tendency to hold to one’s tribal affiliations and discount the approaches and methods used by other fields (Watts 2013). In large-scale field experiments, academic researchers are mainly interested in general scientific questions, companies are primarily concerned with user experience and profit motivations, with competing goals for the design of experimental studies. Moreover, inappropriate presentation of experimental work in the popular media, such as Facebook’s “emotional contagion” study (Kramer et al. 2014), risks a scenario where industry research on real systems become unpublished for fear of retribution in the court of public opinion. On the other hand, collaboration between academics and companies can be a fruitful source of general knowledge, as on Airbnb (Fradkin et al. 2014) and OKCupid (Rudder 2014).

Web-based experiments require **new techniques to collect and analyze data**, using a combination of methods employed in traditional lab and field experiments along with new innovations (Reips 2002). While experiments on systems such as MTurk can be designed with highly standardized procedures and a great deal of control (as in the lab), participants are not constrained to stay if they lose interest and can show attrition (as in the field). As an

online experiment often competes with other demands of the participant for attention (Mao et al. 2013a), experimenters face a new challenge in designing engaging studies without the luxury of a captive audience in the lab. Instructions that may have been previously delivered in a lengthy document or verbal presentation must now be presented concisely and intuitively in software, and participants should be monitored to detect temporary lapses in attention or dropping out altogether.

While online experiments may require less manpower to deploy than physical lab or field studies, the proliferation of experimental software frameworks shows that there is still a **significant amount of effort is often required to conduct and manage an experiment**—effort that is often repeated across many projects. Many common experiment designs involves elements of a distributed software system for human agents, used in applications such as Legion (Lasecki et al. 2011) and in experimental frameworks such as TurkServer (Mao et al. 2012), and can benefit from a systematic and unified approach to designing user interfaces and collecting data. An important aspect of sound experimental design is the ability to replicate a study and reproduce its results—e.g. Klein et al. (2014). By sharing both code and data for software-based experiments and using standardized participant pools and software frameworks, we can potentially reach a point where replication is as easy as cloning and running an open-source project.¹

A primary challenge for designing large-scale experiments is the ability to **reach many participants simultaneously** while also controlling for repeat participation and reaching a sufficiently large sample size. As Amazon Mechanical Turk has an active pool of a few thousand users at a time, it is difficult to achieve all of these objectives simultaneously. While traditional behavioral labs were limited to local participant pools, an appropriately designed online experiment system for the research community can effectively create an even bigger “virtual participant pool” that is much bigger than what is available either on MTurk or in physical labs.

¹The Open Science Framework (<https://osf.io/>) has started to tackle the problem of replication and sharing.

7.3 Future Directions in Online Experiments

Amazon’s Mechanical Turk—adopted simultaneously as a substitute for physical behavioral labs in fields such as psychology (Paolacci et al. 2010, Buhrmester et al. 2011), political science (Berinsky et al. 2012), and linguistics (Sprouse 2011)—is currently the system that most associated with online behavioral experimentation. Recruiting online subjects is certainly more economical in scale, allowing for higher throughput and access to more participants as well as a greater diversity of participants than typical university labs (Reips 2000). An on-demand participant pool allows for faster iteration in the cycle of designing experiments, collecting data, and forming new hypotheses compared to constraints of scheduling physical studies.

However, online experiments have far more potential than straightforward replication of traditional lab environments on the Internet. Particularly valuable is the flexibility of design: past experiment designers have faced the choice between a lab experiment with more procedural control or a field experiment with better external validity (List 2008). While physical lab experiments face difficulty in generalizing to the real-world environments they simulate (Winer 1999), web-based experiments can bridge the gap between traditional lab and field environments, running on live systems or other realistic online environments while still allowing for unobtrusive but extensive instrumentation, procedural control, and data collection through software.

In the future, it will be natural to design future studies on the web to study online behavior using the Internet itself, in an environment that captures social online interaction much more accurately than any behavioral lab can reflect the virtual world—perhaps even allowing for new types of studies that were impossible before. For example, experiments on organization and communication in groups have been of limited value when conducted in physical laboratories because of a lack of resemblance to real organizations. Distant relatives of today’s web-based experiments were attempted in several decades ago in social science laboratories, but were eventually abandoned due to lack of progress—see Shure and Meeker (1970) for one of many examples. Zelditch Jr (1969) summarizes the era by asking the

rhetorical question of whether an “army” could be studied experimentally, concluding that it was both unnecessary and infeasible. Yet, web applications routinely connect dozens or hundreds of users together in many different types of tasks, making it possible to study collective behavior experimentally and outside the artificial environment of a lab.

Many current assumptions in recruiting paid participants from systems such as MTurk can be bypassed in novel ways. Mason and Suri (2012a) demonstrated the novel approach of recruiting many subjects simultaneously and studying their interaction, demonstrating the concept of large interactive studies of many more people than could possibly fit in a physical lab. Another implicit constraint carried over from laboratory studies is that experiments take place over a short, contiguous period of time. However, by tracking participants’ identities and storing some persistent information, we can facilitate asynchronous interactions between large numbers of people—hundreds, or even thousands—over longer periods of time, such as weeks or months. This allows for potentially large-scale studies of interaction between networked users and observing learning over longer periods of time, which is impossible with short, ephemeral studies.

There are several other ways to engage significant numbers of experimental participants on the web. The Zooniverse (Raddick et al. 2013, Reed et al. 2013), a citizen science platform, uses millions of volunteers to crowdsource various scientific tasks, and is simultaneously building infrastructure for enable real-time, large scale behavioral studies. Another approach are self-hosted online volunteer laboratories that recruit large numbers of voluntary participants by simply providing something of value or interest. This approach has been used in studies of cognitive function (Halberda et al. 2012) and visual aesthetics (Reinecke and Gajos 2014) to produce studies with many thousands or even millions of participants and unprecedented amounts of data.

A final and particularly exciting direction is to leverage the prevalence of smartphones and apps to access participants for experimental studies (Miller 2012). While only a few thousand users are available at any particular time on Amazon’s Mechanical Turk, there will be an estimated 200 million smartphone users in 2016 in the United States alone—many of

whom have their devices at arm's length. Tapping into the huge potential of smartphone users also allows us to bridge the gap from experiments in the digital world back to the real world, using GPS locations and other proximate interactions to allow large numbers of people to interact at scale.

In summary, our emerging ability to study people in the digital and physical worlds at large scale and in natural contexts will revolutionize our understanding of human behavior—not only for the design of social computing systems, but for building society as a whole.

Bibliography

- Eytan Adar, Jaime Teevan, and Susan T. Dumais. Large scale analysis of web revisitation patterns. In *Proceedings of the 26th ACM Conference on Human Factors in Computing Systems (CHI)*, 2008.
- Ammar Ammar and Devavrat Shah. Ranking: Compare, dont score. In *Proceedings of the 49th Annual Allerton Conference on Communication, Control, and Computing*, 2011.
- A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec. Steering user behavior with badges. In *Proceedings of the 22nd International World Wide Web Conference (WWW)*, 2013.
- Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. Engaging with massive online courses. In *Proceedings of the 23rd International World Wide Web Conference (WWW)*, pages 687–698, 2014.
- Kenneth J Arrow, Robert Forsythe, Michael Gorham, Robert Hahn, Robin Hanson, John O Ledyard, Saul Levmore, Robert Litan, Paul Milgrom, Forrest D Nelson, et al. The promise of prediction markets. *Science*, 320(5878):877, 2008.
- Hossein Azari Soufiani, David C. Parkes, and Lirong Xia. Random utility theory for social choice: Theory and algorithms. In *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS)*, 2012.
- Eytan Bakshy, Dean Eckles, and Michael S Bernstein. Designing and deploying online field experiments. In *Proceedings of the 23rd International World Wide Web Conference (WWW)*, pages 283–292. International World Wide Web Conferences Steering Committee, 2014.
- Eytan Bakshy, Solomon Messing, and Lada Adamic. Exposure to ideologically diverse news and opinion on facebook. *Science*, page aaa1160, 2015.
- Gary S Becker and Kevin M Murphy. The division of labor, coordination costs, and knowledge. *The Quarterly Journal of Economics*, 107(4):1137–1160, 1992.
- Yochai Benkler. Law, policy, and cooperation. In *Government and markets: Toward a new theory of regulation*, pages 299–332. Cambridge University Press, 2009.

- James Bennett and Stan Lanning. The netflix prize. 2007.
- J Bercovici. Justine sacco and the self-inflicted perils of twitter. *forbes*. retrieved april 7, 2015, 2013.
- Adam J Berinsky, Gregory A Huber, and Gabriel S Lenz. Evaluating online labor markets for experimental research: Amazon. com’s mechanical Turk. *Political Analysis*, 20(3):351–368, 2012.
- Michael S Bernstein, Greg Little, Robert C Miller, Björn Hartmann, Mark S Ackerman, David R Karger, David Crowell, and Katrina Panovich. SoyLent: a word processor with a crowd inside. In *Proceedings of the 23rd Symposium on User Interface Software and Technology (UIST)*, pages 313–322. ACM, 2010.
- Michael S Bernstein, Mark S Ackerman, Ed H Chi, and Robert C Miller. The trouble with social computing systems research. In *Extended Abstracts in the 29th ACM Conference on Human Factors in Computing Systems (CHI)*, pages 389–398. ACM, 2011a.
- Michael S Bernstein, Joel Brandt, Robert C Miller, and David R Karger. Crowds in two seconds: Enabling realtime crowd-powered interfaces. In *Proceedings of the 24th Symposium on User Interface Software and Technology (UIST)*, pages 33–42. ACM, 2011b.
- Michael Scott Bernstein. *Crowd-powered systems*. PhD thesis, Massachusetts Institute of Technology, 2012.
- Luis Bettencourt. The rules of information aggregation and emergence of collective intelligent behavior. *Topics in Cognitive Science*, 1(4):598–620, 2009.
- Jeffrey P Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samuel White, et al. Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the 23rd Symposium on User Interface Software and Technology (UIST)*, pages 333–342. ACM, 2010.
- Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, 2011.
- Robert M Bond, Christopher J Fariss, Jason J Jones, Adam DI Kramer, Cameron Marlow, Jaime E Settle, and James H Fowler. A 61-million-person experiment in social influence and political mobilization. *Nature*, 489(7415):295–298, 2012.
- W. J. Borucki, D. Koch, G. Basri, N. Batalha, T. Brown, D. Caldwell, J. Caldwell, J. Christensen-Dalsgaard, W. D. Cochran, E. DeVore, E. W. Dunham, A. K. Dupree, T. N. Gautier, J. C. Geary, R. Gilliland, A. Gould, S. B. Howell, J. M. Jenkins, Y. Kondo, D. W. Latham, G. W. Marcy, S. Meibom, H. Kjeldsen, J. J. Lissauer, D. G. Monet, D. Morrison, D. Sasselov, J. Tarter, A. Boss, D. Brownlee, T. Owen, D. Buzasi, D. Char-

- bonneau, L. Doyle, J. Fortney, E. B. Ford, M. J. Holman, S. Seager, J. H. Steffen, W. F. Welsh, J. Rowe, H. Anderson, L. Buchhave, D. Ciardi, L. Walkowicz, W. Sherry, E. Horch, H. Isaacson, M. E. Everett, D. Fischer, G. Torres, J. A. Johnson, M. Endl, P. MacQueen, S. T. Bryson, J. Dotson, M. Haas, J. Kolodziejczak, J. Van Cleve, H. Chandrasekaran, J. D. Twicken, E. V. Quintana, B. D. Clarke, C. Allen, J. Li, H. Wu, P. Tenenbaum, E. Verner, F. Bruhweiler, J. Barnes, and A. Prsa. Kepler Planet-Detection Mission: Introduction and First Results. *Science*, 327:977–, February 2010. doi: 10.1126/science.1185402.
- Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D³ data-driven documents. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2301–2309, 2011.
- danah boyd and Kate Crawford. Critical questions for big data: Provocations for a cultural, technological, and scholarly phenomenon. *Information, Communication & Society*, 15(5): 662–679, 2012.
- Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.
- F. Brandt, F. Fischer, P. Harrenstein, and M. Mair. A computational analysis of the Tournament Equilibrium Set. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI)*, pages 38–43, 2008.
- Frederick P Brooks. *The mythical man-month*. Addison-Wesley Reading, MA, 1975.
- Michael Buhrmester, Tracy Kwang, and Samuel D Gosling. Amazon’s mechanical turk a new source of inexpensive, yet high-quality, data? *Perspectives on Psychological Science*, 6(1): 3–5, 2011.
- Colin F Camerer and Robin M Hogarth. The effects of financial incentives in experiments: A review and capital-labor-production framework. *Journal of Risk and Uncertainty*, 19(1-3): 7–42, December 1999. URL <http://ideas.repec.org/a/kap/jrisku/v19y1999i1-3p7-42.html>.
- John R Carlson, Joey F George, Judee K Burgoon, Mark Adkins, and Cindy H White. Deception in computer-mediated communication. *Group decision and negotiation*, 13(1): 5–28, 2004.
- Jesse Chandler, Pam Mueller, and Gabriele Paolacci. Nonnaïveté among amazon mechanical turk workers: Consequences and solutions for behavioral researchers. *Behavior Research Methods*, 2013.
- Xi Chen, Paul N. Bennett, Kevyn Collins-Thompson, and Eric Horvitz. Pairwise ranking aggregation in a crowdsourced setting. In *Proceedings of the 6th ACM Conference on Web Search and Data Mining (WSDM)*, WSDM ’13, pages 193–202, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1869-3. doi: 10.1145/2433396.2433420. URL <http://doi.acm>.

org/10.1145/2433396.2433420.

- V. Conitzer. Computing Slater rankings using similarities among candidates. In *Proceedings of the 21st AAAI Conference on Artificial Intelligence (AAAI)*, pages 613–619, 2006.
- V. Conitzer and T. Sandholm. Common voting rules as maximum likelihood estimators. In *Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 145–152, 2005.
- V. Conitzer, A. Davenport, and H. Kalagnanam. Improved bounds for computing Kemeny rankings. In *Proceedings of the 21st AAAI Conference on Artificial Intelligence (AAAI)*, pages 620–626, 2006.
- V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3):1–33, 2007.
- V. Conitzer, M. Rognlie, and L. Xia. Preference functions that score rankings and maximum likelihood estimation. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 109–115, 2009.
- S. Cooper, F. Khatib, A. Treuille, J. Barbero, J. Lee, M. Beenen, A. Leaver-Fay, D. Baker, and Z. Popović. Predicting protein structures with a multiplayer online game. *Nature*, 466: 756–760, 2010a.
- Seth Cooper, Adrien Treuille, Janos Barbero, Andrew Leaver-Fay, Kathleen Tuite, Firas Khatib, Alex Cho Snyder, Michael Beenen, David Salesin, David Baker, and Zoran Popović. The challenge of designing scientific discovery games. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games, FDG '10*, pages 40–47, New York, NY, USA, 2010b. ACM. ISBN 978-1-60558-937-4. doi: 10.1145/1822348.1822354. URL <http://doi.acm.org/10.1145/1822348.1822354>.
- Dan Cosley, Dan Frankowski, Loren Terveen, and John Riedl. Using intelligent task routing and contribution review to help communities build artifacts of lasting value. In *Proceedings of the 2006 ACM Conference on Human Factors in Computing Systems (CHI)*, CHI '06, pages 1037–1046, New York, NY, USA, 2006. ACM. ISBN 1-59593-372-7. doi: 10.1145/1124772.1124928. URL <http://doi.acm.org/10.1145/1124772.1124928>.
- Matthew JC Crump, John V McDonnell, and Todd M Gureckis. Evaluating amazon’s mechanical turk as a tool for experimental behavioral research. *PloS one*, 8(3):e57410, 2013.
- Laura Dabbish, Colleen Stuart, Jason Tsay, and Jim Herbsleb. Social coding in github: transparency and collaboration in an open software repository. In *Proceedings of the 15th ACM Conference on Computer Supported Cooperative Work (CSCW)*, pages 1277–1286. ACM, 2012.

- P. Dai, Mausam, and D. S. Weld. Decision-theoretic control of crowd-sourced workflows. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI)*, pages 1168–1174, 2010a.
- Peng Dai, Mausam, and Daniel S. Weld. Decision-theoretic control of crowd-sourced workflows. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI)*, 2010b.
- Peng Dai, Mausam, and Daniel S. Weld. Artificial intelligence for artificial artificial intelligence. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI)*, 2011.
- Paul DiMaggio, Eszter Hargittai, W Russell Neuman, and John P Robinson. Social implications of the internet. *Annual review of sociology*, pages 307–336, 2001.
- Peter Sheridan Dodds, Roby Muhamad, and Duncan J Watts. An experimental study of search in global social networks. *science*, 301(5634):827–829, 2003.
- C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th International World Wide Web Conference (WWW)*, pages 613–622, 2001.
- Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords. Technical report, National Bureau of Economic Research, 2005.
- David Engel, Anita Williams Woolley, Lisa X Jing, Christopher F Chabris, and Thomas W Malone. Reading the mind in the eyes or reading between the lines? theory of mind predicts collective intelligence equally well online and face-to-face. *PloS one*, 9(12):e115212, 2014.
- P. Faliszewski and A. D. Procaccia. AI’s war on manipulation: Are we winning? *AI Magazine*, 31(4):53–64, 2010.
- Ernst Fehr and Klaus M Schmidt. The economics of fairness, reciprocity and altruism—experimental evidence and new theories. In *Handbook of the economics of giving, altruism and reciprocity*, volume 1, pages 615–691. Elsevier, 2006.
- Urs Fischbacher. z-tree: Zurich toolbox for ready-made economic experiments. *Experimental economics*, 10(2):171–178, 2007.
- Debra A. Fischer, Megan E. Schwamb, Kevin Schawinski, Chris Lintott, John Brewer, Matt Giguere, Stuart Lynn, Michael Parrish, Thibault Sartori, Robert Simpson, Arfon Smith, Julien Spronck, Natalie Batalha, Jason Rowe, Jon Jenkins, Steve Bryson, Andrej Prsa, Peter Tenenbaum, Justin Crepp, Tim Morton, Andrew Howard, Michele Belevu, Zachary Kaplan, Nick vanNispen, Charlie Sharzer, Justin DeFouw, Agnieszka Hajduk,

- Joe P. Neal, Adam Nemec, Nadine Schuepbach, and Valerij Zimmermann. Planet Hunters: the first two planet candidates identified by the public using the kepler public archive data. *Monthly Notices of the Royal Astronomical Society*, 419(4):2900–2911, 2012. ISSN 1365-2966. doi: 10.1111/j.1365-2966.2011.19932.x. URL <http://dx.doi.org/10.1111/j.1365-2966.2011.19932.x>.
- R. Forsythe, T. Rietz, R. Myerson, and R. Weber. An experimental study of voting rules and polls in three-candidate elections. *International Journal of Game Theory*, 25(3):355–383, 1996.
- Andrey Fradkin, Elena Grewal, David Holtz, and Matthew Pearson. Reporting bias and reciprocity in online reviews: Evidence from field experiments on airbnb. 2014.
- Xi Alice Gao, Andrew Mao, Yiling Chen, and Ryan P. Adams. Trick or treat: Putting peer prediction to the test. In *Proceedings of the 15th ACM Conference on Electronic Commerce (EC)*, 2014.
- Alan S. Gerber and Donald P. Green. *Field Experiments*. W. W. Norton, 2012.
- Laura Germine, Ken Nakayama, Bradley C Duchaine, Christopher F Chabris, Garga Chatterjee, and Jeremy B Wilmer. Is the web as good as the lab? comparable performance from web and lab in cognitive/perceptual experiments. *Psychonomic bulletin & review*, 19(5):847–857, 2012.
- Gerd Gigerenzer and Reinhard Selten. *Bounded rationality: The adaptive toolbox*. Mit Press, 2002.
- Daniel G Goldstein, R Preston McAfee, and Siddharth Suri. The cost of annoying ads. In *Proceedings of the 22nd International World Wide Web Conference (WWW)*, pages 459–470, 2013.
- Richard Z Gooding and John A Wagner III. A meta-analytic review of the relationship between size and performance: The productivity and efficiency of organizations and their subunits. *Administrative science quarterly*, pages 462–481, 1985.
- Roger Guimera, Brian Uzzi, Jarrett Spiro, and Luis A Nunes Amaral. Team assembly mechanisms determine collaboration network structure and team performance. *Science*, 308(5722):697–702, 2005.
- Justin Halberda, Ryan Ly, Jeremy B Wilmer, Daniel Q Naiman, and Laura Germine. Number sense across the lifespan as revealed by a massive internet-based sample. *Proceedings of the National Academy of Sciences*, 109(28):11116–11120, 2012.
- Jeffrey T Hancock, Lauren E Curry, Saurabh Goorha, and Michael Woodworth. On lying and being lied to: A linguistic analysis of deception in computer-mediated communication.

Discourse Processes, 45(1):1–23, 2007.

Christopher Harris. You’re Hired! An Examination of Crowdsourcing Incentive Models in Human Resource Tasks. In *Proceedings of the Workshop on Crowdsourcing for Search and Data Mining (CSDM) at the Fourth ACM International Conference on Web Search and Data Mining (WSDM)*, pages 15–18, Hong Kong, China, February 2011.

E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Exact analysis of Dodgson elections: Lewis Carroll’s 1876 voting system is complete for parallel access to NP. *Journal of the ACM*, 44(6):806–825, 1997.

Joseph Henrich, Steven J. Heine, and Ara Norenzayan. The weirdest people in the world? *Behavioral and Brain Sciences*, 33:61–83, 6 2010. ISSN 1469-1825.

Chien-Ju Ho, Aleksandrs Slivkins, and Jennifer Wortman Vaughan. Adaptive contract design for crowdsourcing markets: Bandit algorithms for repeated principal-agent problems. In *Proceedings of the 15th ACM Conference on Electronic Commerce (EC)*, pages 359–376. ACM, 2014.

Chien-Ju Ho, Aleksandrs Slivkins, Siddharth Suri, and Jennifer Wortman Vaughan. Incentivizing high quality crowdwork. In *Proceedings of the 24th International World Wide Web Conference (WWW)*, 2015.

Nathan O Hodas, Farshad Kooti, and Kristina Lerman. Friendship paradox redux: Your friends are more interesting than you. In *Proceedings of the 7th AAAI Conference on Web and Social Media (ICWSM)*, 2013.

Bengt Holmstrom. Moral hazard in teams. *The Bell Journal of Economics*, pages 324–340, 1982.

John J. Horton. The dot-guessing game: A fruit fly for human computation research. Available at SSRN: <http://ssrn.com/abstract=1600372> or <http://dx.doi.org/10.2139/ssrn.1600372>, May 2010.

John J Horton, David G Rand, and Richard J Zeckhauser. The online laboratory: Conducting experiments in a real labor market. *Experimental Economics*, 14(3):399–425, 2011.

John Joseph Horton and Lydia B. Chilton. The labor economics of paid crowdsourcing. In *Proceedings of the 11th ACM Conference on Electronic Commerce (EC)*, pages 209–218, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-822-3. doi: 10.1145/1807342.1807376. URL <http://doi.acm.org/10.1145/1807342.1807376>.

Tanjim Hossain and John Morgan. ... plus shipping and handling: Revenue (non) equivalence in field experiments on ebay. *Advances in Economic Analysis & Policy*, 5(2), 2006.

- Bernardo A. Huberman, Daniel M. Romero, and Fang Wu. Crowdsourcing, attention and productivity. *J. Inf. Sci.*, 35(6):758–765, December 2009. ISSN 0165-5515. doi: 10.1177/0165551509346786. URL <http://dx.doi.org/10.1177/0165551509346786>.
- David R. Hunter. MM algorithms for generalized Bradley-Terry models. In *The Annals of Statistics*, volume 32, pages 384–406, 2004.
- Alan G Ingham, George Levinger, James Graves, and Vaughn Peckham. The ringelmann effect: Studies of group size and group performance. *Journal of Experimental Social Psychology*, 10(4):371–384, 1974.
- Panagiotis G. Ipeirotis. Analyzing the amazon mechanical turk marketplace. *XRDS*, 17(2):16–21, December 2010. ISSN 1528-4972. doi: 10.1145/1869086.1869094. URL <http://doi.acm.org/10.1145/1869086.1869094>.
- Irving L Janis. Victims of groupthink: a psychological study of foreign-policy decisions and fiascoes. 1972.
- Ece Kamar, Severin Hacker, and Eric Horvitz. Combining human and machine intelligence in large-scale crowdsourcing. In *Proceedings of the 11th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2012.
- Toshihiro Kamishima. Nantonac collaborative filtering: Recommendation based on order responses. In *Proceedings of the 9th International Conference on Knowledge Discovery and Data Mining (KDD)*, 2003.
- Steven J Karau and Kipling D Williams. Social loafing: A meta-analytic review and theoretical integration. *Journal of personality and social psychology*, 65(4):681, 1993.
- Nicolas Kaufmann, Thimo Schulze, and Daniel Veit. More than fun and money. worker motivation in crowdsourcing—a study on mechanical turk. In *Proceedings of the Seventeenth Americas Conference on Information Systems*, pages 1–11, 2011.
- Michael Kearns, Siddharth Suri, and Nick Montfort. An experimental study of the coloring problem on human subject networks. *Science*, 313(5788):824–827, 2006.
- Firas Khatib, Seth Cooper, Michael D Tyka, Kefan Xu, Ilya Makedon, Zoran Popović, David Baker, and Foldit Players. Algorithm discovery by protein folding game players. *Proceedings of the National Academy of Sciences*, 108(47):18949–18953, 2011.
- Aniket Kittur and Robert E. Kraut. Harnessing the wisdom of crowds in wikipedia: quality through coordination. In *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work (CSCW)*, CSCW ’08, pages 37–46, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-007-4. doi: 10.1145/1460563.1460572. URL <http://doi.acm.org/10.1145/1460563.1460572>.

- Aniket Kittur, Jeffrey V. Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. The future of crowd work. In *Proceedings of the 2013 ACM Conference on Computer Supported Cooperative Work (CSCW)*, pages 1301–1318, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1331-5. doi: 10.1145/2441776.2441923. URL <http://doi.acm.org/10.1145/2441776.2441923>.
- Richard A Klein, Kate A Ratliff, Michelangelo Vianello, Reginald B Adams Jr, Štěpán Bahník, Michael J Bernstein, Konrad Bocian, Mark J Brandt, Beach Brooks, Claudia Chloe Brumbaugh, et al. Investigating variation in replicability: A “many labs” replication project. *Social Psychology*, 45(3):142, 2014.
- Ron Kohavi, Roger Longbotham, Dan Sommerfield, and Randal M Henne. Controlled experiments on the web: survey and practical guide. *Data Mining and Knowledge Discovery*, 18(1):140–181, 2009.
- Adam DI Kramer, Jamie E Guillory, and Jeffrey T Hancock. Experimental evidence of massive-scale emotional contagion through social networks. *Proceedings of the National Academy of Sciences*, 111(24):8788–8790, 2014.
- Anand Kulkarni, Matthew Can, and Björn Hartmann. Collaboratively crowdsourcing workflows with turkomatic. In *Proceedings of the 15th ACM Conference on Computer Supported Cooperative Work (CSCW)*, 2012.
- Sébastien Lahaie, David M Pennock, Amin Saberi, and Rakesh V Vohra. Sponsored search auctions. In *Algorithmic game theory*, pages 699–716. 2007.
- Walter S Lasecki, Kyle I Murray, Samuel White, Robert C Miller, and Jeffrey P Bigham. Real-time crowd control of existing interfaces. In *Proceedings of the 24th Symposium on User Interface Software and Technology (UIST)*, pages 23–32. ACM, 2011.
- Walter S Lasecki, Samuel C White, Kyle I Murray, and Jeffrey P Bigham. Crowd memory: Learning in the collective. *arXiv preprint arXiv:1204.3678*, 2012.
- Walter S Lasecki, Rachel Wesley, Jeffrey Nichols, Anand Kulkarni, James F Allen, and Jeffrey P Bigham. Chorus: a crowd-powered conversational assistant. In *Proceedings of the 26th Symposium on User Interface Software and Technology (UIST)*, pages 151–162. ACM, 2013.
- Bibb Latane, Kipling Williams, and Stephen Harkins. Many hands make light the work: The causes and consequences of social loafing. *Journal of personality and social psychology*, 37(6):822, 1979.
- Patrick R Laughlin, Erin C Hatch, Jonathan S Silver, and Lee Boh. Groups perform better than the best individuals on letters-to-numbers problems: effects of group size. *Journal of Personality and social Psychology*, 90(4):644, 2006.

- Edward P. Lazear. Performance pay and productivity. *The American Economic Review*, 90 (5):pp. 1346–1361, 2000. ISSN 00028282. URL <http://www.jstor.org/stable/2677854>.
- David Lazer and Allan Friedman. The network structure of exploration and exploitation. *Administrative Science Quarterly*, 52(4):667–694, 2007.
- David Lazer, Alex Sandy Pentland, Lada Adamic, Sinan Aral, Albert Laszlo Barabasi, Devon Brewer, Nicholas Christakis, Noshir Contractor, James Fowler, Myron Gutmann, et al. Life in the network: the coming age of computational social science. *Science (New York, NY)*, 323(5915):721, 2009.
- John O Ledyard. Public goods: A survey of experimental research. 1994.
- Christopher H. Lin, Mausam, and Daniel S. Weld. Dynamically switching between synergistic workflows for crowdsourcing. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*, 2012.
- Harold A Linstone, Murray Turoff, et al. *The Delphi method: Techniques and applications*, volume 29. Addison-Wesley Reading, MA, 1975.
- C. J. Lintott, K. Schawinski, A. Slosar, K. Land, S. Bamford, D. Thomas, M. J. Raddick, R. C. Nichol, A. Szalay, D. Andreescu, P. Murray, and J. Vandenberg. Galaxy Zoo: morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey. *Monthly Notices of the Royal Astronomical Society*, 389:1179–1189, September 2008. doi: 10.1111/j.1365-2966.2008.13689.x.
- Chris J. Lintott, Megan E. Schwamb, Thomas Barclay, Charlie Sharzer, Debra A. Fischer, John Brewer, Matthew Giguere, Stuart Lynn, Michael Parrish, Natalie Batalha, Steve Bryson, Jon Jenkins, Darin Ragozzine, Jason F. Rowe, Kevin Schwainski, Robert Gagliano, Joe Gilardi, Kian J. Jek, Jari-Pekka Pkknen, and Tjapko Smits. Planet Hunters: New Kepler planet candidates from analysis of quarter 2. *The Astronomical Journal*, 145(6): 151, 2013. URL <http://stacks.iop.org/1538-3881/145/i=6/a=151>.
- John A List. Introduction to field experiments in economics with applications to the economics of charity. *Experimental Economics*, 11(3):203–212, 2008.
- G. Little, L. B. Chilton, M. Goldman, and R. C. Miller. TurkIt: Human computation algorithms on Mechanical Turk. In *Proceedings of the 23rd Symposium on User Interface Software and Technology (UIST)*, pages 57–66, 2010a.
- G. Little, L. B. Chilton, M. Goldman, and R. C. Miller. Exploring iterative and parallel human computation processes. In *Proceedings of the 2nd Human Computation Workshop (HCOMP)*, 2010b.
- Glenn E Littlepage. Effects of group size and task characteristics on group performance: A

- test of steiner’s model. *Personality and Social Psychology Bulletin*, 17(4):449–456, 1991.
- Tie-Yan Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.
- Jan Lorenz, Heiko Rauhut, Frank Schweitzer, and Dirk Helbing. How social influence can undermine the wisdom of crowd effect. *Proceedings of the National Academy of Sciences*, 108(22):9020–9025, 2011.
- T. Lu and C. Boutilier. Learning Mallows models with pairwise preferences. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 145–152, 2011.
- R. Duncan Luce. *Individual Choice Behavior: A Theoretical Analysis*. Wiley, 1959.
- Colin L. Mallows. Non-null ranking model. *Biometrika*, 44(1/2):114–130, 1957.
- Thomas W Malone and Kevin Crowston. The interdisciplinary study of coordination. *ACM Computing Surveys (CSUR)*, 26(1):87–119, 1994.
- Lena Mamykina, Bella Manoim, Manas Mittal, George Hripcsak, and Björn Hartmann. Design lessons from the fastest q&a site in the west. In *Proceedings of the 29th ACM Conference on Human Factors in Computing Systems (CHI)*, pages 2857–2866. ACM, 2011.
- Andrew Mao, Yiling Chen, Krzysztof Z. Gajos, David Parkes, Ariel D. Procaccia, and Haoqi Zhang. Turkserver: Enabling synchronous and longitudinal online experiments. In *Proceedings of the 4th Human Computation Workshop (HCOMP)*, 2012.
- Andrew Mao, Ece Kamar, Yiling Chen, Eric Horvitz, Megan E. Schwamb, Chris J. Lintott, and Arfon M. Smith. Volunteering vs. work for pay: Incentives and tradeoffs in crowdsourcing. In *Proceedings of the 1st AAAI Conference on Crowdsourcing and Human Computation (HCOMP)*, 2013a.
- Andrew Mao, Ece Kamar, and Eric Horvitz. Why stop now? predicting worker engagement in online crowdsourcing. In *Proceedings of the 1st AAAI Conference on Crowdsourcing and Human Computation (HCOMP)*, 2013b.
- Andrew Mao, Ariel D. Procaccia, and Yiling Chen. Better human computation through principled voting. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI)*, 2013c.
- W. Mason and S. Suri. Conducting behavioral research on amazon’s mechanical turk. *Behavior Research Methods*, 44(1):1–23, March 2012a.
- Winter Mason and Siddharth Suri. Conducting behavioral research on amazons mechanical turk. *Behavior research methods*, 44(1):1–23, 2012b.

- Winter Mason and Duncan J. Watts. Financial incentives and the "performance of crowds". In *Proceedings of the 1st Human Computation Workshop (HCOMP)*, pages 77–85, 2009.
- Winter Mason and Duncan J Watts. Collaborative learning in networks. *Proceedings of the National Academy of Sciences*, 109(3):764–769, 2012a.
- Winter Mason and Duncan J Watts. Collaborative learning in networks. *Proceedings of the National Academy of Sciences*, 109(3):764–769, 2012b.
- Rachel E McCaffrey. Using citizen science in urban bird studies. *Urban Habitats*, 3(1):70–86, 2005.
- Daniel McFadden. Conditional Logit Analysis of Qualitative Choice Behavior. In P. Zarembka, editor, *Frontiers in econometrics*, pages 105–142. Academic Press, New York, 1974.
- Patrick Meier. *Digital Humanitarians*. Taylor and Francis Press, 2015.
- Geoffrey Miller. The smartphone psychology manifesto. *Perspectives on Psychological Science*, 7(3):221–237, 2012.
- Nolan Miller, Paul Resnick, and Richard Zeckhauser. Eliciting informative feedback: The peer-prediction method. *Management Science*, 51(9):1359–1373, 2005.
- Frederick Mosteller. Remarks on the method of paired comparisons: I. the least squares solution assuming equal standard deviations and equal correlations. *Psychometrika*, 16(1):3–9, March 1951.
- Lev Muchnik, Sinan Aral, and Sean J Taylor. Social influence bias: A randomized experiment. *Science*, 341(6146):647–651, 2013.
- Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani. *Algorithmic game theory*. Cambridge University Press, 2007.
- Scott E Page. *The difference: How the power of diversity creates better groups, firms, schools, and societies*. Princeton University Press, 2008.
- Thomas R. Palfrey. Laboratory experiments in political economy. *Annual Review of Political Science*, 12:379–388, 2009.
- Gabriele Paolacci, Jesse Chandler, and Panagiotis G Ipeirotis. Running experiments on amazon mechanical turk. *Judgment and Decision making*, 5(5):411–419, 2010.
- Manoj Parameswaran and Andrew B Whinston. Research issues in social computing. *Journal of the Association for Information Systems*, 8(6):22, 2007.

- Judea Pearl. *Causality: models, reasoning and inference*, volume 29. Cambridge Univ Press, 2000.
- T. Pfeiffer, X. A. Gao, A. Mao, Y. Chen, and D. G. Rand. Adaptive polling and information aggregation. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*, 2012.
- Galen Pickard, Wei Pan, Iyad Rahwan, Manuel Cebrian, Riley Crane, Anmol Madan, and Alex Pentland. Time-critical social mobilization. *Science*, 334(6055):509–512, 2011.
- R. L. Plackett. The analysis of permutations. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 24(2):193–202, 1975.
- Reid Priedhorsky, Jilin Chen, Shyong Tony K Lam, Katherine Panciera, Loren Terveen, and John Riedl. Creating, destroying, and restoring value in wikipedia. In *Proceedings of the 2007 international ACM conference on Supporting group work*, pages 259–268. ACM, 2007.
- A. D. Procaccia and J. S. Rosenschein. Junta distributions and the average-case complexity of manipulating elections. *Journal of Artificial Intelligence Research*, 28:157–181, 2007.
- A. D. Procaccia, S. J. Reddi, and N. Shah. A maximum likelihood approach for selecting sets of alternatives. In *Proceedings of the 28th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, 2012. Forthcoming.
- Tao Qin, Xiubo Geng, and Tie-Yan Liu. A new probabilistic model for rank aggregation. In *Proceedings of the 2010 Annual Conference on Neural Information Processing Systems (NIPS)*, pages 1948–1956, 2010.
- Alexander J Quinn and Benjamin B Bederson. Human computation: a survey and taxonomy of a growing field. In *Proceedings of the 29th ACM Conference on Human Factors in Computing Systems (CHI)*, pages 1403–1412. ACM, 2011.
- M. Jordan Raddick, Georgia Brace, Pamela L. Gay, Chris J. Lintott, Carie Cardamone, Phil Murray, Kevin Schawinski, Alexander S. Szalay, and Jan Vandenberg. Galaxy Zoo: Motivations of citizen scientists. *Astronomy Education Review*, 12(1):010106, 2013. doi: 10.3847/AER2011021. URL <http://link.aip.org/link/?AER/12/010106/1>.
- Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Journal of Machine Learning Research*, 11(Apr):1297–1322, 2010.
- Jason Reed, M Jordan Raddick, Andrea Lardner, and Karen Carney. An exploratory factor analysis of motivations for participating in zooniverse, a collection of virtual citizen science projects. In *System Sciences (HICSS), 2013 46th Hawaii International Conference on*, pages 610–619. IEEE, 2013.

- Katharina Reinecke and Krzysztof Z Gajos. Quantifying visual preferences around the world. In *Proceedings of the 32nd ACM Conference on Human Factors in Computing Systems (CHI)*, pages 11–20. ACM, 2014.
- Ulf-Dietrich Reips. The web experiment method: Advantages, disadvantages, and solutions. In *Psychological experiments on the Internet*, pages 89–117. 2000.
- Ulf-Dietrich Reips. Standards for internet-based experimenting. *Experimental psychology*, 49(4):243–256, 2002.
- Paul Resnick and Hal R Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.
- Jakob Rogstadius, Vassilis Kostakos, Aniket Kittur, Boris Smus, Jim Laredo, and Maja Vukovic. An assessment of intrinsic and extrinsic motivation on task performance in crowd-sourcing markets. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media: Barcelona, Spain*, 2011.
- Christian Rudder. *Dataclysm*. Crown Publishers, 2014.
- Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th International World Wide Web Conference (WWW)*, pages 851–860. ACM, 2010.
- Mahyar Salek, Yoram Bachrach, and Peter Key. Hotspotting – a probabilistic graphical model for image object localization. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI)*, July 2013.
- M. J. Salganik and K. E.C. Levy. Wiki surveys: Open and quantifiable social data collection. Technical Report arXiv:1202.0500, 2012.
- Matthew J. Salganik. (personal communication), 2014.
- Matthew J Salganik, Peter Sheridan Dodds, and Duncan J Watts. Experimental study of inequality and unpredictability in an artificial cultural market. *science*, 311(5762):854–856, 2006.
- Megan E. Schwamb, Chris J. Lintott, Debra A. Fischer, Matthew J. Giguere, Stuart Lynn, Arfon M. Smith, John M. Brewer, Michael Parrish, Kevin Schawinski, and Robert J. Simpson. Planet Hunters: Assessing the Kepler inventory of short-period planets. *The Astrophysical Journal*, 754(2):129, 2012. URL <http://stacks.iop.org/0004-637X/754/i=2/a=129>.
- D. Sculley, Robert Malkin, Sugato Basu, and Roberto J. Bayardo. Predicting bounce rates in sponsored search advertisements. In *Proceedings of the 15th International Conference*

- on *Knowledge Discovery and Data Mining (KDD)*, 2009.
- D. Shahaf and E. Horvitz. Generalized task markets for human and machine computation. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI)*, pages 986–993, 2010.
- Aaron D. Shaw, John J. Horton, and Daniel L. Chen. Designing incentives for inexperienced human raters. In *Proceedings of the 2011 ACM Conference on Computer Supported Cooperative Work (CSCW)*, pages 275–284, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0556-3. doi: 10.1145/1958824.1958865. URL <http://doi.acm.org/10.1145/1958824.1958865>.
- Jesse Shore, Ethan Bernstein, and David Lazer. Facts and figuring: An experimental investigation of network structure and performance in information and solution spaces. *Organization Science*, (<http://dx.doi.org/10.1287/orsc.2015.0980>), 2015.
- Gerald H Shure and Robert J Meeker. A computer-based experimental laboratory. *American Psychologist*, 25(10):962, 1970.
- Choon-Ling Sia, Bernard CY Tan, and Kwok-Kee Wei. Group polarization and computer-mediated communication: Effects of communication cues, social presence, and anonymity. *Information Systems Research*, 13(1):70–90, 2002.
- Yaron Singer and Manas Mittal. Pricing mechanisms for crowdsourcing markets. In *Proceedings of the 22nd International World Wide Web Conference (WWW)*, pages 1157–1166. International World Wide Web Conferences Steering Committee, 2013.
- Peter K Smith, Jess Mahdavi, Manuel Carvalho, Sonja Fisher, Shanette Russell, and Neil Tippet. Cyberbullying: Its nature and impact in secondary school pupils. *Journal of child psychology and psychiatry*, 49(4):376–385, 2008.
- Charles Spearman. "general intelligence," objectively determined and measured. *The American Journal of Psychology*, 15(2):201–292, 1904.
- Jon Sprouse. A validation of amazon mechanical turk for the collection of acceptability judgments in linguistic theory. *Behavior research methods*, 43(1):155–167, 2011.
- Kate Starbird, Jim Maddock, Mania Orand, Peg Achterman, and Robert M Mason. Rumors, false flags, and digital vigilantes: misinformation on twitter after the 2013 boston marathon bombing. 2014.
- Hal Stern. Models for distributions on permutations. *Journal of the American Statistical Association*, 85(410):pp. 558–564, 1990. ISSN 01621459. URL <http://www.jstor.org/stable/2289798>.
- Greg Stoddard. Popularity dynamics and intrinsic quality on reddit and hacker news. In

- Proceedings of the 9th AAAI Conference on Web and Social Media (ICWSM)*, 2015.
- Brian L Sullivan, Christopher L Wood, Marshall J Iliff, Rick E Bonney, Daniel Fink, and Steve Kelling. ebird: A citizen-based bird observation network in the biological sciences. *Biological Conservation*, 142(10):2282–2292, 2009.
- Siddharth Suri and Duncan J Watts. Cooperation and contagion in web-based, networked public goods experiments. *PLoS One*, 6(3):e16836, 2011.
- Siddharth Suri, Daniel G Goldstein, and Winter A Mason. Honesty in an online labor market. In *Human Computation*, 2011.
- Diane Tang, Ashish Agarwal, Deirdre O’Brien, and Mike Meyer. Overlapping experiment infrastructure: More, better, faster experimentation. In *Proceedings of the 16th International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 17–26. ACM, 2010.
- Jeffrey Travers and Stanley Milgram. An experimental study of the small world problem. *Sociometry*, pages 425–443, 1969.
- Lyle Ungar, Barb Mellors, Ville Satopää, Jon Baron, Phil Tetlock, Jaime Ramos, and Sam Swift. The good judgment project: A large scale test. *AAAI Technical Report*, (FS-12-06), 2012.
- L. von Ahn and L. Dabbish. Designing games with a purpose. *Communications of the ACM*, 51(8):58–67, 2008.
- Luis Von Ahn. *Human Computation*. PhD thesis, Carnegie Mellon University, 2005.
- Luis Von Ahn and Laura Dabbish. Designing games with a purpose. *Communications of the ACM*, 51(8):58–67, 2008.
- Luis Von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum. re-captcha: Human-based character recognition via web security measures. *Science*, 321(5895):1465–1468, 2008.
- Fei-Yue Wang, Kathleen M Carley, Daniel Zeng, and Wenji Mao. Social computing: From social informatics to social intelligence. *Intelligent Systems, IEEE*, 22(2):79–83, 2007.
- Fei-Yue Wang, Daniel Zeng, James A Hendler, Qingpeng Zhang, Zhuo Feng, Yanqing Gao, Hui Wang, and Guanpi Lai. A study of the human flesh search engine: crowd-powered expansion of online knowledge. *Computer*, 43(8):0045–53, 2010.
- Duncan J. Watts. Computational social science: Exciting progress and future directions. *The Bridge on Frontiers of Engineering*, 43(4):5–10, 2013.

- Duncan J. Watts. Common sense and sociological explanations. *American Journal of Sociology*, 120(2):pp. 313–351, 2014. ISSN 00029602. URL <http://www.jstor.org/stable/10.1086/678271>.
- Duncan J Watts and Steven H Strogatz. Collective dynamics of small-world networks. *Nature*, 393(6684):440–442, 1998.
- Susan A Wheelan. Group size, group development, and group productivity. *Small Group Research*, 2009.
- Russell S Winer. Experimentation in the 21st century: the importance of external validity. *Journal of the Academy of Marketing Science*, 27(3):349–358, 1999.
- Joshua N. Winn. Transits and occultations. Chapter of the graduate-level textbook, EXOPLANETS, ed. S. Seager, University of Arizona Press, 2010. <http://arxiv.org/abs/1001.2010>.
- Justin Wolfers and Eric Zitzewitz. Prediction markets. Technical report, National Bureau of Economic Research, 2004.
- Anita Williams Woolley, Christopher F Chabris, Alex Pentland, Nada Hashmi, and Thomas W Malone. Evidence for a collective intelligence factor in the performance of human groups. *science*, 330(6004):686–688, 2010.
- Stefan Wuchty, Benjamin F Jones, and Brian Uzzi. The increasing dominance of teams in production of knowledge. *Science*, 316(5827):1036–1039, 2007.
- Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, pages 1192–1199. ACM, 2008.
- John I. Jr. Yellott. The relationship between Luce’s Choice Axiom, Thurstone’s Theory of Comparative Judgment, and the double exponential distribution. *Journal of Mathematical Psychology*, 15(2):109–144, 1977.
- Ming Yin, Yiling Chen, and Yu-An Sun. The effects of performance-contingent financial incentives in online labor markets. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI)*, 2013.
- H. P. Young. Condorcet’s theory of voting. *The American Political Science Review*, 82(4): 1231–1244, 1988.
- Morris Zelditch Jr. Can you really study an army in the laboratory. *A sociological reader on complex organizations*, pages 528–39, 1969.

Appendix A

Tutorial Text for Crisis Mapping Experiment

The participant tutorial for the experiment described in Chapter 4 was presented as an interactive step-by-step interface requiring participants to complete each step below, many of which required interaction with the mapping tool itself. Figure 6.1 shows a screenshot of the tutorial as an example, and Section 6.2.3 describes the design of the software used to present the tutorial. The text of the tutorial below is adapted from the web-based HTML version¹, which included icons.

Welcome to the Crisis Mapping Project, a Microsoft Corporation research project. Before beginning, you will complete a brief tutorial to familiarize yourself with the Crisis Mapping interface.

You would have completed this tutorial when you registered for the Crisis Mapping Project, but this will refresh your memory.

To learn more about the team conducting this research, please visit the following links:

- http://research.microsoft.com/en-us/groups/oess_nyc/
- <http://www.andrewmao.net/>

For the best experience, please maximize the window containing this task or make it as large as possible.

We need your help to better understand how teams of people can work together on a crisis mapping problem.

A “crisis map” is created in response to a natural or humanitarian disaster by monitoring social media for reports of events on the ground and placing them on a map.

In this project, you will work with a team of other crisis mappers to create a crisis map for Typhoon Pablo, a Category 5 hurricane which hit the Philippines in December 2012.

NOTE: This hurricane and the data you see is historical. This is not an actual current crisis.

¹See code at <https://github.com/mizzao/CrowdMapper>.

<p>To create the crisis map, you and your team members will be shown actual Twitter reports from Typhoon Pablo. Your team must figure out which reports describe relevant crisis events. In collaboration with your team, you must record these events describing the types of events, their descriptions, and their locations.</p> <p>Relevant reports include damage to infrastructure, reports of flooded regions, casualties, local residents that have been displaced, and evacuation centers. These should be recorded on the crisis map.</p> <p>Other reports are irrelevant or inaccurate and should not be recorded.</p>
<p>This tutorial will provide a general description of the task and explain the basic functions of the Crisis Mapping tool, but it is your job to figure out how to solve the problem.</p> <p>By effectively solving this problem and creating an accurate crisis map with your teammates, you can help us learn how to better respond to future natural disasters.</p> <p><i>In some cases, you may have a small team, or be the only person on your team, and you may not be able to finish recording events for all of the reports. You should focus on creating accurate, verifiable records over incomplete or partially categorized events.</i></p>
<p>Data reports appear as Tweets in the Stream on the left.</p> <p><i>This tutorial shows a few pieces of example data. You will see much more actual data in the task.</i></p>
<p>You should click on links to see what reports refer to, and to determine whether or not the data is relevant. You may search the Internet for places and phrases mentioned in reports. You may also hide tweets that appear to be irrelevant by clicking the red <i>✕</i> button. When you hide a tweet, it disappears for everyone.</p> <p>To continue, find one irrelevant tweet and hide it.</p>
<p>The navbar allows you to access the three primary ways to record information and work with others: <i>event records</i>, <i>map</i>, and <i>documents</i>.</p> <p>It also shows <i>notifications</i> you receive from other mappers.</p>
<p>The <i>Event Records</i> pane is the primary tool for recording events.</p> <p>Here you will see all of the event records you and your collaborators create.</p>
<p>Event records comprise several fields describing information about events.</p> <p>Your job is to fill in these fields for each relevant crisis event.</p> <p>Some information about the event can be found in the tweet itself, while other information can be found online or will require your own judgment.</p>
<p>Create a new event by clicking the "Create New Event" button at the <i>bottom of the page</i>.</p> <p>Create a new event now.</p>
<p>After you create an event, you will automatically be placed in edit mode.</p> <p>You may also edit an existing event record by double-clicking or by clicking the blue (pencil) button. You will see edits by other mappers in real-time.</p> <p>Only one person can edit an event record at a time, so when you're done, click the green (Save) button to save the event.</p>
<p>The first column shows records numbered in the order they were created.</p> <p>You can use this number to coordinate with other mappers in the chat room, which we'll show later.</p>

<p>The <i>Sources</i> column shows all tweets that refer to the same event.</p> <p>You can attach a tweet to the event record by dragging its <i>blue header</i> and dropping it on the record.</p> <p>Once you attach the tweet to an event record, you may hover over it to see its contents.</p> <p>Drag a tweet over to the event you just created.</p>
<p>Events that are relevant for crisis mapping will fit into one of these types:</p> <ul style="list-style-type: none"> • Damaged bridges • Damaged crops • Damaged hospitals/health facilities • Damaged housing • Damaged roads • Damaged schools • Damaged vehicles • Damaged infrastructure (other) • Death(s) reported • Displaced population • Evacuation center • Flooding <p>When you are editing an event record, you can <i>click a field</i> to enter a value. Fields that are empty will display the text (empty).</p> <p>Classify the event by clicking the <i>Type</i> field and choosing from the dropdown menu.</p>
<p>The <i>Description</i> column allows you to record a brief description of the event.</p> <p>Write something in the event's description.</p>
<p>The <i>Region</i> column shows the region where the event occurred.</p> <p>The Philippines has 17 administrative regions. You may need to look for regions and cities on other websites if you can't find them directly on the map.</p> <p>Add your best guess of the region from the tweet you just attached. Hover your mouse over the tweet you just attached to see its text.</p>
<p>The <i>Province</i> column is the province where the event occurred.</p> <p>There are 80 provinces in the Philippines.</p> <p>If you can find the province for the tweet, enter it in now.</p>

<p>The <i>Location</i> column shows the longitude/latitude for the event.</p> <p>If an event record does not have a location yet, you can place it on the map. We will show you how to do this momentarily.</p> <p>If you find these numbers from a map service on the Internet, you can use these numbers while editing the event.</p>
<p>You may also <i>add additional tweets to existing event records</i> if they refer to the same event. You can also <i>drag tweets from one event to another</i> to correct mistakes or reorganize records. If you <i>hover over an attached tweet</i> and find it to be irrelevant, you can use the (red X) to hide it.</p> <p>If you can find another tweet that refers to the same event as this one, add it to the record by dragging and dropping.</p>
<p>You can sort events by their data by clicking the arrows in the event header.</p>
<p>The <i>Map</i> displays a marker for each recorded event.</p> <p>If no longitude/latitude has been entered for an event, the marker will not appear on the map.</p>
<p>Use these map controls to pan around the map, and zoom in and out. You can also scroll and drag with your mouse.</p> <p>With your mouse, you can also pan by dragging and zoom by scrolling.</p> <p><i>This map is limited to the area around the Philippines, so you will need to first zoom in before you can pan the view.</i></p>
<p>You can press the (Locate) button to place an event on the map. You may need to search on the Internet to find a precise location for an event that you see.</p> <p>As you move your mouse over the map, note that the coordinates are shown in the <i>bottom right</i>. Use this to place the event in a precise location.</p> <p>Using the location information from the tweet, place the event on the map.</p>
<p>When looking at the map, you may hover or click on any event to view its details, and drag any event marker to change its geographical location.</p> <p>As you learn more information about an event, you will be able to pinpoint its location more accurately.</p>
<p>Notice that the longitude/latitude has automatically been updated in the event record. Now, you can also edit the coordinates by clicking them directly.</p> <p>Now, select the province, if you know it, and save the record using the (Save) button. This will let someone else edit the event.</p> <p>Save the event now.</p> <p>Once you have saved the event, you may click its location to see it on the map.</p>
<p>You may also check the accuracy of the events records created by other crisis mappers by verifying their data. The (checkbox) area shows that no one has verified this event yet.</p> <p>If you think an event record is accurate, you may verify it by hovering over this area and clicking the (Verify) button. You can also remove your vote for an event if you think it is no longer accurate.</p> <p>Since you created this event and believe it is accurate, check it now.</p>

<p><i>Documents</i> are available for recording information relevant to the task but not included in event records. They are shared with everyone else and can be edited simultaneously.</p> <p>Create a document by clicking the <i>New Document</i> button.</p>
<p>You can rename any document by clicking on its title and create new documents as necessary.</p> <p>Type something in the document. Other mappers will see this when they read the document.</p> <p>When you're done typing, continue to the next step.</p>
<p>The <i>User List</i> shows team members who are currently online in (green), and those who have stopped participating in (grey). You are shown in (black).</p> <p>When other users are online, you may hover your mouse over their name to invite them to chat.</p> <p><i>If you are working by yourself, you will see only your own name here.</i></p>
<p>You may create and join <i>chat rooms</i> to communicate with other users. Use the (New room) button to create a chat room.</p> <p>Create a new chat room now.</p>
<p>When other users invite you to chat or mention you in a chat room, you will see notifications appear here.</p> <p>Click on them to go to the chat room.</p>
<p>You will see a list of all chat rooms and the number of people in them, but you can only be in one chat room at a time.</p> <p>To join a chat room, just click on it.</p>
<p>When you are in a chat room, you can see who else is in that room. Each chat room displays the entire history of chats in that room.</p> <p>To leave a chat room, you can either click on "leave" or click on another chat room, which will then become active.</p>
<p>When you are in a chat room, you can use special characters to notify users and reference tweets or events.</p> <ul style="list-style-type: none"> • Use @ to mention a user by name. This will notify them. • Use ~ (tilde) to reference a tweet by its number. • Use # to reference an event by its number. <p>When you start typing the symbols above, an automatic filtered list will appear. Selecting an item in this list will create a link in your chat message that users can click on to jump to the tweet or event.</p> <p>Type something in the chat. For example, to tell another user to look at the event you created, type "@myself can you take a look at event #1?" You are talking to yourself here, but you will be chatting with other users in the actual task.</p>
<p>The project will run for one hour.</p> <p>You may quit the project at any time.</p> <p>If you perform no activity for several minutes, your session will be automatically idled.</p>

Your payment will be determined by how much time you personally spend working on the task, and also the overall performance of your team.

Your team may be competing with other teams, and your team's performance will be measured relative to either the best-performing team or other benchmarks. Your compensation will be calculated as:

(minutes you worked)/60 × (\$6 + your team bonus)

Your team bonus will be calculated as:

(Valid events your team correctly records)/(Highest number of valid events labeled by any team)

If you work for the full one hour, your compensation will be no less than \$6 and as much as \$15, depending on your team's performance.

Your payment will be shown during the task in the highlighted region. If you quit early or are idle you will only be paid for the fraction of time you work.

Thus, you should not just work hard yourself, but also help your team succeed.

Sometimes, you may be on a team by yourself. In this case, you will be evaluated against others who are also working by themselves.

Please read and acknowledge the *terms of use* for participating in this project.

Click to view the terms of use.

By clicking the checkbox below you confirm that you are at least 18 years of age, and that you understand what the project is about and how and why it is being done.

Some tweets may contain descriptions or images of human or animal suffering, and may be disturbing to some users. If you know that you are adversely affected by this kind of content, feel free to exit the experiment. By checking the box, you also acknowledge that you accept this possibility through your participation in crisis mapping.

(checkbox) I have read and acknowledged the terms of use.

If you have any questions about this project, please contact Andrew Mao at *mao@seas.harvard.edu*.

Click 'Finish' below to join a team of other workers and start crisis mapping!