Toward Theoretical Measures for Systems Involving Human Computation

A dissertation

submitted by

R. Jordan Crouser, B.S., M.S.

In partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

in

Computer Science

TUFTS UNIVERSITY

August 2013

Advisor: Remco Chang

For my family.

Acknowledgments

I would like to begin by thanking my adviser, Remco Chang. I did not intend to spend my life taking apart visualizations to see what makes them tick, or probing into psychology to try to detangle the human analytical process. And yet, listening to him talk about why Visual Analytics works (we don't know) and why it matters (because insight generation is much more intricate than it seems), I don't know that I'll ever be able to stop thinking about thinking. I am deeply grateful for the opportunity to have worked alongside him during the birth and development of the Visual Analytics Laboratory at Tufts. Through this experience, I have learned firsthand how to establish a research center, recruit collaborators, define a research agenda, court funding agencies and pursue often-elusive answers to truly important questions. This opportunity is rare, and has helped shape me both as a researcher and as an intellectual. For his contagious enthusiasm, for his guidance and for his trust, I owe him an unrepayable debt. It has been an honor and a pleasure to be his first doctoral student; I look forward to the generations to come.

I would also like to thank my unofficial co-adviser, Ben Hescott. His list of academic achievements is a testament to his intellect, without which much of these results would have remained undiscovered. I am grateful to have served under him as a GIFT Fellow, and I know that this experience will prove invaluable even as I pursue a career outside the formal classroom. However, my gratitude to him runs much deeper. We have in our lives the opportunity to cross paths with only a handful of truly exemplary mentors and friends. From his relentless pursuit of new ways to engage and cultivate the growing minds of future generations to his unflinching compassion in the face of another person's humanity, he is exemplary. Ben, thank you. From me, and from all of us.

Because my academic path has been somewhat unusual (as has, perhaps, the wider path of my life), there has been a great deal of confusion as well as occasional frustration. Without the support of my family and friends, this undertaking would surely have proven intractable. To Jisoo, Dana, Gillis, Jim, and James: thank you for your friendship. To my parents, Lill and Donald, and their incredible (respective) co-parents, Dave and Lisa: thanks for your unending love and support, even when I do things you don't quite understand. To my big sisters, Justyn, Vanessa and Olivia; to my not-so-little-anymore sister, Sarah; to my wonderful cousins, nieces and nephew, who are the light of my life; and in memory of my grandmothers Norma and Saralyn, who were my unyielding champions and who both passed during the writing of this dissertation, I dedicate this work.

And to Morganne: how do you thank someone for the hundred thousand tiny gestures that fill a life with warmth? For being at the same time immensely patient and terribly pushy, to you I am forever grateful. I'm sure my fumbling for words will not surprise you in the least, and I hope that it will make you smile.

This thesis represents the collective efforts of an amazing group of interdisciplinary collaborators, without whom this work would have been impossible: from the VALT, Alvitta Ottley, Eli T. Brown, Jeremy G. Freeman, Daniel E. Kee, Shaomeng (Samuel) Li, Orkun Özbek, and Jieqiong (Helen) Zhao; from the Tufts HCI Lab, Michael Horn, Erin T. Solovey, Evan Peck, Dan Afergan, and Beste Yuksel; from the Tufts CS Department, Dave Kiger and Andrew Winslow; from the University of the District of Columbia, Dong Hyun Jeong; from Brown University (and later Aptima, Inc.), Caroline Ziemkiewicz; from Lustick Consulting, Ian Lustick, Miguel Garces, and Brandon Alcorn. For the remainder of this thesis, I will adopt the collective pronoun "we" to acknowledge the critical contributions of my collaborators. For their time, their insight, their dedicated pursuit of knowledge, I am eternally grateful. May this be only the beginning.

In closing, I would like to thank the members of my committee, Professors Robert J.K. Jacob and Mary Glaser of Tufts University, and Dr. Matthew C. Schmidt of MIT Lincoln Laboratory for their time and insight. This work is supported in part by grant number BCS-0904646 from the National Science Foundation. Data used in this research was provided with support received from the Defense Ad-

vanced Research Projects Agency (DARPA) through the Advanced Technology Laboratories wing of Lockheed-Martin in the Integrated Crisis Early Warning System (ICEWS) project (Prime Contract #FA8650-07-C-7749). The results and findings in no way represent the views of the Department of Defense or DARPA.

R. Jordan Crouser

TUFTS UNIVERSITY

August 2013

Toward Theoretical Measures for Systems Involving Human Computation

R. Jordan Crouser

Advisor: Remco Chang

As we enter an age of increasingly larger and noisier data, the dynamic interplay between human and machine analysis grows ever more important. At present, balancing the cost of building and deploying a collaborative system with the benefits afforded by its use is precarious at best. We rely heavily on researcher intuition and current field-wide trends to decide which problems to approach using collaborative techniques. While this has led to many successes, it may also lead to the investment of significant time and energy into collaborative solutions for problems that might better have been (or have already been) solved by human or machine alone. In the absence of a secret formula to prescribe this interplay, how do we balance the expected contributions of human and machine during the design process? Can we describe the high-level complexity of these systems with the same robust language as we use to describe the complexity of an algorithmic system? In this work, we investigate the complementary nature of human and machine computation as used in visual analytics and human computation systems, and present a theoretical model to quantify and compare the algorithms that leverage this interaction.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

As we enter an age of increasingly larger and noisier data, the dynamic interplay between human and machine analysis grows ever more important. Researchers and toolbuilders work to better understand and support these analytical processes through systems that couple interactive interfaces with robust computational support. These systems leverage the acuity of the human visual system as well as our capacity to understand and reason about complex data, nuanced relationships, and changing situations. By pairing the human analyst with a machine collaborator for computational support, we hope to overcome some of the limitations imposed by the human brain such as limited working memory, bias, and fatigue. Similarly, we rely on the intuition that the lived experience, perceptual advantage, and adaptability of the human analyst may prove crucial in areas where purely computational analyses fail.

This strategy has lead to incredible advances in the development of novel tools for use in many historically challenging domains. In just the past five years, we have seen interactive data-driven systems shift financial fraud detection from a black art to a science [CLG^{+}08], and witnessed the discovery of new protein structures predicted with help from the crowd [CKT^{+}10]. We have also made dramatic improvement in the tools available for modeling and predicting complex social behavior, such as those we designed for the analysis of political systems [CKJC12] which will be discussed at length in Chapter 3. With so many promising examples of human-machine collaboration in the literature and everyday life, how do we tell if a new problem would benefit from similar strategies – and if so, how should we allocate computational tasks?

At present, balancing the cost of building and deploying a collaborative system with the benefits afforded by its use is precarious at best. We rely heavily on researcher instinct and current field-wide trends to decide which problems to approach using collaborative techniques. While this has led to many successes, it may also lead to the investment of significant time and energy into collaborative solutions for problems that might better have been (or have already been) solved by human or machine alone.

In addition to the challenges raised in determining when human-computer collaboration is appropriate, we presently lack appropriate mechanisms for evaluating systems once we've built them. While in-house experimentation and in situ studies help us determine whether or not our systems are useful, they fall short of explaining *why* we see the results we do. In truth, we critique [KDHL08] these systems, rather than scientifically validate their performance. This often results in the rote and incremental recycling of known techniques, as we are left to speculate about the reasons underlying observed (in)effectiveness. In order to advance the science of human-computer collaborative systems, it is important that we develop theoretical models of the complementary roles played by both human and machine to better inform our reasoning about their performance. One such model will be proposed in Chapter 4.

A system's effectiveness is determined by how well it leverages its resources while minimizing waste. While we have come a long way from listing tasks best assigned to human or machine [Fit51], appropriate function allocation in collaborative systems is still far from a perfect science [She00]. In the absence of a secret formula to prescribe this interplay, how do we balance the expected contributions of human and machine during the design process? Is it possible to describe the high-level complexity of human-computer collaborative systems with the same robust language as we use to describe the complexity of an algorithmic system?

1.1 Purpose and Outline of this Work

The purpose of this dissertation is to investigate the complementary nature of human and machine computation as used in visual analytics and human computation systems, as well as to develop theoretical models to quantify the algorithms and systems that leverage their interaction. While characterizing human processing through cognitive modeling and other means is of critical import to the development of a holistic understanding of the cost and benefits of human-computer symbiosis, the topic of modeling the human brain in general is beyond the scope of this thesis. The aim of this work is to characterize and quantify the *use of human processing power as part of an algorithmic process*, rather than to model and measure the cost of the human's computational processes themselves. By separating questions of per-operation cost from questions of resource utilization, we posit that the models presented in this work will be robust even as more nuanced and complete models of the human brain come to light. It is through this focus on the use of human processing as a computational resource and its impact on computability that this dissertation contributes to the field of human computing and human-computer interaction.

Toward that end, this dissertation is organized as follows. We begin with an overview of related work (Chapter 2). We then describe the development of two visual analytics systems for use in the analysis and prediction of the behavior of political systems in southeast Asia, and report the results of an expert comparison of these systems against traditional analytic practices in this area (Chapter 3). This vignette demonstrates the utility of a human-computer collaborative approach in a complex real-world application domain. We will then go on to consider the relative strengths of human and machine collaborators, and provide a framework for cataloguing this and other existing work in human-computer collaborative systems according to these affordances (Chapter 4).

From there, we discuss the open problem of complexity measures for algorithms involving human computation, and present the Human Oracle Model as a high-level tool for characterizing and comparing these algorithms (Chapter 5). We demonstrate the utility of this model for comparing and analyzing several well-known human computation systems for image labeling (Chapter 6), and subsequently discuss how this model can be used to characterize the space of human computation (Chapter 7). Finally, we will discuss the model's limitations as well as its potential for broader impact (Chapter 8), and provide a summary of the key contributions of this dissertation (Chapter 9). We hope that this work will leave the reader with an improved understanding of the complementary strengths of human and machine, how human and machine computation can be interleaved as part of an algorithmic process, as well as actionable information about best practices for real world design.

Chapter 2

Related Work

The earliest known reference to the word *computer* dates back to the early 17th century, at which time it referred to humans tasked with performing manual calculations. This definition would survive for the better part of two and a half centuries, before gradually being reappropriated to refer to machines performing similar calculations. In more recent history, the capacity of the human brain to contribute to computational processes has come back into the spotlight. In this chapter, we will provide a brief overview of relevant literature on the use of human processing power in computation.

2.1 Leveraging Human Expertise in Computation

Lived experience and the associated knowledge developed over significant periods of time can prove difficult, if not impossible, to encode into a mechanical computation system. At the same time, this supplemental information about the larger domain is often of critical importance to solving real-world problems. Because of this, it is in some cases more advantageous to leverage the human analyst's expertise directly rather than invest significant resources in approximating it. In machine learning, this expertise is used to generate labeled training datasets. These methods have proven highly effective in handwriting recognition [XKS92], classifying text documents [Seb02], learning realistic human motion from video [LMSR08], and other areas where predetermining a clear set of classification rules is intractable. Similarly, visual analytics systems rely on human expertise and the "human capacity to perceive, understand, and reason about complex and dynamic data and situations" [TC05] to identify patterns in data that are difficult or impossible to detect using purely mechanical means. Systems leveraging expert input have demonstrated success in analyzing trends in medical image datasets [BJVH11], detecting fraudulent financial transactions [CLG$^+$08], diagnosing network faults [LLKM10], and many other applications. In Chapter 3, we will discuss the design and evaluation of two systems for analyzing and predicting patterns in political systems. For a survey of other visual analytics systems, see Keim et al. [KKEM10]. In both machine learning and visual analytics there is an implicit understanding that human time and effort is *expensive*, and that this resource should therefore be utilized as efficiently as possible.

2.2 Relative Strengths of Human and Machine

In 1951, Fitts made the first published attempt to categorize tasks better allocated to humans or machines [Fit51], often abbreviated in the literature as HABA-MABA (*"humans-are-better-at / machines-are-better-at"*). While for many years this list was viewed as mantra for the division of labor, frequent and consistent technological advances in computation, automation and robotics make function allocation and the HABA-MABA list a moving target. The distinction between human and machine is now less clear. For example, while in the 1950s humans were indeed better at storing large amounts of information, today's machines far exceed the storage capacity previously imagined, and the advent of distributed storage is rapidly enabling the outpacing of human memory by machines.

While the goal of Fitts' lists was simply to compare humans and machines for basic labor division, for many years it was mistakenly interpreted as gospel for function allocation for human-machine collaborative systems. Jordan [Jor63] criticized this approach, stating that humans and machines are complementary rather than antithetical. Price [Pri85] also supported this view, arguing that function allocation could be better conceptualized as an interactive process rather than a divisive listing and that there may exist several optimal solutions for a given problem. Nonetheless, Fitts' list laid the foundation for thinking about the respective strengths of humans and machines.

In recent years, researchers have argued that the original understanding of function allocation and Fitts' list no longer makes sense [She00]. Dekker and Woods [DW02] also provided counterarguments to the validity of Fitts' list by arguing that human-machine interaction transforms human practice, causing analysts to adapt their skills and analytic processes. They advocated for a shift in attention, moving away from allocation of tasks to a focus centered on how to design for harmonious human-machine cooperation. That is, how do we get humans and machines to play nicely, and work effectively?

2.3 Human Computation: a Brief Introduction

In this section, we will describe some important concepts and terminology that will be utilized extensively throughout this manuscript. We begin with a short introduction to *human-computer collaboration* and *human computation*.

2.3.1 Human-Computer Collaboration

In a 1993 symposium at AAAI, researchers from a variety of backgrounds came together to discuss challenges and benefits in the emerging field of human-computer collaboration. They defined **collaboration** as *a process in which two or more agents work together to achieve shared goals*, and **human-computer collaboration** as *collaboration involving at least one human and at least one computational agent* [Ter95]. This collaboration has also been called *mixed-initiative systems* [Hor99], in which either the system or the user can initiate action, access information and suggest or enact responses [TC05]. Mixed-initiative systems have been explored in diverse areas including knowledge discovery [VP99], problem-solving in AI [FA98], procedural training in virtual reality [RJ99] and much more.

2.3.2 Terminology

In his 2005 doctoral thesis [VA05], Luis von Ahn introduced the term **human computation**; that is, harnessing human time and energy for solving problems that have to date proven computationally intractable. This is accomplished by treating human brains as processors in a distributed system. It is important to note that the term *human computation* is not synonymous with *collective intelligence*, *crowdsourcing*, or *social computing*, although they are related. Before we continue, we will first define these terms in the interest of developing a context for defining human computation.

Definition *Crowdsourcing* is the practice of obtaining services, ideas, or content by soliciting contributions from a large group of people.

Definition *Collective intelligence* is the notion that groups of individuals working together can display intelligent behavior that transcends individual contributions.

Definition *Social computing* is the intersection between people's social behaviors and their interactions with technology.

In many cases, a single system could be classified under more than one of these headings. At the same time, none of them fully captures the notion of human computation. As such, there are many working definitions of human computation in the literature:

> . . . using human effort to perform tasks that computers cannot yet perform [LVA09]. . .
>
> . . . a technique that makes use of human abilities for computation to solve problems [CKY09]. . .
>
> A computational process that involves humans in certain steps [YZG+08]. . .

> . . . systems of computers and large numbers of humans that work together in order to solve problems that could not be solved by either computers or humans alone [QB09]. . .

Working from these definitions, we can begin to come to consensus regarding what constitutes human computation. First, the problem must involve some form of *information processing*. This may occur as part of an algorithmic process, or may emerge through the observation and analysis of technology-mediated human behavior. Second, human participation must be *integral to the computational system or process*. In this work, we will consider systems with only superficial human involvement to fall outside the scope of human computation.

2.3.3 Human Computation in Practice

With the advent of online marketplaces providing an on-demand workforce for microtasks, we have seen an explosion of work utilizing human processing power to approach problems that have previously proven intractable. Examples include image labeling [DSG07, HCL$^+$09, VAD04, VAGK$^+$06, WY12, SDFF12], optical character recognition [VAMM$^+$08, NGR$^+$11, CS11], annotating audio clips [LVADC07, ME08, BOTL09], evacuation planning [SRSJ11] and protein folding [CKT$^+$10]. Human computation has also been used to develop logical models of mutual exclusion [CCH11], as well as find cases where a predictive model is confident but incorrect [AIP11]. Intuitively, human computation has shown great promise in helping refine models of human behavior [BKAA11, LALUR12] and natural language [KJB12, WY10, SCVAT10, CPK09], and has even been used to recursively define subtasks for future human computation [KCH11]. For detailed surveys of research in the area of human computation, please see [QB11, YCK09].

While research in this area has demonstrated much success in harnessing humans' computational power, there is a temptation to use human workers as an easy out. In his article entitled "Why I Hate Mechanical Turk Research (and Workshops)" [Ada11], Eytan Adar argues:

> We should not fool ourselves into believing that all hard problems [warrant human computation] or completely distract ourselves from advancing other, computational means of solving these problems. More importantly, we should not fool ourselves into believing that we have done something new by using human labor. . . Showing that humans can do human work is not a contribution.

This sentiment has prompted fascinating debate about *when* and *how* to leverage human intelligence in computation.

2.4 Balancing Human and Machine Contributions

Under our working definition of human computation, we see that crowdsourcing is just the tip of the iceberg. We can think of human computation as a kind of human-computer collaboration, dividing the computational workload between both human and machine processors. Along a continuum between human-heavy and machine-heavy collaboration [BL10], crowdsourcing falls at one extreme (see Fig. 2.1).

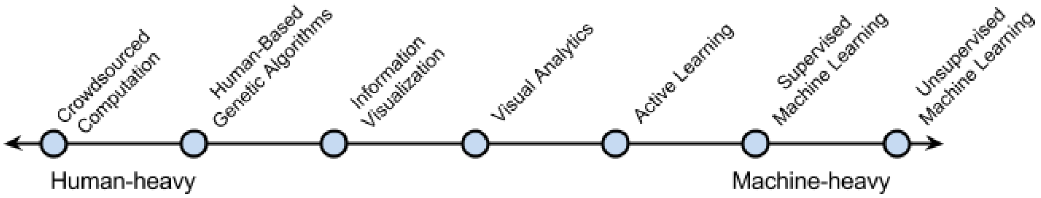

Crowdsourced Computation
Human-Based Genetic Algorithms
Information Visualization
Visual Analytics
Active Learning
Supervised Machine Learning
Unsupervised Machine Learning
Human-heavy
Machine-heavy

Figure 2.1: Examples of human computation along a continuum from human-heavy to machine-heavy collaboration.

With few exceptions, the computational burden falls almost entirely on the human collaborators in typical crowdsourced computation applications such as image labeling and text translation. Human-based genetic algorithms also fall on the human-heavy end of the continuum, as the human agents determine both population fitness and genetic variation. In these systems, the primary role of the machine collaborator is to distribute tasks and collect results, a role with relatively trivial computational requirements. On the other extreme, algorithms for unsupervised learning functions with near autonomy from the human collaborator. Here, the human's role is to set the parameters of the algorithms and to verify the results. In the center, we see a number of algorithmic approaches that attempt to maximize the contributions from both collaborators in a joint effort to solve complex problems.

Without question, the term human computation spans a wide range of possible applications and computational distributions. Among all these, many of the most interesting and successful human computation systems not only balance the contribution of human and machine, but also leverage the complementary computational strengths of both parties. In Chapter 4, we will explore some of these strengths and how they can impact the distribution of labor in a human computation system.

2.5 Challenges in Using Human Computation

While it is may be tempting to view human processing as panacea to many challenging computational problems, it is important to recognize some fundamental challenges to using human computation as a computational resource.

2.5.1 Quality Control

As with any biologically-generated signal, the results of human computation are inherently noisy. While processes leveraging expert computation often assume that expertise implies accuracy, general human computation requires the integration of quality control measures in order to ensure quality [Gri11, Lea11]. In many cases, intelligently combining individual responses can produce higher quality than any individual contribution [GVGH12]. In Games with a Purpose, implicit validation methods such as *output-agreement* [VAD04], *input-agreement* [LVA09], or *complementarity-agreement* [LA11] are woven into the game mechanics. For applications with a larger number of contributors, simple majority vote from a collection of users is sufficient to validate a proposed solution [BLM^{+}10]. More advanced voting rules can provide improved guarantees on accuracy over basic majority voting under some noise models [JL11, MPC12], and the level of redundancy can be adjusted on the fly to ensure a confidence threshold is met [BKW$^+$11].

Tournament selection can also improve quality over independent agreement in complex tasks [SRL11], leveraging humans' ability to recognize correct answers even when they have a limited ability to generate them. In addition, new active learning paradigms that balance traditional close-to-boundary sampling with global distribution of unlabeled data have shown promising results with noisy, unreliable oracles [ZSS11], as have matrix factorization methods for counteracting sparse, imbalanced samples [JL12]. For *open problems*, where answers are being sampled from a countably infinite rather than finite set, decision-theoretic models can be useful for quality control [LW$^+$12].

In addition to controlling for the quality of individual answers, human computation systems are also concerned with the overall quality of individual contributors. Some systems interject questions with known correct answers to directly estimate a contributor's quality [OSL$^+$11]. Others have proposed using support vector machines [HB12] or Z-score outlier detection [JL11] to identify those whose responses are excessively noisy, as well as using confusion matrices to separate contributors exhibiting occasional bias from true substandard contributors [IPW10]. These methods are intended to filter out workers of unacceptable quality from the resource pool. Though some can be performed on the fly, throughout the remainder of this dissertation we will assume that this filtering has occurred during a preprocessing step, rather than during the execution of the human computation system itself.

2.5.2 User Modeling for Human Computation

In addition to introducing noise, human contributors operate within a complex system of social, behavioral, and economic factors. To better understand the role these factors play in the design of effective human computation systems, researchers have developed and analyzed models of the interaction between tasks, environments, and contributors. Several studies have contributed semi-ethnographic characterizations of workers on Amazon's Mechanical Turk [SRIT10, SGM11], as well as models for how workers enter and exit the market and the factors that influence how they select tasks [FHI11]. These models can help inform optimal incentive structures [HZVvdS12] and workflows [Dai11], as well as contribute to quality control [WBPB10].

While these models can be helpful in designing more efficient systems, the human's underlying computational processes remain largely a mystery. Sadly, our ability to model how the human brain computes is hindered by a limited understanding of the biological mechanisms that enable that computation. Cognitive modeling has demonstrated success in simulating processes such as visual word recognition [Dav10] and memory recognition [NO03], but it is unclear how to compare between these models or to determine whether a given model is minimal and complete. Until our understanding of the cognitive processes involved in computation is more fully developed, it seems likely that the human will generally remain a (somewhat finicky) black box. In the interim, we can begin to develop a higher-level notion of the complexity of systems involving human computation.

2.6 Measuring the Complexity of Human+Machine

Existing complexity models classify computational problems by evaluating the time and space required to solve the problem using a computer. Under these models, many interesting real-world problems are known to be computationally infeasible, even if the path to finding the solution is clear. For example, we know how to solve the Traveling Salesman problem, but computing the solution is intractable for all but a handful of special cases. Other problems, like general image recognition, have no known solution and are believed to be unsolvable by even the most powerful machines.

In contrast, many of these problems appear relatively easy for humans. Some of this disparity can be attributed to the advantages of robust biological perceptual systems which have been honed through millennia of evolutionary refinement. While our understanding of the biological mechanisms that enable computation in the human brain is still limited, we have evidence to support the intuition that human computational processes are different from, and in may cases complementary to, mechanical computation.

Emerging research in Artificial Intelligence extends theoretical models of complexity to include computation performed by human-level intelligence [**?**, DSC10, Yam11, Yam12, Yam13]. One major contribution of this extension is that it provides a mechanism to verify the existence of a human-level intelligence by outlining classes of problems which only such an intelligence could solve. If a solution to any such problem could be yielded purely through mechanical computation, that would be sufficient to prove that the machine performing the computation was exhibiting human-level intelligence.

Research in the field of Artificial Intelligence seeks to model and emulate human intelligence using a machine. Research in human computation leverages *actual* human intelligence to perform computationally-difficult tasks. Both fields hinge on the long-held belief that there exist problems that require human-level intelligence and reasoning to solve. Because of this relationship, we believe that theoretical models from the Artificial Intelligence community may be a useful starting point for understanding and comparing human computation problems and their solutions. Beginning in Chapter 5, we will expand upon one such model and adapt it for use in measuring the complexity of human computation systems. This dissertation provides a critical first step in quantifying the use of human input as a computational resource, and helps us to better understand the intricate relationships between different problems and problem families when viewed through the lens of human computation.

Chapter 3

Two Visual Analytics Systems for Political Science

This chapter is based on the paper:

- Crouser, R., Kee, D. E., Jeong, D. H., & Chang, R. Two visualization tools for analyzing agent-based simulations in political science. *IEEE Computer Graphics and Applications*, 32(1), 67-77, 2012.

3.1 Introduction

In this chapter, we present two human-computer collaborative systems designed to support inquiry and inference by social scientists using agent-based simulations to model political phenomena. In collaboration with domain experts, we designed these systems to provide interactive exploration and domain-specific data analysis tools. Through in situ analysis by expert users, we validated that these systems provide an efficient mechanism for exploring individual trajectories and the relationships between variables. In addition, we demonstrated that these systems more effectively support hypothesis generation when compared with existing best practices by enabling analysts to group simulations according to multidimensional similarity and drill down to investigate further.

3.2 Domain Characterization

Behavioral simulation analysis is an important component of social and political science research. In studying these models, scientists seek to uncover the sociopolitical and socioeconomic forces at work in controlling and influencing group behaviors, as well as to make predictions about behavioral patterns using data collected in the real world. Better understanding of how these forces influence group behavior and the ability to make more accurate predictions can greatly influence how we view real-world behavioral systems and better inform decisions regarding domestic stability, foreign policy and more.

The first step in this process is constructing an accurate model. Research in these areas often utilizes a technique called *agent-based modeling* (ABM). In ABM, a behavioral system is modeled as a collection of autonomous entities or *agents*. Each agent interacts with other agents according to a set of rules and goals, and over time it may influence and be influenced by the agents around it. ABM has been used to model complex behaviors such as collaboration [Axe97], conflict [SPRK03], violence [BB00], and population change [AED$^+$02]. Agent-based models have also been used to identify a country's political patterns, which might indicate the imminence of civil unrest and help predict catastrophic events [LAGR10].

After building an agent-based model from existing political theories based on observed behaviors and interactions, the model is then seeded with data collected in the field about political party affiliation, level of violence, protest, regional and local conflict, and more [AHG11]. Using this information, the agent-based model produces a large amount of data representing a distribution of possible behavioral patterns over a fixed period of time. Analysts then use this data to construct a cohesive narrative explaining the relevant interactions as well as to identify interesting or highly likely future outcomes.

As computing power becomes more widely available, scientists are able to simulate increasingly complex systems. This in turn generates increasingly large datasets, which must then be analyzed and interpreted. Correctly interpreting these simulation results can help social and political scientists to better understand the forces at work in complicated social behaviors, such as those leading to patterns of violence and socioeconomic repression, political unrest and instability, and even help identify factors that might lead to catastrophic events. Conversely, incorrectly interpreting the results of these simulations can result in suboptimal decision-making and misallocation of resources in high impact, real-world situations.

Unfortunately, the existing methods and tools available to social scientists for analyzing simulation results are not able to support datasets of this magnitude, making it difficult for scientists to effectively interpret and analyze the results of these simulations [Lus02]. While statistical analysis of the resulting data can be performed, it often proves insufficient. Due to the complex nature of these simulations, expert analysis of the resulting datasets is required to interpret the results as valid behavioral patterns and fully understand the forces controlling the interactions observed in the simulation. The size of the data is so large that it would require countless hours to examine by hand, and so the data must often be simplified and some detail sacrificed in the interest of conserving analyst time and energy.

Data size and dimensionality are not the only challenges facing social scientists when using large-scale agent-based simulations to model complex behaviors. ABM is a stochastic simulation technique, utilizing small random perturbations to the interaction rules and running each simulation hundreds or even thousands of times times to avoid local minima and to generate a distribution of sample behavioral patterns. Because of this, it is critical for analysts to be able to compare simulated behaviors between and across distinct runs, and to be able to piece together many simulation runs into a single, cohesive overview.

For these reasons, computational support and effective, domain-specific visualization tools are critical for effective analysis of these simulations. By understanding the patterns being modeled by the simulation, scientists can better understand the sociopolitical forces at work in real-world social and political systems, which can in turn enable them to better inform decision-makers and international policy. To begin to address this need, we formed a collaborative partnership with domain experts to investigate novel approaches for supporting this analysis process.

3.3 Design Considerations

Through informal brainstorming sessions with a group of domain experts, we identified three areas of critical need that are insufficiently addressed by existing analytical systems for use in exploring agent-based simulation data:

- Support for exploring the dataset as a whole to generate initial hypotheses
- Efficient mechanisms for the comparison of individual simulation runs
- Incorporation of domain expertise into the data analysis tool

Using these three design considerations as a foundation, we developed interactive exploratory visual analytics systems to support analysis of agent-based models in political science. Each of these systems utilizes a coordinated multiple views architecture, allowing the analyst to customize the views to suit her analytical process. The systems are developed using C++, OpenGL, and wxWidgets, and as such are deployable to any machine regardless of its operating system.

To evaluate these systems, we performed an expert analysis with a group of analysts working with data from an agent-based simulation of political violence and unrest in Thailand. From this analysis, we found that most analysts considered our systems to be invaluable in supporting and streamlining their analytical processes. In collaboration with these experts, we also identified areas for further refinement of these systems.

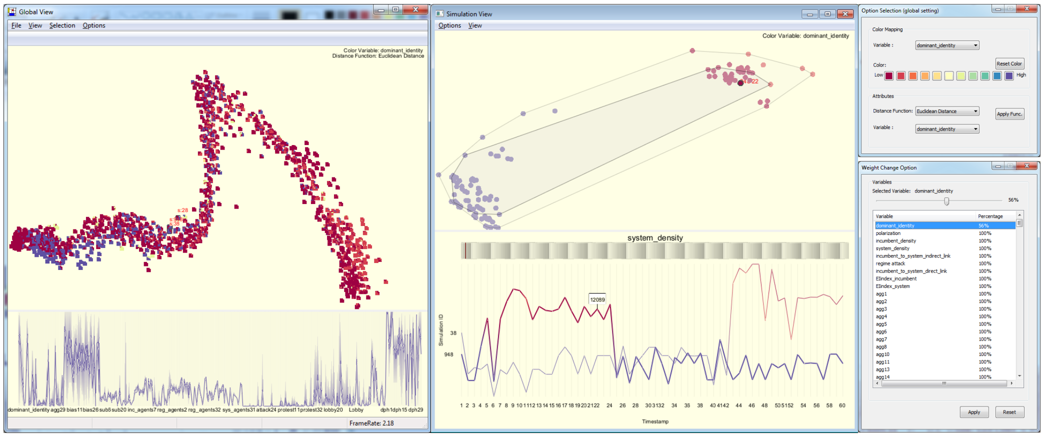
3.4 Macroanalysis using MDSViz

To address the first area of critical need, we present MDSViz, a visual analytics system designed to enable to analyst to examine the aggregated data, determine the similarities and differences between high-dimensional simulation runs, and identify trends and outliers for further exploration. Through our informal interviews, analysts reported that their existing best practices involved using line graphs and statistical plots of each dimension in order to make comparisons, and manually comparing the values of individual variables to drill down into a single run. This process is laborious, highly error-prone, and fails to provide a real overall sense of how the dimensions interact with one another.

3.4.1 System Design

To support global analysis across all simulation runs, the data from all simulations are centrally managed and projected to highlight similarities. Because the simulations have high dimensionality (1,000 simulations × 60 timesteps × 351 attributes), a distance function is necessary to describe the similarity of two given states (see Section 3.6 for a detailed discussion on selecting a distance function). With an appropriate distance function, multidimensional scaling (MDS) is applied to reduce the dimensionality of the data. Since the dimensionality is high in our input data (a distance matrix of 60,000 × 60,000 is possible), the system computes the mean variance of each simulation by referencing all 60 timesteps. Each simulation is represented as mean values of 351 variables, and so the size of the distance matrix can be reduced to 1,000 × 1,000. Based on this generated distance matrix, MDS is performed to reduce the dimensionality of the simulations further.

To support analysis on complex political simulations, the MDSViz system is designed using a coordinated multiple view (CMV) architecture. Within the CMV framework, any interaction with one view is immediately reflected to all the other views. To effectively coordinate each view, we implemented an interaction manager which handles all keyboard and mouse interactions. In addition, the selection operation in all views and the zooming mechanism in the Projection and Cluster views helps users focus their attention on interesting simulations or timesteps. A detailed explanation of supported interactions in each system is included in following sections.



Color Variable: dominant_identity
Distance Function: Euclidean Distance

Color Variable: dominant_identity

Variable : dominant_identity

Color: Low 🟥 🟥 🟧 🟧 🟨 🟨 🟩 🟩 🟩 🟦 🟪 High

Distance Function: Euclidean Distance

Variable : dominant_identity

Selected Variable: dominant_identity

56%

Variable	Percentage
dominant_identity	56%
polarization	100%
incumbent_density	100%
system_density	100%
incumbent_to_system_indirect_link	100%
regime attack	100%
incumbent_to_system_direct_link	100%
EIindex_incumbent	100%
EIindex_system	100%
agg1	100%
agg2	100%
agg3	100%
agg4	100%
agg5	100%
agg6	100%
agg7	100%
agg8	100%
agg10	100%
agg11	100%
agg13	100%
agg14	100%

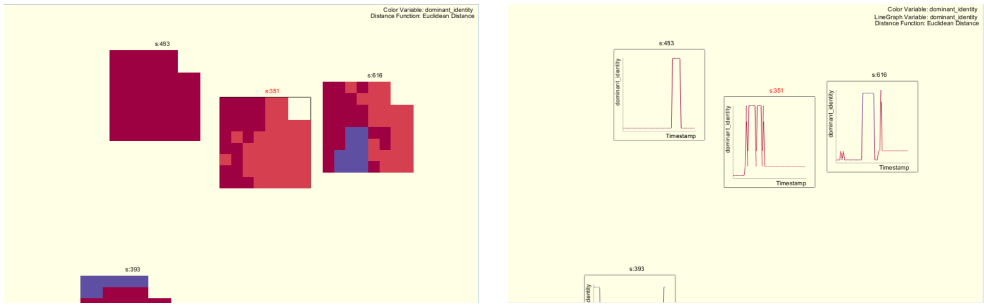
(a) (b) (c)

Figure 3.1: The MDSViz system, utilizing a coordinated multiple views (CMV) architecture: (a) a Global view using MDS Projection (top) and parallel coordinates (bottom), (b) Simulation view, and (c) control panels

Projection View

All simulations are represented by applying a distance function and multi-dimensional scaling (MDS) in the Projection view. Because there are limitations on applying MDS directly to large-scale input data, a statistical variance analysis is performed in advance. Mean variance is computed to determine the center of the variable distribution for each simulation, and a distance function is then applied. Although finding a semantically meaningful distance function is important, identifying the appropriate contribution of all variables requires significant computational time. We use a simple Euclidean distance function and allow the user to manually control the weighted contribution of each dimension. MDS is then applied to reduce dimensionality of the simulations. By default, we run MDS for 1,000 iterations, though this parameter can be tuned.

Figure 3.1(a)-top shows all 1,000 political simulations in the Projection View. Each simulation is represented as a pixel-oriented glyph by arranging each timestep following an 8th order Hilbert curve. This technique has the advantage of providing continuous curves while maintaining good locality of information. For mapping each timestep, we set the Hilbert curve order to 8 which covers up to 8 × 8 sizes. Color coding is then used to represent the selected variable at each timestep. This


Color Variable: dominant_identity
Distance Function: Euclidean Distance
s:483
s:351
s:616
s:393
Color Variable: dominant_identity
LineGraph Variable: dominant_identity
Distance Function: Euclidean Distance
s:483
dominant_identity
Timestamp
s:351
dominant_identity
Timestamp
s:616
dominant_identity
Timestamp
s:393
dentity

(a)

(b)

Figure 3.2: Glyph representation can be toggled while navigating the projection space: (a) Pixel-oriented glyphs display all 60 timesteps of each simulation following the Hilbert curve ordering method and (b) Line graphs represents the temporal changes through time on a selected variable.

parameter can be selected by the user in the the control panel (see Fig. 3.1(c)-top). Alternatively, the user can switch from the pixel-oriented glyph to a line graph representation (see Fig. 3.2).

Data View

Each simulation is controlled by 351 variables. To represent the variables, we utilize a well-known visualization technique called parallel coordinates. Although visualizing 1,000 simulations with 351 variables through a parallel coordinates visualization can prove difficult because of a *cluttering* problem, this visualization technique is useful when the data exhibit patterns or underlying structure. Within the parallel coordinates visualization, a color attribute is selected by referencing the political structure of each simulation. Since most variables are mapped by the Dynamic Political Hierarchy (DPH), which characterizes the political structure of a country based on the relationships and strengths of individual political, racial, ideological, and religious groups, the frequency analysis counts the political structure in order to determine the most dominant political identity present in each simulation. The corresponding color attribute is then used to represent the simulation as a line graph.

In the Data view, each line denotes one of the simulations. When the user highlights or selects simulations in the Projection view, the highlighted or selected simulations are emphasized by hiding all other simulations in the parallel coordinates visualization. In addition, the mean variance of the highlighted simulation is displayed with a gradient color mapping method (see Fig. 3.1(a)-bottom). With this feature, the user can intuitively identify the variance over the course of 60 timesteps in each simulation.

Cluster View

Once the analyst has identified and selected interesting simulations in the Projection view, all timesteps in the selected simulations are represented in the Cluster view. Each simulation spans 60 timesteps, and each timestep is mapped to a unique circle in this view (see Fig. 3.1(b)-top). Similar to the Projection view, we apply MDS to reduce dimensionality across all timesteps in the selected simulations. Since each timestep is an individual data element in the Cluster view, similarities among 120 data elements will be computed when two simulations are selected. When multiple simulations are selected, representing all corresponding timesteps in this Projection view makes it difficult for the the user determine which simulation produced each timestep. To avoid this ambiguity, the convex hull is computed to form a boundary around each simulation as shown in Figure 3.1(b)-top. If the user highlights an item (i.e. timestep) by hovering over the item, the convex hull of the corresponding simulation will also be highlighted.

Temporal View

In the Temporal view, all attributes related to each timestep are displayed in a parallel coordinates visualization. As shown in Figure 3.1(b)-bottom, the layout has two components: a variable selector and a parallel coordinates visualization. The variable selector is positioned above the parallel coordinates visualization. Since each small subregion of the parallel coordinates view is mapped directly to a variable, the user can interactively select a variable by simply choosing a subregion. Alternatively, the user can select a variable from the control panel. Based on the selection, the corresponding information is displayed in the parallel coordinates visualization. In this visualization, timesteps are indicated intuitively along x-axis.

As shown in Figure 3.1(b), the color attributes from the Cluster view are used when rendering lines in the parallel coordinates. From this, the user is able to identify what factors influence DPH structures.

Control Panels

Two control panels are designed to allow the user to manage input parameters to the visualization. The first is used to modify the attributes of the visualization. In this panel, the user is able to change variables and modify the color mapping. Since the color mapping is created by referencing the selected variable, whenever the user selects a different variable in the control panel, the corresponding information will be represented to the visualization. The other panel is used for controlling the amount of contribution of a variable in the MDS calculation. Changing the contribution from 100% to 50% indicates that the weight of the selected variable is set to 0.5. When the contribution is diminished to 0%, the selected variable will not be used in computing similarity.

3.4.2 Case Studies

In the following section, we demonstrate the efficacy of MDSViz when deployed for real-world analytical tasks in modeling political systems through case studies developed in collaboration with expert analysts in political science. In both case studies, the MDSViz system was initialized with the VirThai [AHG11] simulation dataset created by our expert analysts.

Identifying Trends in Potential Outcomes

The analysts began by representing the data with pixel-oriented glyphs of the Dominant Identity attribute in the Projection view (see Fig. 3.3) to explore how the simulation runs are clustered and how the clustering correlates to the Dominant Identity attribute. Because the Dominant Identity attribute has a small contribution to the distance function, it can be utilized as a label for each simulation in this context.

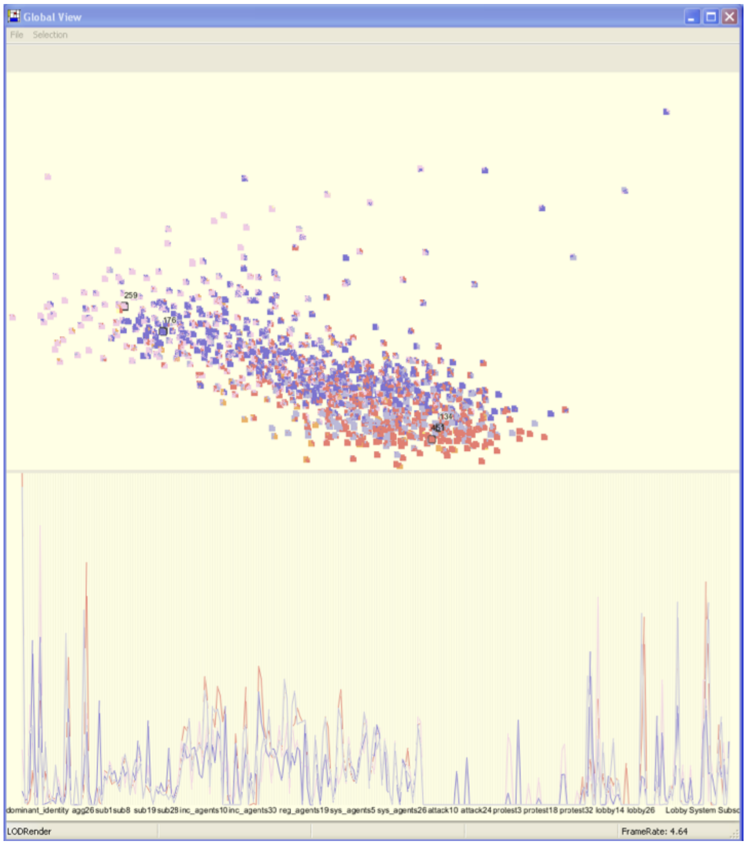
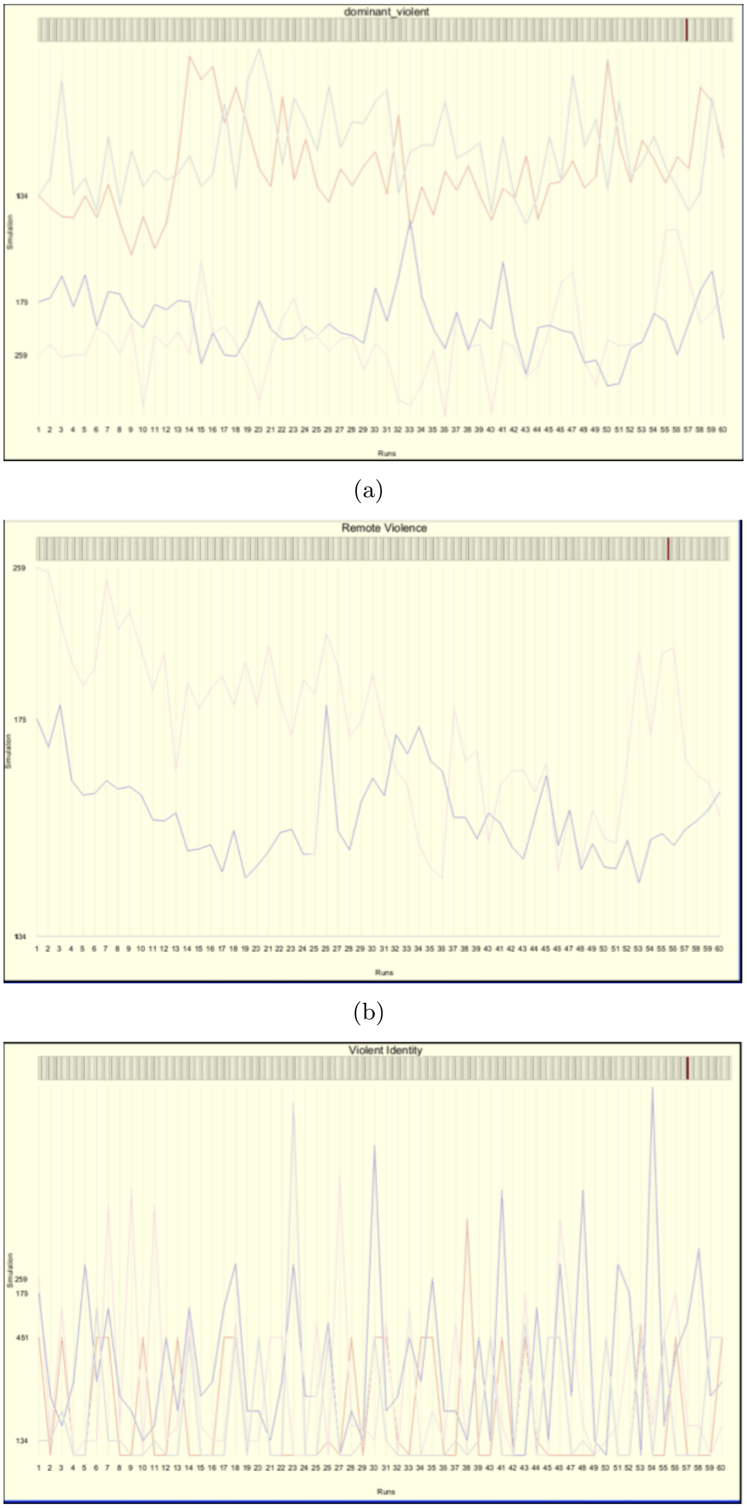

Global View
File
Selection
259
176
134
dominant_identity agg26 sub1 sub8 sub19 sub28 inc_agents10 inc_agents30 reg_agents19 sys_agents5 sys_agents26 attack10 attack24 protest3 protest18 protest32 lobby14 lobby26 Lobby System Subsc
LODRender
FrameRate: 4.64

Figure 3.3: A representation of the data with pixel-oriented glyphs of the Dominant Identity attribute in the Projection view.

In Figure 3.3, analysts observed that runs that more prominently feature Buddhist (red) or Thai Ethnic groups (light purple) as the Dominant Identity are clustered on the right side, whereas runs that more prominently feature the Red Shirts (dark purple) or Yellow Shirts (pink) are clustered on the left. Because the Buddhist/Thai Ethnic clustering is roughly the same size as the Red Shirts/Yellow Shirts clustering, the probability of Thailand's future resembling either of the two outcomes is similar.

The analysts then selected one run from each of the Dominant Identities present in the two clusters to see how the attributes of each run differ. They looked specifically at the Lobby (Fig. 3.4(a)), Protest (Fig. 3.4(b)), and Attack (Fig. 3.4(c)) attributes. As indicated by the graphs shown in Figure 3.4, there are significant differences between the two clusters for the Lobby and Protest attributes, but not the Attack attribute.



(a)

(b)

(c)

Figure 3.4: MDSViz Parallel Coordinates view of individual simulation runs across two clusters for various attributes: (a) Lobby, (b) Protest, and (c) Attack.

While the analysts could not make strong predictions about Thailand's future from this analysis, they hypothesized that one important distinction between the two clusters is that runs in the Buddhist/Thai Ethnic clustering exhibit high levels of legal lobbying and low levels of protest, whereas runs in the Red Shirts/Yellow Shirts clustering exhibit the opposite. To confirm their hypothesis, our collaborators then selected ten runs from each cluster and observed similar patterns for each attribute (see Fig. 3.5)

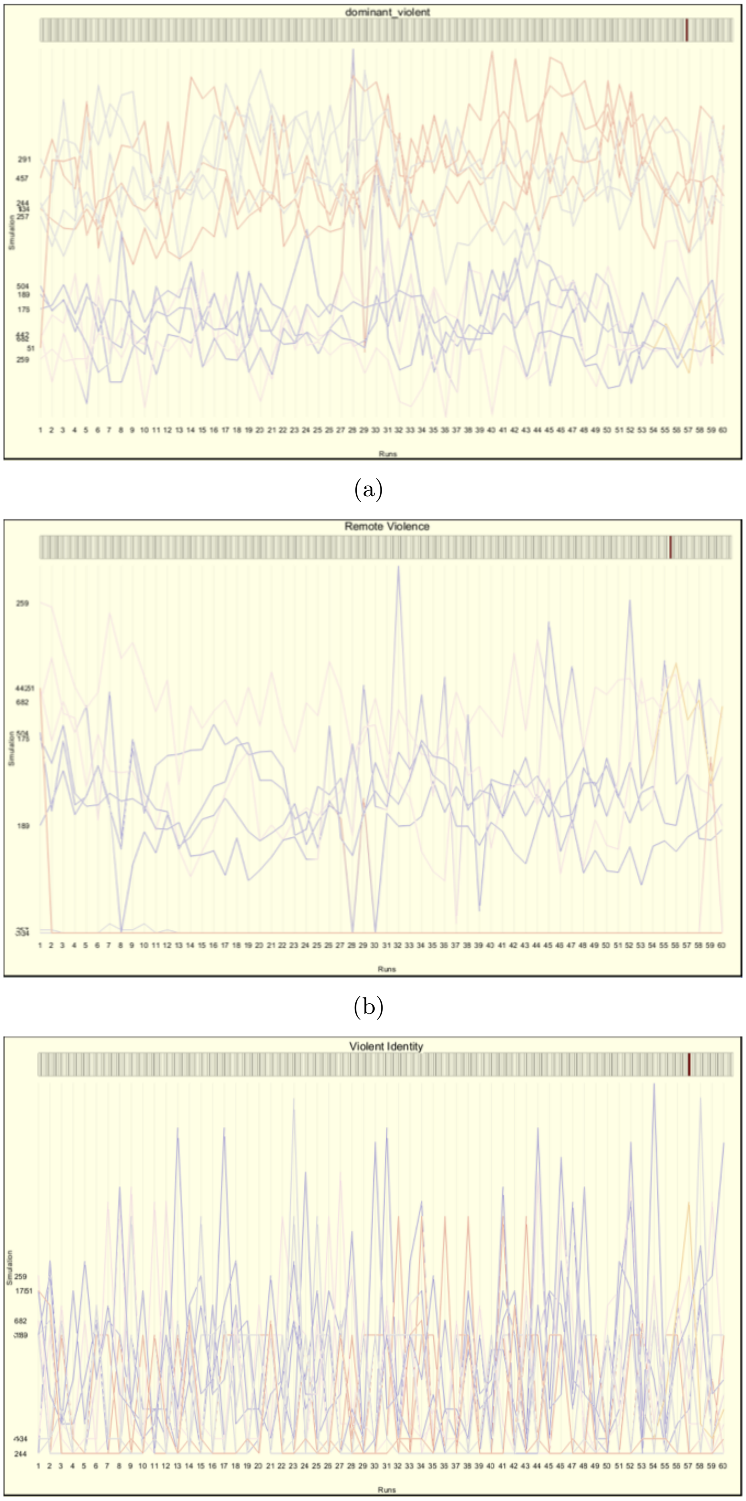
Identifying Unlikely Yet High Impact Outcomes

To analyze unlikely, yet potentially high impact outcomes, the analysts returned to the Projection view (Fig. 3.3) and focused their attention on outliers. Adding two of these outliers to the subset of runs selected in the previous scenario, analysts turned to the Temporal view of the Attack attribute shown in Figure 3.6. In the four runs from the "Identifying Potential Outcomes" usage scenario, there was little noticeable difference between the level of the Attack attribute for these runs, but the additional outlier runs show several spikes of very high levels of Attack relative to the runs from within the clusters.

3.4.3 Qualitative Analysis

Expert analysis revealed that MDSViz was overwhelmingly useful for comparing runs according to their similarity across multiple data dimensions. One analyst reported that "[t]his is the first time we've really been able to group runs according to multidimensional similarity. Until this point we didn't even really have a rudimentary strategy... and even univariate similarity comparisons relied on comparing [a] large number of time series or comparing means." MDSViz has broadened the range of possibilities for analysis by providing a straightforward mechanism for performing multivariate clustering on complex data, as well as greatly reducing the computation time for performing traditional comparisons.

The analysts also reported that the barrier to entry to their analytical process would be greatly reduced by using MDSViz. They report that while identifying and



(a)

(b)

(c)

Figure 3.5: MDSViz Parallel Coordinates view of 10 sample simulation runs across two clusters for various attributes: (a) Lobby, (b) Protest, and (c) Attack.

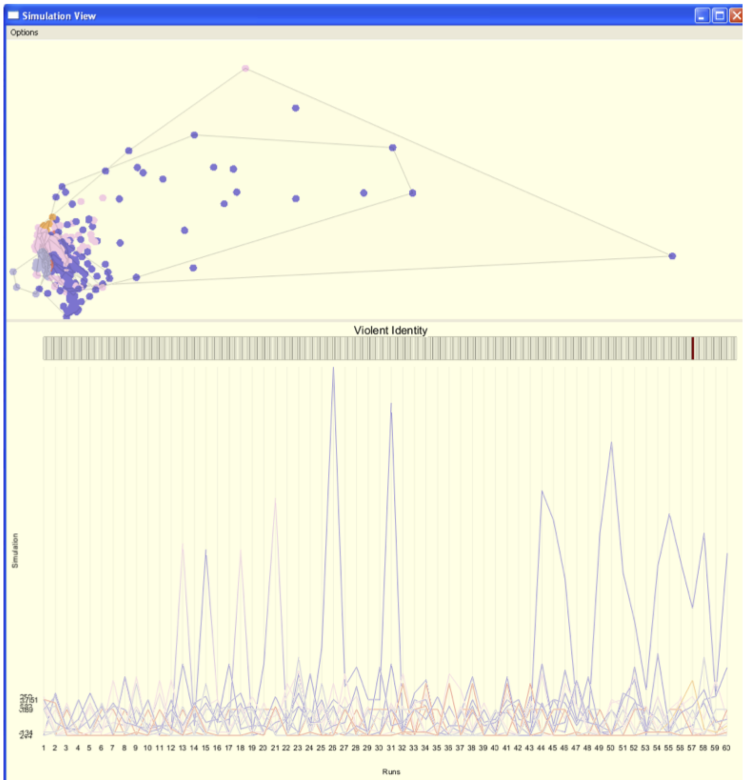


Figure 3.6: A comparison between two outlier runs to more characteristic runs using the Temporal view to explore the Attack attribute.

grouping similar runs and then drilling down into the data to determine what makes those runs unique was possible "based on a high level of familiarity with the model... the process was often opaque." By using MDSViz to identify groups of similar runs and then utilizing the Parallel Coordinates and Time Series views to examine the details of the simulation runs, "a new user is able to explore a data set and find interesting relationships or an experienced user can more quickly understand a new data set."

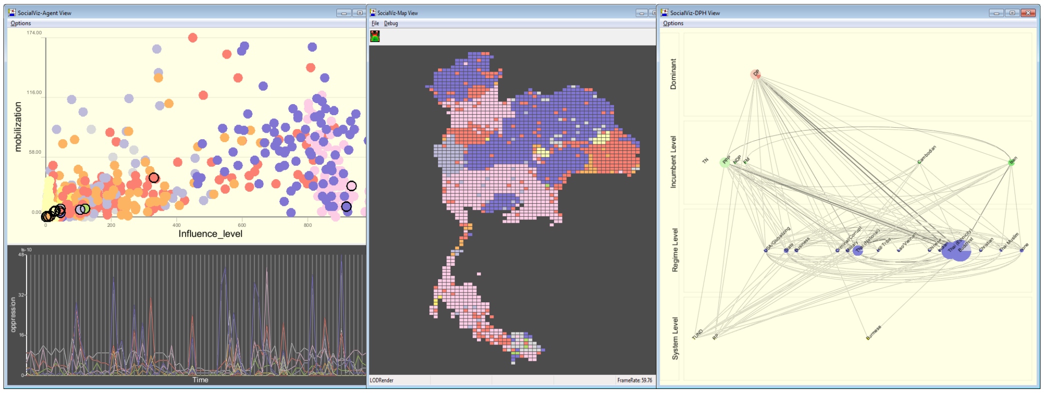
During the evaluation, the experts also identified a few shortcomings of the existing system. In particular, they noted that while MDSViz is a powerful tool for analysis, it is not particularly well-suited for presentation due to the challenges in comparing across multivariate space. They also noted that while they found it useful to be able to alter their distance function by using the control panel to modify the variable weights, computation speed can be problematic. One final drawback of the multidimensional component of MDSViz that the analysts identified is that patterns across many dimensions can tend to cancel each other out. They suggested that in some cases, patterns among fewer variables might be more intuitive and show stronger relationships. In a data set where relationships are generally weak, this technique might help illuminate less obvious patterns.

3.5 Single Run Analysis with SocialViz

In addition to developing intuitions about the dataset as a whole, there are many instances where it is useful to be able to compare individual simulations runs. For example, analysts might want to explore outliers to determine whether or not they represent legitimate but unlikely outcomes, or whether they are simply noise. To compare simulation runs, the values of each variable must be compared independently, leaving the analyst without a holistic overview of the similarities and differences between the compared runs. To tackle this problem, we present SocialViz, an organized mechanism for drilling down into a single run, enabling analysts to explore the behaviors of a single set of conditions, as well as providing a useful tool for debugging the simulation.

3.5.1 System Design

SocialViz enables analysts to perform analyses on the detailed, lower-level information of an individual simulation. In SocialViz, the analyst has access to information about the variables controlling each individual agent at every timestep of the simulation. As shown in Figure 3.7, the four views (Bubble Chart, Temporal, Geographical, and DPH) are designed to support the analysis of correlation, temporal trends, geographical trends, and changes to the Dynamic Political Hierarchy, respectively. All views are coordinated to support a user's interactions between different views.



(a) (b) (c)

Figure 3.7: The four views of the SocialViz system. (a) A Bubble Chart view (top) and a Temporal view (bottom) are designed to support correlation and temporal analysis. (b) A Geospatial view of the overall system including all the agents. (c) The Dynamic Political Hierarchy (DPH) view.

Bubble Chart View

The Bubble Chart view displays the correlation between two intersecting variables. If the two variables maintain a positive correlation, the slope of the pattern of dots will be from lower left to upper right. With this approach, the user is able to examine the actions and interactions of each agent or political group by comparing the correlation between its controlling variables. The analyst can select variables to compare through a control panel. The color attribute is determined by referencing the activated identity within each group, utilizing the same encoding metaphor used in MDSViz.

Temporal View

In the Temporal view, the activities of each agent or political group over time can be represented as line, with the color of the line matched with the color of each group. The line indicates the activities of each group over time. By highlighting the line or time dimension, the corresponding information will be reflected in all other views.

Geospatial View

Location information corresponding to each agent is represented in the Geospatial view. Because the political simulation run in this case was performed on data gathered in Thailand, a geographical map of Thailand is used. Here, each agent is mapped to a region whose color corresponds to the activated identity with each group.

DPH View

The DPH view shows the groups of agents and how their relationships impact the structure and stability of a system. The configuration of the Dynamic Political Hierarchy (DPH) characterizes the political structure of a country based on the relationships and strengths of individual political, racial, ideological, and religious groups [LAGR10]. In this model, each identity is assigned a level in the hierarchy: dominant, incumbent, regime, system, and anti-system. The line between groups represents their relationship, and the thickness of the line indicates how strongly the two groups are connected. By default, all linkages among groups are displayed. Since the DPH View uses a graph drawing approach, commonly known limitations (i.e. *cluttering* and *line crossing*) in graph drawing approaches are also present in the DPH View. To minimize these limitations, a B-spline approach is used to create a curved line. In addition, only highlighted linkages are emphasized when the user interacts with group(s).

Each agent may subscribe to any number of identity groups. At each timestep, an agent will be considered *active* under only one of its subscribed identities. In the DPH View, each identity group is represented as a piechart depicting the number of activated agents and total number of subscribed agents. The darker region in the piechart indicates the proportional percentage to the number of activated agents.

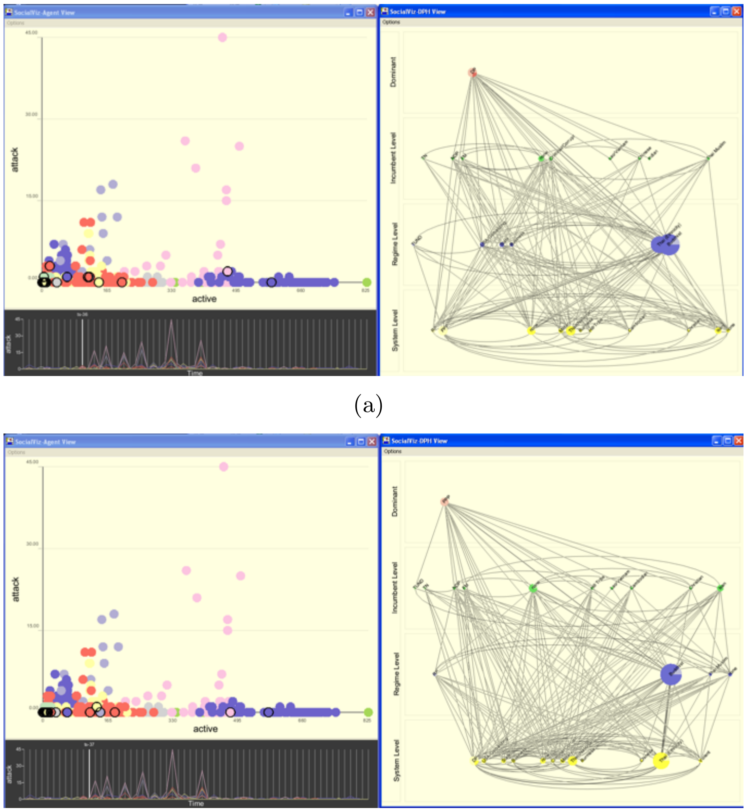
3.5.2 Case Studies

To demonstrate the complementarity of SocialViz to MDSViz, we return to where the previous case study left off. To further explore *why* outliers display such a high level of Attack, the analysts switch to using SocialViz to explore an individual history at an in-depth level. They begin their analysis by using the Temporal and Bubble Chart views to confirm the spikes in Attack that they observed using MDSViz. To understand why these spikes occur, they then examine the DPH view of the timesteps immediately preceding the increase in attacks. In this view, they observe a pattern: in the two timesteps immediately preceding the attacks, there is a shift in the DPH level of the Thai Ethnic identity from the Regime level (Fig. 3.8(a)) to the System level (Fig. 3.8(b)). Additionally, there is also a shift in the Isan group, bringing them from the System level (Fig. 3.8(a)) up to the Incumbent level (Fig. 3.8(b)). Both of these patterns occur immediately before nearly all of the spikes in Attack. From this, the analysts leverage their domain expertise to conclude that, for this run, the high levels of violent attacks probably result from the alienation of the Thai Ethnic group whenever the Red Shirts align themselves closely with the minority Isan ethnicity.

3.5.3 Qualitative Analysis

Analysts agreed that SocialViz provided them with a much more efficient framework for exploring individual trajectories and different variables. One expert stated that to accomplish this task previously they would “have to open the model in PS-I and watch the particular trajectory run or use off-the-shelf software (e.g. Excel, STATA).” The SocialViz system enabled analysts to straightforwardly access and visualize many of the variables at work in their model.

Another analyst noted that “one of the great advantages of SocialViz is its speed, which allows a user to analyze the configuration of a landscape over an entire run very quickly without having to flip back and forth between a series of images. Some of the views, like the sequential DPH visualization, were not available to us



(a)

(b)

Figure 3.8: DPH view of the timesteps immediately preceding an increase in Attacks in a sample outlier run.

at all; [before SocialViz] we only had the ability to generate the visualization from individual timesteps, which is a very time-intensive process." The only drawbacks to the SocialViz system that were noted by the analysts were that not all variables and attributes within the model were available to be viewed, such as the rules and functions operating within the model.

Overall, the analysts reported that MDSViz and SocialViz are invaluable tools that met all of the design considerations that we had collectively identified at the onset of our partnership. They indicated that in many cases, both MDSViz and SocialViz would significantly streamline their analytical processes, support them in identifying interesting patterns, and help them explore how different factors influence political systems.

3.6 Discussion and Future Work

In this section, we discuss the current limitations of our system and identify areas for future research.

Identifying Appropriate Distance Functions

In our current implementation, we use Euclidean distance as a proof-of-concept distance function. However, in this distance measure the attributes are not normalized and thus have uneven weighting depending on the range of values for each individual attribute. While it is possible to compensate for this by adjusting the contribution for an over- or under-represented attribute in the MDSView control panel, it would be much more intuitive if equal contribution values in the control panel equated to equal contribution of attributes in the distance function. Along with this normalization, we would like to explore the utility of offering the user several initial predefined options depending on the data being examined in order to minimize the amount of time and effort required to properly tune the distance function.

Another issue with using Euclidean distance for comparing time series is that it tends to perform poorly when similar features are shifted slightly in time. This weakness is exploited especially by the agent-based simulation data used in this paper, where attributes can vary greatly between consecutive timesteps. Intuitively, the distance between two runs that are identical with the exception of a slight shift in time should be almost nonexistant. However, Euclidean distance has no mechanism to recognize this.

Because of this, we have considered several other distance measures. The first alternative is dynamic time warping (DTW) [BC94], which can be very effective at handling temporal shifting, but is unfortunately computationally intensive. Another alternative is symbolic aggregate approximation (SAX) [LKWL07], which can be used to determine a lower bound on Euclidean distance between two time series in a fraction of the time, and so could be applied to subsequences of the time series to quickly find similar features that are shifted in time.

In our future work, we would like to continue to explore different distance measures to afford analysts better performance and increased control when using these tools. One area of particular interest is the automatic generation of distance functions. One method currently being evaluated in our lab is the effectiveness of using a computational "best guess" approach coupled with an iterative refinement process in partnership with the user to assist the analyst in externalizing their intuitions about the data and thereby computing an appropriate, custom distance function.

Improve MDS Performance

Another area for future improvement is modifying the multidimensional scaling component to enable real-time user interaction with attribute weighting. In particular, we are interested in leveraging the work of Ingram et al. [IMO08] on utilizing the GPU for MDS computation. This research reported speedup factors of 10 to 15 times when using the GPU for their MDS algorithm. As noted in the expert analysis of these systems, the ability to perform multidimensional comparisons between simulations runs provides a previously unexplored opportunity for examining agent-based simulation data in close detail. However, due to the lengthy computation time, analysts are unable to iteratively refine their comparison by modifying the weight distribution across several variables and recomputing the distances between simulations. By refining and speeding up these calculations, we would enable analysts to better explore a range of hypotheses about the factors influencing sociopolitical interactions observed in their simulations.

Integrating MDSViz and SocialViz

Although both systems were very well-received, there has been some discussion about whether the functionality of both MDSViz and SocialViz should be combined into a single system. Because of the memory management issues that arise when working with such large datasets, having both systems combined into a single tool would require the system to dynamically load data, potentially resulting in diminished performance. However, the benefits to a combined system that does not require context-switching on the part of the analyst warrant further investigation into its development. This is especially true when combined with the potential for a dramatic performance increase that could be gained by leveraging the GPU for MDS computation, which would offset some of the dynamic loading bottleneck.

3.7 Summary

Analyzing and interpreting the results of agent-based models is a critical component of current research in social and political science. These simulations can help scientists to better understand the forces at work in social and political systems, which can in turn enable them to better inform decision-makers and international policy. Although there exist robust systems for developing and running these simulations, it is difficult for social scientists to interpret the results of their increasingly complex simulations without appropriate tools.

We have presented two systems specifically designed to support inquiry and inference by social scientists using agent-based simulations to model political phenomena. We designed these systems in collaboration with domain experts to provide interactive exploration and domain-specific data analysis tools. Through evaluation by domain experts, we validated that these systems provide an efficient framework to explore simulation data and confirmed both their novelty and utility. In the following chapter, we will examine some of the characteristics of these systems as well as their contemporaries in analytical and other domains to develop a framework for understanding their strengths and shortcomings.

Chapter 4

Affordances in Human-Computer Collaborative Systems

This chapter is based on the paper:

- Crouser, R. & Chang, R. An affordance-based framework for human computation and human-computer collaboration, *IEEE Transactions on Visualization and Computer Graphics*, 18(12), 2859-2868, 2012.

4.1 Introduction

Thomas and Cook define the field of Visual Analytics as "the science of analytical reasoning facilitated by visual interactive interfaces" [TC05]. By leveraging increasing computational power and the significant bandwidth of human visual processing channels, it strives to facilitate the analytical reasoning process and support the "human capacity to perceive, understand, and reason about complex and dynamic data and situations" [TC05]. As the field matures, it is increasingly imperative to provide mechanisms for approaching analytic tasks whose size and complexity render them intractable without the close coupling and dynamic interplay of both human and machine analysis. Primary goals of this field are to develop tools and methodologies that facilitate human-machine collaborative problem solving, and to understand and maximize the benefits of such a partnership.

Researchers have explored this coupling in many venues: IEEE Conference on Visual Analytics Science and Technology (VAST), IEEE Visualization Conference (Vis), IEEE Information Visualization Conference (InfoVis), ACM Conference on Human Factors in Computing Systems (CHI), ACM Conference on Knowledge Discovery and Data Mining (KDD), ACM Conference in Intelligent User Interfaces (IUI), and more. The study of general human-computer collaboration offers a plethora of examples of successful human/machine teams [CBY10, DWCR11, IV11, KCD+09, KBGE09, LSD+10, MvGW11, SGL09, SSJKF09, TAE+09, ZAM11]. Developments in supervised machine learning in the visualization community present several vetted techniques for human intervention into computationally complex tasks [AWD11, AAR+09, BJVH11, CLKP10, FWG09, GRM10, IMI+10, LLKM10, MW10b, RBBV11]. The emerging field of human computation inverts the traditional paradigm of machines providing computational support for problems that humans find challenging, and demonstrates success using aggregated human processing power facilitated by machines to perform difficult computational tasks such as image labeling [DSG07, HCL+09, VAD04, VAGK+06], annotating audio clips [LVADC07, ME08], and even folding proteins [CKT+10].

While there have been a multitude of promising examples of human-computer collaboration, there exists no common language for describing such partnerships. This begs several questions:

Problem selection

How do we tell *if a problem would benefit from a collaborative technique?* Balancing the cost of building and deploying a collaborative system with the benefits afforded by its use is currently precarious at best. Recent research proposed game-theoretic arguments regarding the kinds of problems that might be effectively crowdsourced [RV12], but these may be difficult to extend to broader human-computer collaborative efforts. Without a framework in which to situate the development of new systems, we rely heavily on researcher intuition and current fieldwide trends to decide which problems to approach using these techniques. This is akin to looking for the sharpest needle in a haystack of needles, and while it has led to many novel approaches to hard problems, it has also led to the investment of significant time and energy into inefficient collaborative solutions for problems that might better have been (or have already been) solved by human or machine techniques alone.

Function allocation

How do we decide *which tasks to delegate to which party, and when?* It has long been stated (even by the author himself) that Fitts' HABA-MABA lists [Fit51] are insufficient and out-of-date. Sheridan notes that function allocation in collaborative systems is far from a perfect science [She00]. Dekker argues that static function allocation consistently misses the mark because humans adapt to their surroundings, including systems with which they work [DW02]. However, the effectiveness of any collaborative system is deeply rooted in its ability to leverage the best that both humans and machine have to offer. Without a language for describing the skills and capacity of the collaborating team, it is difficult to characterize the resources available to the computational process.

Comparative analysis

Finally, *how does one system compare to others trying to solve the same problem?* With no common language or measures by which to describe new systems, we must rely on observed performance alone. This information is often situation dependent. This makes it challenging to reproduce results and to build on previous discoveries, leading to the development of many one-off solutions rather than a cohesive, directed line of research.

To address these questions, we begin by examining the set of attributes that define and distinguish existing techniques in human-computer collaboration. In work presented in IEEE Transactions on Visualization and Computer Graphics [CC12], we surveyed 1,271 papers from many of the top-ranking conferences in Visual Analytics, Human-Computer Interaction, and related areas. From this corpus, we distilled 49 examples that are representative of the study of human-computer collaborative problem-solving, and provide a thorough overview of the current state-of-the-art. This analysis revealed patterns of design hinging on human- and machine-intelligence affordances: properties of the human and machine collaborators that offer opportunities for collaborative action. The results of this analysis provide a common framework for understanding human-computer collaborative systems and indicate unexplored avenues in the study of this area.

4.2 Previous Frameworks

A few of the existing papers surveying work in Human-Computer Collaboration and Human Computation also include discussions of the design dimensions that organize and contextualize their work. In these surveys, the authors provide mechanisms to compare and contrast the systems they review to others along salient dimensions.

Bertini and Lalanne [BL10] survey the intersection of machine learning and visualization, identifying three categories of design hinging on the distribution of labor between human and machine. In *enhanced visualization*, human use of the visualization is the primary data analysis mechanism and automatic computation provides additional support in the form of projection, intelligent data reduction, and pattern disclosure. In *enhanced mining*, data analysis is primarily accomplished by the machine through data mining and visualization provides an advanced interactive interface to help interpret the results through model presentation as well as patterns exploration and filtering. In *integrated visualization and mining*, work is distributed equally between the human and machine collaborators at different stages: white-box integration, where the human and machine cooperate during model-building, and black-box integration, where the human is permitted to modify parameters of the algorithm and immediately visualize the results.

In the area of human computation, Yuen et al. [YCK09] identify three broad categories based on the relative maturity of the system. *Initiatory* systems are the earliest examples of human computation and were generally used to collect commonsense knowledge. *Distributed* systems were the next generation of human computation, aggregating the contributions of Internet users but with limited scalability and without any mechanism to guarantee the accuracy of the information collected. Finally, the authors describe *social game-based* systems, the most recent incarnation of human computation involving enjoyable, scalable and reliable systems for approaching hard AI problems. In a later survey, Quinn and Bederson [QB11] identify six dimensions along which they characterize human computation systems. *Motivation* describes the mechanism for encouraging human participation. *Quality control* indicates whether and how a quality standard is enforced upon the human workers. *Aggregation* refers to the means by which human contributions are collected and used to solve the problem at hand. The remaining dimensions of *human skill*, *process order*, and *task-request cardinality* are self-explanatory.

Each of these frameworks provides critical insight into organizing the systems appearing in the venues they survey. However, because each is specific to a particular subclass of collaborative systems, it is difficult to extend them to a broader class of human-computer collaborative systems. In the following sections, we provide a detailed survey of the literature across many venues, and argue for examining these systems through the lens of *affordances*; that is, what does each collaborator (machine or human) bring to the table in support of the shared goals of the team?

4.3 Framework: allocation and affordances

We now introduce the foundation upon which we will build our framework for describing and understanding human-computer collaborative systems.

4.3.1 Function allocation in human-machine systems

Researchers have sought a systematic approach for the appropriate allocation of functions to humans and machines for decades. In 1950, Fitts published the first formal attempt to characterize functions performed better by machines than humans, and vice versa [Fit51]. For years, this list was regarded as the definitive mantra for function allocation, despite the author's assertion that to use his list to determine function allocation was to lose sight of the most basic tenet of a human-machine collaborative system. As later articulated by Jordan, this underlying foundation is that humans and machines are *complementary*, rather than antithetical [Jor63]. Price [Pri85] further expanded on this idea by arguing that function allocation is perhaps better envisioned as an iterative process rather than a decisive listing, and that there may be more the one optimal allocation for any given problem. Price also notes that human operators require support to perform optimally, and emphasizes the importance of understanding cognitive loading and engagement.

In more recent work, several contemporaries have argued that the notion of function allocation as it was originally conceived no longer makes sense. Sheridan discussed several problems with function allocation which include ever-increasing computing power, complicated problems with optimal allocation differing at each stage, and ill-defined problem spaces [She00]. Dekker and Woods provided a second counterargument to the validity of any Fitts-style HABA-MABA listing in [DW02]. They pointed out a relationship that is often leveraged (though seldom explicitly stated) by the field of Visual Analytics: human-machine collaboration transforms human practice and forces people to adapt their skills and analytic practices. They advocated for a shift in attention, moving away from allocation of tasks to a focus centered on how to design for harmonious human-machine cooperation. That is, how do we get humans and machines to play nicely, and work effectively?

4.3.2 Affordances

In 1977, American psychologist J.J. Gibson stated his theory that an organism and its environment complement each other [Gib77], which is much in alignment with the work by Jordan cited in the previous section. In this work, Gibson coined the term *affordances*, defining them as the opportunities for action provided to an organism by an object or environment. Norman later appropriated this term as it applies to design and the field of Human-Computer Interaction, redefining it slightly to refer only to the action possibilities that are readily perceivable by a human operator [Nor02]. This definition shifts the concept of affordance toward *relational* rather than subjective or intrinsic; that is, an affordance exists *between* an actor and the object or environment, not existing separate from that relationship.

In the case of human-computer collaboration, we argue that there exist affordances in both directions. Both human and machine bring to the partnership opportunities for action, and each must be able to perceive and access these opportunities in order for them to be effectively leveraged. These affordances define the interaction possibilities of the team, and determine the degree to which each party's skills can be utilized during collaborative problem-solving. In the next sections, we will survey the existing literature through the lens of affordances, providing a common framework for understanding and comparing research in the areas of human-computer collaboration, human intervention, and human computation. The affordances we identify are by no means an exhaustive list; they simply represent the patterns of design that we have seen in the existing literature of an emerging area. Please note that while examples will generally be given under the heading of a single affordance, systems mentioned may utilize multiple affordances (both human and machine) at the same time. For a complete listing of the affordances identified in all systems surveyed, please see Table 4.1 in the Appendix of this work. In Section 4.6, we present case studies of specific systems to discuss the costs and benefits of leveraging multiple affordances.

4.4 Human Affordances

The human-computation and human-computer collaborative systems we have reviewed leverage a variety of skills and abilities afforded by the human participants. In this section, we will offer a brief definition of each of the affordances we have observed in the literature, discuss the utility of these affordances as articulated in the work reviewed and offer an overview of the application of each affordance.

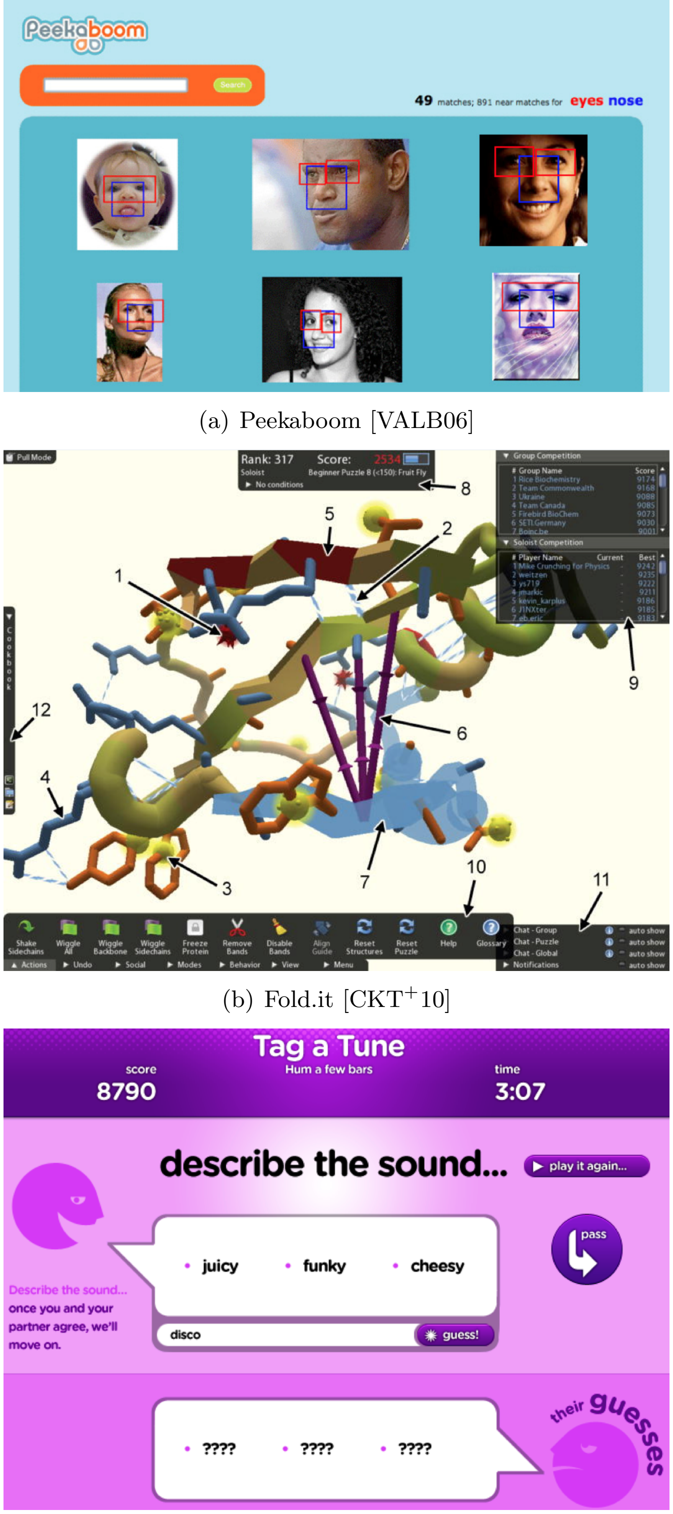
4.4.1 Visual perception

Of the human affordances we will discuss, perhaps the most salient to the study of Visual Analytics is *visual perception*[1]. In [Shn96], Shneiderman comments on humans' capacity for visual processing:

> [T]he bandwidth of information presentation is potentially higher in the visual domain than for media reaching any of the other senses. Humans have remarkable perceptual abilities. . . Users can scan, recognize, and recall images rapidly, and can detect changes in size, shape, color, movement, or texture. They can point to a single pixel, even in a megapixel display, and can drag one object to another to perform an action.

Given its direct applicability, it is perhaps unsurprising that we have seen a plethora of work in Visual Analytics and HCI leveraging human visual processing. For example, human visual perceptive abilities are utilized by Peekaboom [VALB06] to augment image labels on the web (see Fig. 4.1a). For some tasks such as image labeling [DSG07, HCL$^+$09, RTMF08, ST08, VAD04, VAGK$^+$06], visual search [BRB$^+$09, BJJ$^+$10], and query validation [MCQG09, YKG10], the systems presented rely heavily on the users' visual perceptive abilities, with the machine serving only as a facilitator between the human and the data. For other tasks such as exploring high-dimensional datasets [TAE$^+$09, ZAM11], classification [AAR$^+$09, MW10b], and dimension reduction [FWG09, IMI$^+$10], machine affordances (which will be discussed

[1] For more on visual perception, see Gibson [Gib86].



Rank: 317 Score: 2534
Soloist Beginner Puzzle 8 (<150): Fruit Fly
▶ No conditions

▼ Group Competition

#	Group Name	Score
1	Rice Biochemistry	9174
2	Team Commonwealth	9168
3	Ukraine	9088
4	Team Canada	9085
5	Firebird BioChem	9073
6	SETI.Germany	9030
7	Boinc.be	9001

▼ Soloist Competition

#	Player Name	Current	Best
1	Mike Crunching for Physics	-	9242
2	weitzen	-	9235
3	ys719	-	9222
4	jmarkic	-	9211
5	kevin_karplus	-	9186
6	JINXter	-	9185
7	eb.eric	-	9183

- juicy
- funky
- cheesy

disco

Describe the sound... once you and your partner agree, we'll move on.

- ????
- ????
- ????

(a) Peekaboom [VALB06]

(b) Fold.it [CKT+10]

(c) TagATune [LVADC07]

Figure 4.1: Systems leveraging human affordances: (a) *Visual perception*, (b) *Visuospatial thinking*, and (c) *Audiolinguistic ability*.

at length in Section 4.5) are combined with human visual processing to achieve superior results.

4.4.2 Visuospatial thinking

A level deeper than basic visual processing such as image recognition, another skill afforded by human collaborators is *visuospatial thinking*[2], or our ability to visualize and reason about the spatial relationships of objects in an image. These abilities are strongly informed by our experiences in the physical world, which shape our understanding and are intrinsic to our everyday lives. We are able to visualize complex spatial relationships and tune this attention to accomplish specific goals. In an article on the significance of visuospatial representation in human cognition [SM05], Tversky notes:

> For human cognition, [entities] are located in space with respect to a reference frame or reference objects that vary with the role of the space in thought or behavior. Which things, which references, which perspective depend on the function of those entities in context. . . These mental spaces do not seem to be simple internalizations of external spaces like images; rather, they are selective reconstructions, designed for certain ends.

We have seen evidence that progress can be made on computationally intractable problems through the application of human visuospatial thinking. For example, the Fold.it project (see Fig. 4.1b) has demonstrated remarkable success at protein folding [CKT$^+$10], a problem known to be NP-complete [BL98] using purely computational means.

4.4.3 Audiolinguistic ability

Another affordance presented by the human user is *audiolinguistic ability*; that is, our ability to process sound[3] and language[4]. Although separate from the visual

[2] For more on visuospatial thinking, see Shah and Miyake [SM05].

[3] For more on psychoacoustics, see Fastl and Zwicker [FZ07].

[4] For more on language, see Vygotsky [Vyg62].

affordances generally leveraged in Visual Analytics systems, we suggest that the interplay between visual and nonvisual human faculties is equally important in supporting analytical reasoning. In [TC05], Thomas and Cook state:

> We perceive the repercussions of our actions, which also recalibrates perception, ensuring that vision, hearing, and touch maintain their agreement with each other. If we are to build richly interactive environments that aid cognitive processing, we must understand not only the levels of perception and cognition but also the framework that ties them together in a dynamic loop of enactive, or action-driven, cognition that is the cognitive architecture of human-information processing.

The literature contains several examples of systems leveraging this affordance. The well-known reCAPTCHA [VAMM+08] system uses human linguistic ability augment computer vision in an effort to fully digitize the world's libraries. In MonoTrans2 [HBRK11], it is used to improve automated translation results using monolingual translators. In TagATune [LVADC07], human audio processing ability is leveraged to generate descriptive tags for music clips (see Fig. 4.1c). We have also surveyed examples utilizing human audio linguistic ability for audio annotation [BOTL09, LVADC07, ME08], transcription [CLZ11], and even crowdsourced word processing [BLM+10].

4.4.4 Sociocultural awareness

In addition to physical senses, human collaborators also afford attributes such as *sociocultural awareness*, which refers to an individual's understanding of their actions in relation to others and to the social, cultural, and historical context in which they are carried out. Researchers in the area of embodied interaction have long advocated for design that acknowledges the importance of this relationship. In [Dou04], Dourish notes:

> [O]ur daily experience is social as well as physical. We interact daily with other people, and we live in a world that is socially constructed.

> Elements of our daily experience – *family, technology, highway, invention, child, store, politician* – gain their meaning from the network of social interactions in which they figure. So, the social and the physical are intertwined and inescapable aspects of our everyday experiences.

We argue that this can be viewed as an affordance, not just a complicating factor. For example, in Mars Escape [COB10], human participants partner with a virtual robot to complete collaborative tasks to build robust social training datasets for human-robot interaction research. This affordance is integral to the construction of commonsense knowledge databases [KLC$^+$09, LST07, VAKB06], and has been leveraged in domains such as stress relief [CCXC09] and providing social scripts to support children with autism [BKAA11].

4.4.5 Creativity

Another important affordance of human collaborators is *creativity*[5]. As noted by Fitts [Fit51], Dekker [DW02] and many others, humans are capable of incredible creativity, generating spontaneous arrhythmic approaches to problems that may be difficult or impossible to simulate. In [Run07], American psychologist Mark Runco posits:

> It may be that creativity plays a role in all that is human. This surely sounds like a grand claim, but consider how frequently we use language or are faced with a problem. Think also how often problems are subtle and ill-defined. . . [C]reativity plays a role in each of our lives, and it does so very frequently.

We have seen human creativity leveraged to great success in both physical and conceptual design. For example, Yu and Nickerson [YN11] use human creativity to crowdsource design sketches via a human genetic algorithm, and Tanaka et al. [TSK11] use sequential application of crowds to produce creative solutions for

[5] For more on creativity, see Amabile [Ama96].

social problems. Creativity has also been used to augment automated systems and find hidden outliers [LLKM10].

4.4.6 Domain knowledge

The final example of human affordance that we have seen in the literature is straightforward, but worthy of inclusion nonetheless. This is the affordance of *domain knowledge*. In their 2009 article on Knowledge-Assisted Visualization [CEH$^+$09], Chen et al. argue:

> [T]he knowledge of the user is an indispensable part of visualization. For instance, the user may assign specific colors to different objects in visualization according to certain domain knowledge. The user may choose certain viewing positions because the visualization results can reveal more meaningful information or a more problematic scenario that requires further investigation.

Often, this domain knowledge can be difficult or impossible to embed fully into the system itself, or it may be too time-consuming to generate a complete model of the domain. Instead, we can leverage the experience of the human analyst as part of the collaborative process. For example, we have seen domain expertise leveraged to help diagnose network faults [LLKM10], classify MRI data [BJVH11], perform domain-specific data transformations [KPHH11], and infer trends about a specific geographic region [AAR$^+$09].

4.5 Machine Affordances

For over two decades, the HCI community has been engaged in conversation about affordances in technology [Gav91]. While much of the focus has centered on designing interfaces that are intuitive to the user, we would like to take the liberty of broadening the definition of affordances to include more than just design elements. In this section, we survey the literature with an eye toward the conceptual affordances of machine collaborators and discuss how they come into play in human-computer collaboration.

4.5.1 Large-scale data manipulation

As predicted by Moore's Law [M^{+}98], computational power has steadily doubled every two years for the past five and a half decades. Because of this incredible increase in processing ability, machine collaborators afford *large-scale data manipulation* at speeds and scales Fitts never could have imagined. In Visual Analytics, this computational ability has been leveraged to help analysts navigate massive datasets across many domains. For example, RP Explorer uses random projections to approximate the results of projection pursuit to find class-separating views in high-dimensional space where traditional projection pursuit can fail to converge [AWD11]. In ParallelTopics (see Fig. 4.2(a)), computational methods for manipulating large datasets have been used to help users navigate and make sense of massive text corpora [DWCR11]. It has also been utilized to refine classification models and perform dimension reduction [CLKP10, GRM10, MW10b], interactively cluster data [AAR^{+}09], and automatically extract transfer functions from user-selected data [RBBV11]. It has been used to suggest informative data views [ZAM11], and even to help users externalize and understand their own insight generation process [CBY10, KCD^{+}09, KBGE09, LSD^{+}10, SGL09].

4.5.2 Collecting and storing large amounts of data

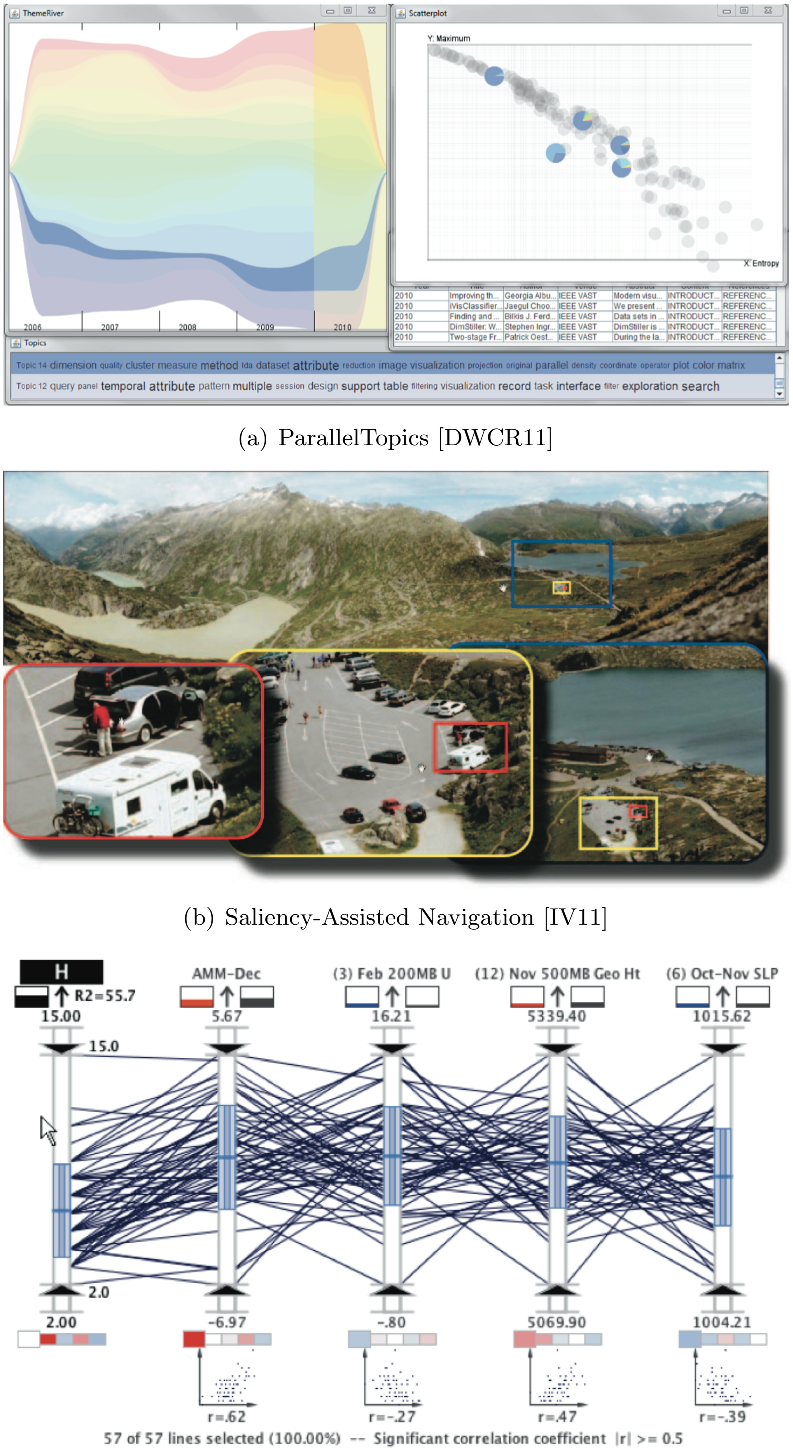
In addition to being able to manipulate large amounts of data at incredible speed, machine collaborators are also able to *efficiently aggregate and store data* for later use. This affordance has been used to support human users in many areas where the data is being generated in large quantities and from multiple sources simultaneously. For example, systems like Verbosity [VAKB06] and others [KLC^{+}09, LST07] aggregate and store information generated by human users to create commonsense knowledge repositories. It is also used in the collection of behavioral scripts for autism treatment [BKAA11] and human-robot interaction [COB10], as well as collecting tags for music and image annotation [BOTL09, BJVH11, DSG07, ME08, RTMF08, ST08, VAD04, VAGK+06]. In a world that is growing ever more *big data*-centric, storage capacity and efficient retrieval are critical advantages afforded by machine collaborators.

4.5.3 Efficient data movement

Thanks to developments in data storage, the advent of fast and reliable networking techniques, and the rapid development of an always-connected society, data has been freed from its historic ties to a geographic location and machine collaborators afford very *efficient data movement*. This implies that data can be collaboratively accessed and manipulated by entities asynchronous in both time and space, with machines affording the efficient transfer of data to the right place at the right time. For example, VizWiz [BJJ+10] leverages efficient data movement to connect visually-impaired users to sighted collaborators to get near real-time answers to visual search questions. This affordance is critical in facilitating distributed collaboration [BLM+10, CAB+11, CCXC09, HCL+09, TSK11, VALB06, YKG10], as well as access to distributed information [HCL+09, LVADC07, MCQG09, VALB06]. Efficient data movement techniques also facilitate rapid access to data that is too large to fit in memory. This has been leveraged to augment human visual processing using saliency modulation [IV11] (see Fig. 4.2(b)), as well as facilitate access to other datasets to numerous to list.

4.5.4 Bias-free analysis

In contrast to the human affordance of sociocultural understanding, machines afford the opportunity for *bias-free analysis*. That is, apart from human bias introduced during the programming of the system, machines are able to operate and report on numerically or computationally significant information without experiential or sociocultural influence. In Visual Analytics, we have seen this affordance leveraged to help analysts direct their attention for natural disaster prediction [SSJKF09] (see Fig. 4.2(c)) as well as propose candidate visualizations for exploring high-



Year	Title	Author	Venue	Abstract	Content	References
2010	Improving th...	Georgia Albu...	IEEE VAST	Modern visu...	INTRODUCT...	REFERENC...
2010	iVisClassifier...	Jaegul Choo...	IEEE VAST	We present ...	INTRODUCT...	REFERENC...
2010	Finding and ...	Bilkis J. Ferd...	IEEE VAST	Data sets in ...	INTRODUCT...	REFERENC...
2010	DimStiller: W...	Stephen Ingr...	IEEE VAST	DimStiller is ...	INTRODUCT...	REFERENC...
2010	Two-stage Fr...	Patrick Oest...	IEEE VAST	During the la...	INTRODUCT...	REFERENC...

Topic 14 dimension quality cluster measure method lda dataset attribute reduction image visualization projection original parallel density coordinate operator plot color matrix

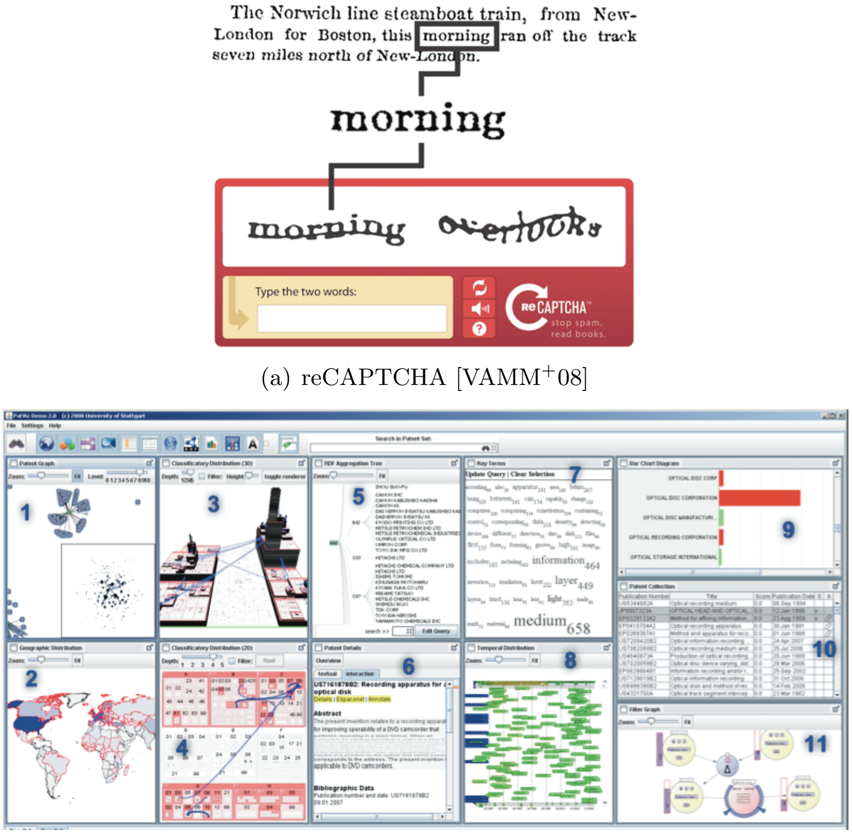
Topic 12 query panel temporal attribute pattern multiple session design support table filtering visualization record task interface filter exploration search

(a) ParallelTopics [DWCR11]

(b) Saliency-Assisted Navigation [IV11]

(c) MDX [SSJKF09]

Figure 4.2: Systems leveraging machine affordances: (a) *Large-scale data manipulation*, (b) *Efficient data movement*, and (c) *Bias-free analysis*.



The Norwich line steamboat train, from New-London for Boston, this morning ran off the track seven miles north of New-London.

(a) reCAPTCHA [VAMM+08]

(b) PatViz [KBGE09]

Figure 4.3: Systems leveraging multiple affordances: (a) reCAPTCHA [VAMM+08] leverages human *visual perception* and *audiolinguistic ability* with machine *storage* and *efficient data movement* to digitize the world's libraries. (b) PatViz [KBGE09] leverages human *visual perception*, *visuospatial ability*, *audiolinguistic ability* and *domain knowledge* with machine *computation*, *storage* and *efficient data movement*.

dimensional data [TAE+09]. It has also been used to help analysts see dissimilarity to existing datapoints [MvGW11], where confirmation or other bias may come into play.

4.6 Multiple Affordances: Case Studies

As stated in the introduction, while we have generally listed examples under a single main affordance, systems may utilize multiple affordances (both human and machine) in pursuit of a common goal. In this section, we analyze a few systems leveraging multiple affordances and discuss the impact of each set of design elements.

4.6.1 reCAPTCHA

reCAPTCHA, first introduced by Luis von Ahn et al. in [VAMM$^+$08] and later acquired by Google, is a web security mechanism that harnesses the effort of humans performing CAPTCHAs along with optical character recognition (OCR) to collaboratively digitize the world's text corpora (see Fig. 4.3(a)). In the first year reCAPTCHA was made available for public use, over 440 million suspicious words were correctly deciphered resulting in over 17,600 successfully transcribed books [VAMM$^+$08]. As of this writing, the system is used over 100 million times every day with an overall success rate of 96.1%, and is currently being utilized to digitize the New York Times archive as well as Google Books. Such widespread adoption and remarkable accuracy mark reCAPTCHA as one of the most widely successful human-computer collaborative initiatives to date.

We posit that the success of the reCAPTCHA system is due in part to its effective combination of human and machine affordances. After performing an initial automated recognition of a document (*computation*), suspicious or unrecognizable words are identified and transmitted (*efficient data movement*) to a collection of human collaborators for evaluation (*visual perception*) and subsequent transcription (*linguistic ability*). Through this division of labor, each party receives manageable tasks to perform according to their skills, and each set of affordances can be leveraged without overloading the collaborator.

4.6.2 PatViz

PatViz [KBGE09] is a Visual Analytics system for the interactive analysis of patent information (see Fig. 4.3(b)). PatViz utilizes a flexible coordinated multiple views (CMV) to support the construction of complex queries and the interactive exploration of patent result sets.

Analysis of patent information is a complex task involving the synthesis of many data dimensions. Because of this, PatViz leverages a multitude of human and machine affordances in an effort to provide intuitive views for various data types: *visual* *perception* for the inspection of image data contained in patent documents, *visuospatial ability* for analyzing the relationships between various patents, *audiolinguistic ability* for evaluating terminology, and *domain knowledge* for understanding the relevance of the patent to its application, as well as with machine *computation* for generating data views on the fly, *storage* for aggregating the analysts' activity, and *efficient data movement* to provide the analyst with the appropriate information on-demand.

However, in the case of leveraging affordances, more is not always better. As articulated in the discussion of the results [KBGE09]:

> One frequently expressed comment indicated that most of the patent experts never worked with a system providing interlinked and interactive visual interfaces. While this was also one of the systems properties that was most appreciated by the users, it became clear that such features are **very difficult to use** without any training.

While the machine collaborator offers many opportunities for the human to utilize many different analytical skills, it falls short in effectively leveraging these affordances by leaving the decision of when and how to select views wholly at the discretion of the human. Because so many different affordances are being leveraged, it is difficult for the human collaborators to organize their strategy in approaching the analysis, resulting in an interface that "is difficult to comprehend. . . without previous instruction" [KBGE09].

4.7 Suggested extensions

The scope of this framework is limited to the affordances we have identified in the existing literature on human-computer collaboration and human computation; it is far from an exhaustive list of the possible affordances that exist between human and machine. We would like posit a few un- or under-explored affordances and suggest scenarios in which these affordances might prove useful.

Human Adaptability: One of the most important components of the human analytic process is the ability to take multiple perspectives on a problem, and adapt hypotheses and mental models in the wake of new information. This adaptability is critical to the successful generation of insight about large datasets. However, most work in this area has centered around *supporting* the adapting user, rather than explicitly leveraging this.

Consider the hypothetical collaborative system leveraging human adaptability suggested by Thomas and Cook [TC05]: as human collaborators are exploring a dataset, the system observes patterns in provenance to try to detect when an analyst has gotten “stuck” in a redundant or potentially fruitless analytical path. When this happens, the system suggests an alternative perspective or avenue for exploration. This encourages the analyst to form new hypotheses or adopt new methods of inquiry, ensuring that the analysis does not become entrenched in a local minimum.

Machine Sensing: With new developments in hardware technology rapidly becoming more readily available, there is the potential for significant advances in the kinds of sensory information that machines can make available. However, to our knowledge, this affordance has not yet been considered as part of a collaborative system.

We see potential for the utility of sensing technology as part of a human-computer collaborative team in two areas. First, sensing technology could be used to make the human collaborator aware of extrasensory information about the environment around them. Second, it could be used to respond to changes in the human collaborator themselves; for example, adapting to the user’s mental state using brain sensing technology to improve the working environment.

These represent just a brief brainstorming of potential additions to the list of affordances we have observed in the literature to date, and we hope that these ideas will inspire intellectual discourse and encourage further inquiry.

4.8 Discussion

We close this paper with a discussion of the utility of this framework for addressing critical need in the area of human-machine collaboration, as well as its shortcomings and areas for future work.

4.8.1 Utility of an affordance-based framework

We claim that with the development of an affordance-based language for describing human-computer collaborative systems, we are indeed in a better position than when we first began. To validate this claim, let us return to the three questions posed in the introduction of this paper:

How do we tell if a problem would benefit from a collaborative technique? We argue that the set of problems warranting a collaborative technique is equivalent to the set problems where there is an opportunity to effectively leverage affordances on both sides of the partnership in pursuit of the solution. By framing potential collaboration in terms of the affordances at our disposal, we can then consider which of these affordances could be used to approach a problem and construct a solution.

How do we decide which tasks to delegate to which party, and when? In adopting this language, we are deliberately moving away from terminology that encourages us to speak in terms of deficiencies; that is, we need the human because computers are bad at X, etc. Instead of deciding who gets (stuck with) which task, we begin to reason about who can contribute to the collective goal at each stage. The answer may not be *only the human*, or *only the machine*, but could in fact be *both.* By designing such that all parties are aware of the affordances made available to them by their collaborators, we encourage the development of more flexible procedures for collective problem-solving.

How does one system compare to others trying to solve the same problem? Of the contributions made by this framework, we believe that providing a common language for discussing human-computer collaborative systems is its greatest strength. We are able to talk about which affordances are being leveraged, and use these to compare and contrast between systems. We may also be able to make hypotheses about how these choices of affordances influence the resulting solutions by comparing performance measures. However, this language does not yet enable a robust, theoretical comparison. To achieve this, we must first build our understanding of the mechanisms underlying these affordances and their associated costs.

4.8.2 Complexity measures for Visual Analytics

While we believe that this framework provides an important foundation for developing a common language, it is only the first of many steps toward a rich vocabulary for describing human-computer collaborative systems. Consider for example the plethora of human computation systems for image labeling that we have reviewed in this work: the ESP Game [VAD04], Ka-captcha [DSG07], KissKissBan [HCL+09], LabelMe [RTMF08] and Phetch [VAGK+06]. Each system leverages the visual perception and linguistic abilities of the human users, and the aggregative capacity of the machine. Given that these systems are all addressing very similar problems using a similar approach, how do they compare to one another? We argue that is it critical to develop a common language not just for describing which affordances are being leveraged, but *how much* and *how well*.

The National Science Foundation CISE directorate has called for the development of theoretical measures for systems involving human computation, calling this one of the five most important questions facing computer science today [Win08]. This need was reiterated at the CHI2011 workshop on Crowdsourcing and Human Computation [Kul11]. Can we begin to describe the complexity of human-computer collaborative systems with a robust language parallel to describing the complexity of an algorithmic system?

Researchers in the field of Artificial Intelligence have begun to imagine the concept of complexity measures for systems involving human contribution. Shahaf and Amir define a *Human-Assisted Turing Machine* using the human as an oracle with known complexity [**?**]. In this work, they demonstrate that much of the standard theoretical language holds true, including *algorithmic complexity, problem complexity, complexity classes* and more. However, they also raise several questions that remain unanswered:

- First, **what is the best way to measure human work**? In terms of *human time, space,* or *utility*? Should we consider the *input size*, that is, how much data does the human need to process? Or to compensate for compression, should we be measuring *information density* instead?

- Second, **how can we assess this human work in practice**? Through *empirical evaluation* of a sample population's performance on a given task, we can begin to understand how the average human performs, but this information is task-specific. Perhaps more broadly applicable would be to develop a set of *canonical actions* that humans can perform with known complexity, but compiling this list is nontrivial.

- Finally, **how do we account for individual differences in human operators**? Perhaps the problem under consideration utilizes skills or knowledge not common to every user (such as bilingual translation). In this case, a general model of humans is insufficient; instead, we need to understand the complexity of the individual candidate. This requires the development of algorithmic systems that are to be able to effectively and efficiently utilize the affordances provided by the humans available to them, rather than only the optimal human collaborator under perfect conditions.

These areas provide many rich opportunities for collaboration with our colleagues in theoretical computer science, as well as in psychology and neuroscience. By engaging in the interdisciplinary pursuit of answers around human affordances, we hope to construct a more complete picture of insight generation, the mechanisms of human understanding, and the the analytic process as a whole.

While we have concentrated our efforts on systems explicitly labeled as human-computer collaboration, mixed-initiative, or human computation, we posit that the framework presented here will benefit the study of a broader class of systems involving both human and machine computation as a whole. While there has been remarkable progress in the development of novel solutions to support analytic processes, we have not yet fully realized our potential as a systematic science that builds and organizes knowledge in the form of testable theories and predictions. In presenting a preliminary framework for describing and comparing systems involving human and machine collaborators, we lay the foundation for a more rigorous analysis of the tools and approaches presented by our field, thereby enabling the construction of an increasingly robust understanding of analytical reasoning and how to best support insight generation. In the following chapter, we present a theoretical model for evaluating the complexity of systems involving human computation, and demonstrate its utility in comparing and assessing human-computer collaborative systems from the literature.

	Human Affordances						Machine Affordances			
	Visual	Spatial	Aud/Ling.	Creativity	Social	Domain	Comp.	Storage	Moving	Bias-Free
PatViz [KBGE09]	✓	✓	✓			✓	✓	✓	✓	
CrowdSearch [YKG10]	✓						✓		✓	
ParallelTopics [DWCR11]	✓		✓				✓			
Dissimilarity [MvGW11]	✓					✓	✓			✓
VH+ML [FWG09]	✓						✓			
Implicit tagging [ST08]	✓						✓			
reCAPTCHA [VAMM+08]	✓		✓					✓	✓	
VizWiz [BJJ+10]	✓				✓				✓	
Phetch [VAGK+06]	✓		✓	✓	✓			✓		
ESP Game [VAD04]	✓		✓	✓				✓		
KissKissBan [HCL+09]	✓		✓	✓				✓		
LabelMe [RTMF08]	✓		✓	✓				✓		
Ka-captcha [DSG07]	✓		✓	✓				✓		
PeekABoom [VALB06]	✓							✓		
MRI [BJVH11]		✓				✓	✓	✓	✓	
iView [ZAM11]		✓				✓	✓		✓	
iVisClassifier [CLKP10]		✓				✓	✓		✓	
Saliency [IV11]		✓				✓	✓		✓	
RP Explorer [AWD11]		✓					✓		✓	✓
DimStiller [IMI+10]		✓				✓	✓	✓		
WireVis [LSD+10]		✓				✓	✓	✓		
Action trails [SGL09]		✓				✓	✓	✓		
NetClinic [LLKM10]		✓				✓	✓			
Trajectories [AAR+09]		✓				✓	✓			
Risk assessment [MW10b]		✓				✓	✓			
Automatic transfer functions [RBBV11]		✓				✓	✓			✓
MDX [SSJKF09]		✓				✓	✓			✓
Automated+viz [TAE+09]		✓				✓	✓			✓
CzSaw [KCD+09]		✓					✓	✓		
Fold.it [CKT+10]		✓					✓			
HRI scripts [COB10]		✓			✓			✓		
Animated agents for VR [RJ99]		✓			✓			✓		
VA Model-learning [GRM10]		✓				✓				
EyeSpy [BRB+09]		✓						✓		
MonoTrans2 [HBRK11]			✓				✓	✓		
CastingWords [CLZ11]			✓				✓			
Click2Annotate [CBY10]						✓	✓	✓		
Wrangler [KPHH11]						✓	✓			
Soylent [BLM+10]			✓						✓	
Crowdsourced solutions [TSK11]				✓					✓	
Crowdsourced design [YN11]				✓					✓	
Stress OutSourced [CCXC09]					✓				✓	
PageHunt [MCQG09]			✓	✓	✓			✓		
Herd It [BOTL09]			✓	✓				✓		
TagATune [LVADC07]			✓	✓				✓		
MajorMiner [ME08]			✓	✓				✓		
Autism scripts [BKAA11]					✓			✓		
Social Games [KLC+09]					✓			✓		
Common Consensus [LST07]					✓			✓		
Verbosity [VAKB06]					✓			✓		

Table 4.1: A table of all surveyed human-computer collaborative systems and the affordances they leverage. The human affordances listed are *visual perception, visuospatial ability, audioliguistic ability, creativity, sociocultural awareness,* and *domain knowledge*. The machine affordances listed are *large-scale data manipulation, collecting and storing large amounts of data, efficient data movement,* and *bias-free analysis*.

Chapter 5

Formalizing Human Computation with an Oracle Model

This chapter is based on the following manuscripts:

- Crouser, R. J., Ottley, A., & Chang, R. (2014, to Appear). Balancing human and machine contributions in human computation systems. In P. Michelucci (Ed.), Handbook of Human Computation. New York, NY: Springer.
- Crouser, R. J., Hescott, B., Glaser, M., & Chang, R. Theoretical Bounds for crowdsourced image labeling under a human oracle model. AAAI Conference on Human Computation & Crowdsourcing. In Submission, 2013.

5.1 Introduction

As previously illustrated, the term *human computation* spans a wide range of possible applications and computational distributions. Among all these, many of the most interesting and successful human computation systems not only balance the contribution of human and machine, but also leverage the complementary computational strengths of both parties. As described in Chapter 4, both human and machine bring to the partnership varying strengths and opportunities for action, and during collaboration, each must be able to perceive and access these opportunities in order for them to be effectively leveraged. These affordances define the interaction possibilities of the team, and determine the degree to which each party's skills can be utilized during collaborative problem solving. The set of problems warranting a collaborative technique is equivalent to the set problems where there is an opportunity to effectively leverage affordances on both sides of the partnership in pursuit of the solution.

Instead of deciding who gets (stuck with) which task, we can begin to reason about which party can best contribute to the collective goal at each stage. The answer may not be only the human, or only the machine, but could in fact be both. By framing potential collaboration in terms of the affordances at our disposal, we can then consider which of these affordances could be used to approach a problem and construct a solution.

5.2 Leveraging Human and Machine Affordances

The success of human-computer collaborative systems hinges on effectively leveraging the skills of both the human and the computer. In order to address the problem of balancing and allocating workload in a human-computer collaborative system, it is first necessary to explore the space of problem difficulty relative to human and machine.

Existing complexity models classify problems by measuring the time and/or space required to find the solution using a computer. Under these models, many interesting real-world problems are known to be intractable, even if the path to finding the solution is clear. Other problems have no known solution at all, and are believed to be unsolvable by any computer, no matter how powerful. In contrast, some of these problems are relatively easy for humans to solve (or at least approximate), a notion which lies at the heart of human computation. We can think about the problem space as having two orthogonal dimensions: human diffi-

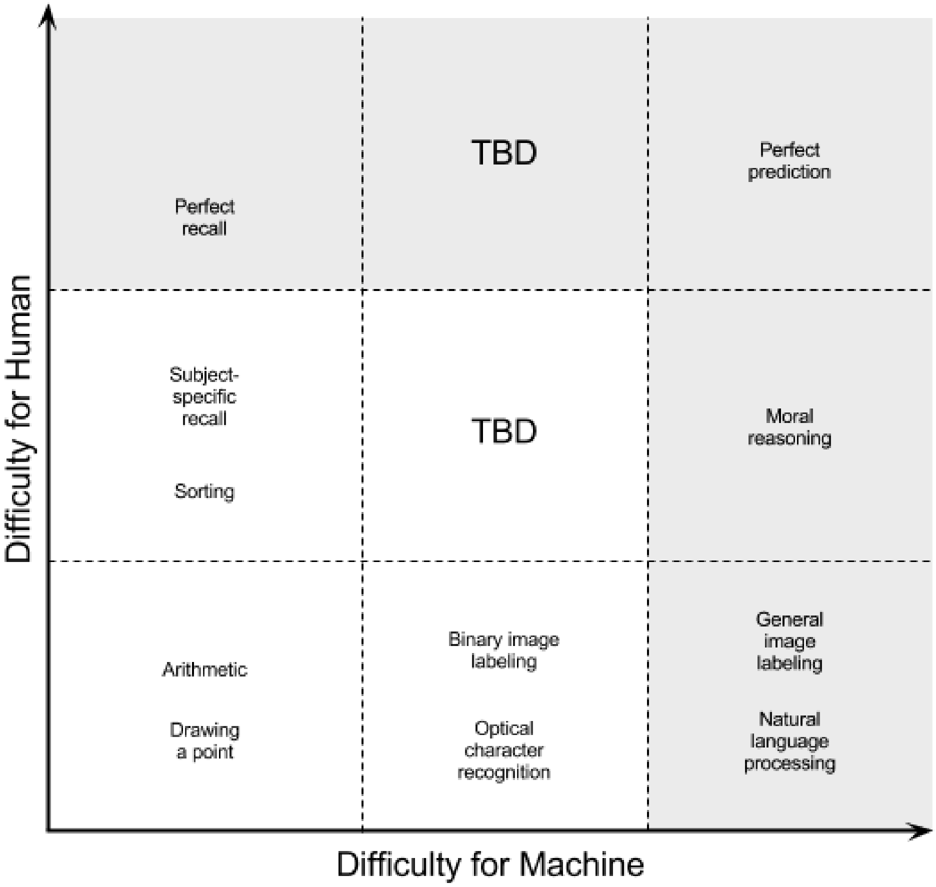


Figure 5.1: A selection of sample problems arranged according to the relative difficulty for human and machine as of this writing. Difficulty increases for the machine as we move to the right along the x axis, and increases for the human as we move up along the y axis.

culty and machine difficulty. Figure 5.1 depicts some well-known sample problems within in this space. In this diagram, problems appearing in the lower left region are trivial; that is, they are comparatively easy for both humans and machines. These problems, such as arithmetic or simple shape rendering, generally do not warrant a human-computer collaborative solution. As we move to the right along the x axis, we encounter many of the problems addressed in early human computation systems: image labeling, character recognition, language processing, etc. These problems are difficult for machines, but relatively straightforward for humans. Here, the overhead cost incurred by involving human processing power is minimal compared with the resources required to achieve comparable performance using a machine. As the field of human computation progresses, we are becoming more invested in applying collaborative techniques to solve problems that are difficult or impossible for either humans or machines alone, but which may be solvable through collaboration. In these problems, we are especially interested in how to best allocate the computational resources of the human and machine collaborators, allowing each party to play to its strengths.

The framework presented in Chapter 4 illustrates the complementary nature of human and machine computation, and attempts to organize existing literature on human-machine collaborative systems according to which skills, or affordances, the system leverages. Other taxonomies [BL10, QB11] propose additional classification dimensions such as *human-machine balance*, *motivation*, *aggregation*, *quality control*, *process order*, and *task-request cardinality*. While these frameworks provide a vocabulary for describing human-computer collaborative systems, they fall short of enabling us to quantify the computational work being done in the human computation algorithms underlying the systems.

Consider for example the numerous published human computation systems in the area of Image Labeling: Ka-captcha [DSG07], Phetch [VAGK+06], KissKissBan [HCL+09], Peek-a-Boom [VALB06], LabelMe [RTMF08] and the ESP Game[VAD04], just to name a few. Categorized within the aforementioned frameworks, these systems have remarkable similarity. Each employs human visual perception and linguistic ability to process and describe images, and uses the machine to distribute tasks and aggregate the results. Many use entertainment as a primary motivator, and redundancy to ensure validity of the resulting data. Given the similarity of the problem as well as the approach to solving it, how do the underlying algorithms compare? We argue that is it critical to develop mechanisms for describing not only *what* each collaborator is tasked with computing, but *how much* they are computing.

5.3 Computation using Human Oracles

Theoretical computer science uses abstract models of computational systems, such as Turing Machines [Tur38], to simulate computational processes and explore the limits of what can be computed. In some cases, it is useful to allow the Turing Machine access to an Oracle – a black box which is able to decide specific problems with perfect accuracy in constant time. Shahaf and Amir proposed an extension to the standard computational model in which questions may be asked of a Human Oracle – an Oracle with human-level intelligence [**?**]. In this model, the Human Oracle is able to answer questions to which a human would be able to respond, even if a machine could not.

In this work, they demonstrate that much of the standard theoretical language holds true when extended to include Human Oracles. This includes concepts such as *algorithmic complexity*, *problem complexity*, and *complexity classes*. They suggest that the complexity of an algorithm executed on such a machine can be represented as a tuple $\langle \Phi_H, \Phi_M \rangle$, where Φ_H indicates the number of queries to the Human Oracle as a function of the input size, and Φ_M is the the complexity of the computation performed by the machine. Whenever the complexity of the machine's computation is the same, the complexity of two algorithms can be compared by considering which requires more queries to the Human Oracle. The minimal complexity of a problem can then be thought of as the minimization of both human and machine cost over all algorithms that correctly solve the problem.

5.3.1 Value of an Oracle Model for Human Computation

Modeling the human contributions to an algorithm as queries to an Oracle captures the underlying behavior of many existing human computation algorithms. For example, in image labeling systems like the ESP Game [VAD04] a human is given some input (an image) and, like an Oracle, is expected to provide a (relatively) correct response to exactly one question: *What do you see?* This interchange, where an external entity is used to inexpensively perform some challenging subroutine, is exactly the kind of system that Oracle machines were designed to describe. Because of this, we adopt the Human Oracle Model as a preliminary mechanism to make quantitative comparisons between human computation algorithms.

Despite the simplicity of the Human Oracle Model, this level of abstraction has several benefits. First, it enables a direct quantification of the cost of an algorithm leveraging human-level intelligence, or human computation, in terms of the number of queries made to the human. This enables a straightforward comparison between two human computation solutions to a given problem on a given input. Second, it enables an objective theoretical comparison between algorithms using humans and existing purely mechanical algorithms, if they exist. Finally, it separates implementation-specific details such as error control, motivation, and interface design from the algorithm itself. This is an important differentiation, and much in keeping with the spirit of traditional complexity models wherein the performance of an algorithm is assessed independent of the languages in which it may later be implemented or the hardware on which it may be run. While questions of recruiting and incentivizing human contributors is by no means unimportant, we specifically investigate the complexity of the underlying algorithms independently.

Technically speaking, a human can simulate any process the machine can execute. After all, we designed the algorithms in the first place. Given an understanding of the process, enough paper and a sufficient supply of pencils, a human operator could write out the contents of each register, perform each bitwise operation, and record each result by hand. However, the time and resources required to compute exactly the same result are exorbitant. In addition, humans are susceptible to fatigue, and we are arguably limited by the capacity of our working memory and unreliable recall. In this sense, human operations are expensive, and there are cases where it is possible to reduce the number of human operations while maintaining optimal performance.

5.3.2 Example: Classification Strategies Using a Human Oracle

Consider the following example from Shahaf and Amir [**?**]: Imagine that we are given n randomly selected samples that we wish to classify. We know that the classifiers are simple threshold functions:

$$h_w(x) = \begin{cases} 1 & : x > w \\ 0 & : x \leq w \end{cases}$$

with the value of w depending on the input. Assume that we do not know the value of w in advance, but that a human can easily partition the data into correct classes. Using the human as an Oracle, there are several ways to approach this problem, each with benefits and drawbacks:

1. We could ignore the human and use a pure machine computational approach, first sorting the set of n samples according to their x values and then choosing a random threshold value that falls between the lowest and highest values. This requires $\langle 0, n \log n \rangle$ time, and guarantees that at least 2 of the samples will be classified correctly. While relatively speedy, this is not a very promising bound on accuracy.

2. We could use a pure human computational approach, asking the human to classify each of the n samples in the dataset. Because as we assumed that the human can always classify samples correctly, this method guarantees 100% accuracy. This method requires $\langle c{*}n, 0 \rangle$ time, where c corresponds to the cost incurred by the human to classify one sample. Under the usual metrics for evaluating algorithmic complexity, the method is technically “faster”. How-

ever, the value of the constant c may be enormous. This would mean that for all reasonably-sized input sets, this approach could be unacceptably slow.

3. Finally, we could try a collaborative solution. First, the machine sorts the set of samples according to their x values, requiring $n \log n$ operations. Next, the human is asked to classify the sample that falls in the middle of the sorted list. If she answers 1, we can infer that all the samples above should also be labeled 1. Similarly, if she answers 0, we know that all the samples below should also be labeled 0. From here, the human is recursively questioned about the middle sample in the remaining half of the list that remains unlabeled. This is simple binary search. Under this approach, the human will be asked to classify at most $\log n$ samples for a total worst-case cost of $\langle c * \log n, n \log n \rangle$. Using this algorithm, we are able to dramatically reduce the workload for the human operator while maintaining 100% accuracy simply by being clever regarding which samples to ask her about.

In this example, the third approach is superior to the other two in terms of maximizing accuracy and minimizing effort. However, the scale of the constant c has yet to be addressed. In human computation, we argue that this scale depends on the affordance being leveraged. This is perhaps most readily apparent in the field of information visualization. Through visualization, we transform the task of assessing abstract numerical information to evaluating visual information, leveraging the human visual processing system and thereby decreasing the per-operation cost c. As designers, it is important to consider the implications of leveraging various combinations of affordances between human and machine. The challenges of assigning numerical value to human processing will be further discussed in Chapter 8.

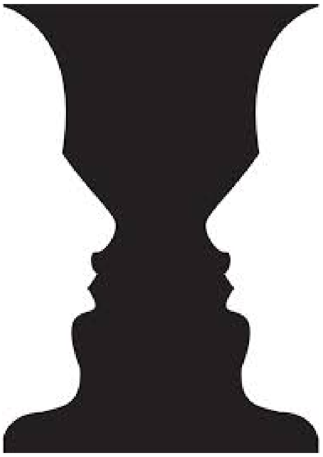


Figure 5.2: The Rubin Vase [Rub15], a bi-stable image with two valid labels: *faces* and *vase*.

5.4 Adapting the Human Oracle Model for Human Computation

Throughout the remainder of this dissertation, we will adopt two slight relaxations of the assumptions about the behavior Human Oracle as differentiated from traditional set-theoretic definitions of an Oracle.

5.4.1 Variability in Human Oracle Responses

By most standard definitions of an Oracle, any two Oracles to the same problem are equivalent with respect to the answers they return to a given query. In contrast, we do not assume that different Human Oracles to the same problem will necessarily return identical answers when queried on the same input. Two Human Oracles may give different answers to the same question when more than one appropriate answer exists. This behavior is perhaps best illustrated through an example. Consider the famous bi-stable image, the Rubin Vase (see Fig. 5.2). In this case, it is difficult to argue that *vase* is a more descriptive label than *faces*, or vice versa; they are equally valid. Whenever there is potential for ambiguity in processing stimuli, there may be more than one valid response for any given input. However, a given individual may strongly favor one label over the other.

We characterize this behavior as follows. Under this model, we will assume that there exist **finitely many** reasonable responses for any query/input pairing:

$$R_Q(x) = \{r_1, r_2, \ldots, r_{n-1}, r_n | r_i \text{ is a reasonable response to query Q on input } x\}$$

We then state that any valid Human Oracle always returns one such reasonable answer, but that we can't predict which one they may decide to return. We can express this nondeterminism by defining the Human Oracle H as having a probability distribution over the collection $R_Q(x)$:

$$D_{H(Q,x)} = \left\{ \langle r_i, P_H(r_i) \rangle | r_i \in R_Q(x), 0 < r_1 \leq 1, \sum_{i=1}^{i \leq n} P_H(r_i) = 1 \right\}$$

where $P_H(r_i)$ is the probability that Human Oracle H returns response r_i when passed query Q on input x. In the simplest case, $n = P_H(r_n) = 1$. That is, if there is only one reasonable response, the Human Oracle will return that response with probability 1. When there are multiple reasonable responses, the Human Oracle's probability distribution may heavily favor some subset of responses. We suggest that this nondeterministic behavior helps capture the influence of individual differences inherent in any human population. These inconsistencies may be due to different lived experiences, internal biases, or preferences. In addition to individual differences, this distribution may be influenced through incentivization. This may happen *a priori*, such as in systems that incentivize the generation of short responses over more verbose ones, or the distribution may be changed *on-the-fly*, such as in gameified systems where the players may be asked to match (or avoid matching) a partner's responses.

In practice, individual differences may dictate that a human's probability for giving a specific response is in fact zero. For example, a person may never have encountered a *durian* before, although if presented with an image of one they may still be able to recognize it as a kind of *fruit*. However, in this model, we will assume nonzero values for all $P_H(r_i)$. That is, we assume that a Human Oracle is aware of all possible reasonable responses to any query, although the probability that they return a specific response may be arbitrarily small. This assumption is consistent with traditional set-theoretic notions of an Oracle and is useful in characterizing the notion of *collective intelligence* relied upon in many human computation applications.

5.4.2 Persistence of Previous Responses

If the same Human Oracle is queried more than once on the same input during the execution of an algorithm, we may wish to assume that it will be aware of its previous responses and will only return each answer once. This is akin to assuming that upon reading the input, the Human Oracle constructs a predefined sequence of answers by ordering their possible responses in decreasing order of probability:

$$A_{H(Q,x)} = (a_1, a_2, \ldots, a_n | P(a_i + 1) < P(a_i) \ \forall \ 1 \leq i \leq n)$$

The Human Oracle will answer each query about that particular input by simply reporting the next unused element in the sequence. This reflects human short-term memory, and can be simulated by recording previous responses in the main algorithm and passing the entire history back as part of the input to a non-persistent Oracle. We will discuss the ramifications of whether or not the Human Oracle is able to *compute* on these previous responses in Chapter 7.

5.4.3 Additional Assumptions

Additionally, we presume that the Human Oracle can efficiently generate an answer to the queries we pose. In traditional computational models, it is assumed that the Oracle can interpret and respond correctly to the query in constant time. However, it is also acceptable to consider Oracles with other (bounded) response time complexities. With Human Oracles, we do not necessarily know how long it takes a person to solve the problem. For simplicity, we will assume a constant cost for each query to the Human Oracle, which enables us to consider the complexity of two algorithms leveraging the same kind of Human Oracle in terms of the number of queries required. This assumption will be discussed in further detail in Chapter 8.

Finally, the study of human computation presumes the existence problems for which humans are *faster* than any known machine algorithm. To that end, we only consider problems in which the Human Oracle's answers are integral to computing the solution. That is, the algorithm querying the Human Oracle cannot efficiently generate answers to its own queries, and must rely on (and potentially validate) the responses it receives.

We believe that these adaptations result in a model that more closely resembles observed behavior in systems involving human computation, and help capture some of the ambiguity inherent in many interesting human computation problems. In the following chapter, we use this model as a lens to explore various problems that fall under the umbrella of image labeling.

Chapter 6

Image Labeling under the Human Oracle Model

This chapter is an extension of the following manuscript:

- Crouser, R. J., Hescott, B., Glaser, M., & Chang, R. Theoretical Bounds for crowdsourced image labeling under a human oracle model. AAAI Conference on Human Computation & Crowdsourcing. In Submission, 2013.

6.1 Introduction

In this chapter, we demonstrate the utility of the Human Oracle model for comparing various Human Computation approaches to Image Labeling. We do not mean to imply that Image Labeling is necessarily a "canonical" or "complete" problem for Human Computation, as this concept is yet ill-defined. However, we believe that a close examination of a well-studied problem through this lens may provide insight into to the structure of Human Computation algorithms. We hope that this will serve as an initial benchmark by which other problems may be measured as we continue to explore the space of Human Computation.

6.2 Example Image Labeling Games

Using the Human Oracle model, we explore how we can describe the underlying algorithmic behavior and performance of these systems. In some cases, we have chosen to model a slight variation of the system in the interest of facilitating a more interesting comparison. When this is the case, we will clearly document any modifications and provide justification for the change.

6.2.1 The ESP Game

The ESP Game [VAD04] is a Human Computation system designed to produce validated labels for Images on the web. Each image is displayed to a randomly-assigned pair, who are then asked to label the image in a finite amount of time. We describe the problem that humans are being asked to solve in the ESP Game in terms of its input and output as:

`DESCRIBE(`I`)`:

Input:	an image I
Output:	a label ℓ describing the image

Natural language analogs to this problem might be *"What do you see in this image?"*

Because the human players cannot communicate with one another as they try to "agree" by guessing the same label, the dynamics of the game incentivize them to try guessing short, intuitive labels. While the ESP Game doesn't explicitly limit the length of the users' responses, this generally limits people's responses to single descriptive words.

A label is accepted once some number of pairs have agreed on it, and is then added to a list of `TABOO` words for that image. Future pairs are presented with the `TABOO` words in addition to the image, and these words are not accepted if guessed. For the purposes of this analysis, we will assume that one just pair must agree for a label to be accepted. A Human Oracle Machine that executes the procedure in Algorithm 1 simulates the ESP Game on one image.

In Algorithm 1, a pair of Human Oracles to the `DESCRIBE` problem are re-

Algorithm 1: Oracle-ESP

```
Input : An image $I$
        A set of unacceptable labels TABOO
Output: A label
Let $H_1, H_2$ be Human Oracles to the DESCRIBE problem
$\texttt{labels}_{H_1}, \texttt{labels}_{H_2} = \{\}$
while ($\texttt{labels}_{H_1} \cap \texttt{labels}_{H_2}$ is empty) do
    $new\_label_1 = H_1.describe(I)$
    if $new\_label_1 \notin$ TABOO then
        $\texttt{labels}_{H_1}.add(new\_label_1)$
    $new\_label_2 = H_2.describe(I)$
    if $new\_label_2 \notin$ TABOO then
        $\texttt{labels}_{H_2}.add(new\_label_2)$

$valid\_label = \texttt{labels}_{H_1} \cap \texttt{labels}_{H_2}$
return $valid\_label$
```

peatedly queried for appropriate labels for the same Image. Querying continues until there is an overlap in their response histories. In the actual implementation of the ESP Game, the players can see the TABOO list, and will avoid those words in the interest of maximizing their score. We assume for simplicity that the Human Oracles ignore the TABOO list; responses that are listed as TABOO words are simply discarded. The resulting output is a new text description of the Image that has been validated by both Human Oracles.

Recall that each Human Oracle has a finite list of labels they could use to describe a given image. In the best case, Algorithm 1 terminates after only 2 queries, one to each Human Oracle whose first choice labels are a match. In the worst case, the Human Oracles' response lists are exactly inverted, and Algorithm 1 requires $n+1$ queries before they overlap. In the event that either H_1 or H_2 cannot return a new label, we assume that the computation will throw an error. When this occurs, we can infer that all valid labels for the input image are already listed in the TABOO list; if this were not the case, then the Human Oracles would have guessed the missing label.

6.2.2 KissKissBan

KissKissBan [HCL+09] is another Human Computation system designed to produce short, validated labels for Images on the web. This system suggests an extension of the ESP Game intended to generate more creative labels by adding a competitive element to the game dynamics. Each image is shown to three online players. Two players are collaborating, as in the ESP Game, to try to guess the same label for the image. The other player, the Blocker, attempts to block the collaborative pair from matching by blocking "obvious" labels at the onset of each round. The collaborators win the game if they successfully match on a non-blocked word before their time runs out, and are penalized for guessing blocked words. If they fail to match on a new word, the Blocker wins.

Algorithm 2 simulates KissKissBan on one image. As in the ESP Game, the human players can be characterized as Human Oracles to the `DESCRIBE` problem. However, unlike the ESP Game, KissKissBan can potentially produce more than one label for the image:

`MULTI_DESCRIBE(`I, k`)`:

Input: an image I
Output: a collection of between 1 and k labels for the image

There are three ways a label can be validated during the game: (1) H_1's label matches one from $H_{Blocker}$, (2) H_2's label matches one from $H_{Blocker}$, or (3) H_1 and H_2 match on a label as in the ESP Game. The resulting output is a *set* of text descriptions that have each been validated by at least two Human Oracles. Note that while matching on a blocked word produces a validated label, the game ends only on a match between the two collaborators. Thus, in the minimal case H_1 and H_2 match on their first label and this label is not blocked, requiring a total of $k+2$ queries to generate a single label. Unlike with Algorithm 1, the minimal case is not optimal in terms of minimizing queries-per-label. In the best case, H_1 and H_2 differ on their first $\frac{k}{2}$ guesses, each of the resulting $k-1$ labels are blocked, and they then match on their next guesses, thus requiring $2k$ queries to generate k labels. In the worst case, H_1 and H_2 are exactly inverted on their first $(n-k)$ responses and none of these are blocked. When this is the case, Algorithm 2 requires $k + \frac{n-k}{2}$ queries to generate a single label.

```
Algorithm 2: Oracle-KissKissBan
   Input : An image $I$
           An integer $k$
   Output: A label
 1 Let $H_1, H_2, H_{Blocker}$ be Human Oracles to the DESCRIBE problem
 2 valid, labels$_{H_1}$, labels$_{H_2}$, BANNED $= \{\}$
 3 for $i$=1 to $k$-1 do
 4     BANNED.$add(H_{Blocker}.describe(I))$
 5
 6 while (labels$_{H_1} \cap$ labels$_{H_2}$ is empty) do
 7     $new\_label_1 = H_1.describe(I)$
 8     if $new\_label_1 \notin$ BANNED then
 9         labels$_{H_1}$.$add(new\_label_1)$
10     else
11         valid.$add(new\_label_1)$
12     $new\_label_2 = H_2.describe(I)$
13     if $new\_label_2 \notin$ BANNED then
14         labels$_{H_2}$.$add(new\_label_2)$
15     else
16         valid.$add(new\_label_2)$
17
18 $valid.add($labels$_{H_1} \cap$ labels$_{H_2})$
19 return $valid$
```

Note the similarity between lines 3–11 of Algorithm 1 and lines 6–18 of Algorithm 2; the only modification is the recording of matches to blocked labels on lines 10–11 and 16–17 of Algorithm 2. Under this model, KissKissBan appears to be running the ESP Game as a subroutine between the two collaborators. This similarity will be discussed in detail in the section on Comparing Image Labeling Algorithms.

6.2.3 Polarity

Polarity [LA11] is a two-player Human Computation game to validate existing image labels or attributes, as well as reapply them to similar images. These attributes can be generated through mechanisms such as the ESP Game [VAD04] or KissKiss-Ban [HCL$^+$09], through other interactive attribute generation methods [PG11] or through manual curation (for example [KBBN09]). In this game, two players are presented with a set of images and an attribute (e.g., "has a blue body"). Each player is assigned one of two roles – the *positive* player is asked to select images that the attribute describes, while the *negative* player is asked to select images that the named attribute **does not** describe.

After each player has selected a subset of the images according to her role, the resulting partitions are then compared. All images that were selected only by the positive player are considered matches to the attribute, all images that were selected only by the negative player are considered matches to the negative of the attribute (e.g., "does **not** have a blue body"), and any image that was either selected by both players or left unselected is considered ambiguous and marked for further review. In this form of mutual validation (known as "complementarity agreement" [LA11]), the players are penalized for any overlap in their responses. To discourage trivial complementarity, where one partner selects all images and the other partner selects none, players receive a joint score of $(|I_{hit}| \times |I_{miss}|) - c \cdot |I_{hit} \cap I_{miss}|$, where I_{hit} is the set of images selected by the positive player, I_{miss} is the number of images selected by the negative player, and c is the penalty for selections that overlap between the two players.

At the task-per-image level, we can think of the human players as Oracles to the *decision problem analog* to the previous `DESCRIBE` problem:

`VALIDATE(`I, ℓ`)`:

Input:	an image I and a label ℓ
Output:	`TRUE` if ℓ describes I, `FALSE` otherwise

The Human Oracles' responses are then aggregated in order to solve a larger problem of classifying the images by their relationship to the input label ℓ:

CLASSIFY(I^*, ℓ):

Input: a set of images I^* and a label ℓ
Output: a partition of the images into those that match ℓ and those that match $\neg\ell$

Algorithm 3 simulates Polarity on a collection of k images using a pair of Human Oracles to the VALIDATE problem. Each Human Oracle is queried once per image for a total of $2k$ queries. Note that for Algorithm 3, the number of queries in the best and worst cases is the same. In the best case the partition is unambiguous, and so all k images can be labeled either ℓ or $\neg\ell$. In the worst case all images are added to both the $H_{positive}$ and $H_{negative}$ sets, and none of the k images can be labeled.

```
Algorithm 3: Oracle-Polarity
Input : A set of images $I^* = \{I_1, \ldots, I_k\}$
        A label $\ell$
Output: A collection of images that match $\ell$ and a collection of images
        that match $\neg\ell$
Let $H_{positive}, H_{negative}$ be Human Oracles to the VALIDATE problem
hits, misses = {}
for $i \in 1, \ldots, k$ do
    if ($H_{positive}.validate(I_i)$ == TRUE) then
        hits.$add(I)$

    if ($H_{negative}.validate(I_i)$ == FALSE) then
        misses.$add(I)$

$positive\_matches$ = hits \ (hits ∩ misses)
$negative\_matches$ = misses \ (hits ∩ misses)
return {$positive\_matches, negative\_matches$}
```

6.2.4 Peekaboom

Peekaboom [VALB06] is a Human Computation system designed to augment image labels with information about the location of the objects being described. Two players are partnered at random and are each assigned a role: Peek and Boom.

Boom is presented with an image and a word, and Peek is presented with a blank screen. Boom is tasked with revealing enough of the image to Peek that she can guess the word. As Boom clicks on parts of the image, a small region of the image under the clicked location is revealed, and the incomplete image is sent to Peek. The task given to Peek is identical to players of both the ESP Game and KissKissBan: given an image (in this case, an incomplete image), provide a description. Both players are incentivized to reveal and guess as efficiently as possible. The game infers that if Peek is able to guess the word, then Boom must have revealed the correct location. Once Peek has successfully matched the original word, a minimal bounding box is computed from the regions revealed by Boom. Experimental data suggest that the bounding boxes produced by multiple pairs when averaged tend toward minimally bounding the region containing the object.

Again, we can consider Peek to be a Human Oracle to the `DESCRIBE` problem. The task given to Boom is one we have not yet seen:

`LOCATE_OBJECT(`I, t_{obj}`):`

Input:	an image I
	a textual description t_{obj} of an object *obj*
Output:	(x, y) location of part of *obj* in I

In this problem, we either assume that the textual description has been validated a priori, or that the computation will throw an error if the object does not appear in the image. It is relevant to note that the larger problem being solved in Peekaboom is different from the either of the problems being solved by the two Human Oracles. Instead, the goal of Peekaboom is:

`BOUND_OBJECT(`I, t_{obj}`):`

Input:	an image I
	a textual description t_{obj} of an object *obj*
Output:	a minimal bounding box $\langle (x_1, y_1), (x_2, y_2) \rangle$ around *obj* in I

Peekaboom solves `BOUND_OBJECT` using the aggregated result of many queries to a Human Oracle to `LOCATE_OBJECT`, which is validated using a Human Oracle to `DESCRIBE`. Algorithm 4 captures this behavior. In the best case, one reveal from

Algorithm 4: Oracle-Peekaboom

```
Input : An image $I$
        A label $\ell_{obj}$
Output: A bounding box
Let $H_{Peek}$ be a Human Oracle to the DESCRIBE problem
Let $H_{Boom}$ be a Human Oracle to the LOCATE_OBJECT problem
labels, points = {}
$I'$ = a blank image the size of $I$
while $\ell \notin$ labels do
    $new\_point = H_{Boom}.locate(I, \ell_{obj})$
    points.$add(new\_point)$
    // Construct new image $I'$ by
    // aggregating subimages
    $I' = I' \cup$ (subimage of $I$ under $new\_point$)
    $new\_label = H_{Peek}.describe(I')$
    labels.$add(new\_label)$

Compute a bounding box $B$ around all $p \in$ points
return $B$
```

H_{Boom} is sufficient for H_{Peek} to guess correctly on the first try, for a total of 2 queries to validate ℓ. In the worst case, H_{Boom} must reveal all $r \times r$ subregions of the entire $m \times m$ image before H_{Peek} can identify the correct label, resulting in a total of $2(m \div r)^2 = O(m^2)$ queries to validate ℓ. In contrast to the two previous applications, in which humans are asked perform the same task and their responses are used to to verify one another, Peekaboom uses implicit validation on humans performing different tasks. This hints at an underlying relationship between different Image Labeling problems, which will be further discussed in the following section.

6.3 Relative Computability using Reductions

In the study of computability and computational complexity, a **reduction** is a procedure for transforming one problem into another. Intuitively, a reduction from one problem A to another problem B demonstrates that access to an algorithm for solving problem B could also be used as a subroutine to solve problem A. This relationship holds regardless of which algorithm for solving B might be chosen, and even in cases where such an algorithm does not exist (i.e. using an Oracle). Reductions are often used to describe the relative difficulty of two problems: that is, a reduction from A to B (written $A \leq B$) may be used to demonstrate that solving A is *no more difficult* than solving B or equivalently, B is *at least as difficult* as solving A.

Reductions illustrate that a solution to problem A is *computable* given an algorithm to solve problem B, but do not guarantee that this computation is necessarily efficient. Such reductions may require calling the subroutine to solve B multiple times. This is sometimes accomplished through **nondeterminism**; that is, the procedure may simultaneously follow more than one path to compute a solution. Because nondeterministic machines are no more computationally powerful than deterministic machines, this convention can be used to describe the relationship more elegantly. In this section, we demonstrate reductions between the problems being solved by the ESP Game, KissKissBan, Polarity, and Peekaboom. These reductions enable us to compare the relative complexity of these games under the Human Oracle Model. Throughout the remainder of this chapter, we will assume for the sake of discussion that the cost of generating a label for an image, validating whether or not a label matches an image, and locating an object in an image is the same, under the precedent set by [MS12]. For further discussion on the challenges of quantifying cost, see Chapter 8.

6.3.1 Comparing ESP and KissKissBan

Intuitively, the ESP Game and KissKissBan appear very similar both in terms of the problem they are trying to solve as well as the approach to finding a solution. To explore this similarity, we compare the ESP Game and KissKissBan both by reduction between their underlying problems and by analyzing their algorithmic complexity. Specifically, we demonstrate that their underlying problems are equivalent and that the ESP Game and KissKissBan have identical performance in both the best and worst case.

6.3.1.1 Problem Reduction

We begin by demonstrating that solving MULTI_DESCRIBE (solved by KissKissBan) is no more difficult than solving DESCRIBE (solved by the ESP Game), and vice versa.

Lemma 6.3.1 DESCRIBE $\leq$ MULTI_DESCRIBE.

Proof: On any image passed to DESCRIBE, call MULTI_DESCRIBE as a subroutine with $n = 1$ to give us a single label, and return this label. □

Lemma 6.3.2 MULTI_DESCRIBE $\leq$ DESCRIBE.

Proof: On any image passed to MULTI_DESCRIBE, we can call DESCRIBE as a subroutine n times to give us a n labels. □

By demonstrating (trivial) reductions in both directions, we verify our intuition that these problems are equivalent:

Theorem 6.3.3 MULTI_DESCRIBE *and* DESCRIBE *are equivalent.*

6.3.1.2 Algorithmic Comparison

Because their underlying problems are equivalent and their Human Oracles differ in number but not in function, we can directly compare the performance of the ESP Game and KissKissBan algorithms under the Human Oracle Model. Specifically, we show that the best and worst case performance of both games is identical (measured by the ratio of Human Oracle queries to number of labels produced).

Theorem 6.3.4 *The worst case performance of the ESP Game requires no more queries per label than the worst case performance of KissKissBan.*

Proof: Recall that in the worst case, KissKissBan returns just a single label with a large number of queries to the Human Oracles. All $k - 1$ queries to $H_{Blocker}$ were wasted because none of the BLOCKED labels were matched, and the collaborators go

$\frac{n-(k-1)}{2}$ rounds before finding a match for a total cost of $n+1$ queries. In this case, returning a single label could have been accomplished using one round of the ESP Game at an identical cost of $n+1$ queries. While the two Human Oracles may take just as long to find a match, there is no added benefit to including a third Human Oracle in the worst case. Thus, the worst-case number of queries to generate a single label in KissKissBan is equal to the worst-case cost of the ESP Game. $\square$

Theorem 6.3.5 *The best case performance of KissKissBan requires no fewer queries per label than the best case performance of the ESP Game.*

Proof: In the best case, KissKissBan returns k unique labels using only $2k$ queries to the Human Oracles: $(k-1)$ to $H_{Blocker}$ to set up the `BLOCKED` list, $(k-1)$ queries divided between H_1 and H_2, each of which matches a unique word on the `BLOCKED` list, and 2 final queries, one to each of H_1 and H_2, on which they match. This match causes the algorithm to terminate. To produce the same number of labels, we would require k sequential rounds of the ESP Game (recall that each round produces at most 1 label). By making `TABOO` the list of labels generated through previous rounds, we ensure that all k labels produced by the sequence of ESP Games will be unique. In the best case, the pair is able to match on their first try in each round, for a total of $2k$ queries to the Human Oracles. Thus, the minimum number of queries per label in the best-case performance of KissKissBan is equal to the best case cost of k rounds of the ESP Game. $\square$

It is reasonable to argue (as do the authors of [HCL^{+}09]) that KissKissBan may produce more diverse labels than the ESP game in the short term. In the long term, there are no labels that KissKissBan would produce that would not also eventually be discovered in the ESP Game; they may just be validated in a different order. This model and the corresponding proofs above demonstrate that this effect is due more to the incentive structure of the game than to any underlying computational differences. However, from an algorithmic perspective, KissKissBan demonstrates no advantage over the ESP Game in terms of the number of queries per label.

6.3.2 Comparing ESP and Polarity

In this section, we will explore the relationship between the ESP Game, which generates labels for an image using two Human Oracles to the `DESCRIBE` problem and Polarity, which attempts to `CLASSIFY` a set of images according to an existing labels using two Human Oracles to `VALIDATE`, the decision-problem analog to `DESCRIBE`.

6.3.2.1 Problem Reduction

We will use bounded nondeterminism to demonstrate that:

Lemma 6.3.6 `CLASSIFY` $\leq$ `DESCRIBE`.

Proof: Given a collection of images I^* and a description ℓ, nondeterministically query `DESCRIBE` for all possible descriptions for each image $I_i \in I^*$. If any of the returned descriptions for an image I_i matches ℓ, classify I_i as a match. Otherwise, classify I_i as a non-match. □

The number of possible valid descriptions for any input image is finite due to the limitations of both image resolution and language. As part of the reduction, we assume we have access to an efficient method for solving `DESCRIBE` that may return any one of these finitely many descriptions, but won't return anything else. Thus, if ℓ is a valid description for any image in the collection, it will eventually show up as one of the suggested descriptions returned by `DESCRIBE`. We are therefore guaranteed that this nondeteriministic "guess-and-check" method will eventually identify all images that match ℓ, as well as all images that don't.

In this case, the reduction only holds in one direction; it is not possible to reduce from `CLASSIFY` to `DESCRIBE`. This is because one of the inputs `CLASSIFY` is a validated label. Although a guess-and-check method could be used again, it would rely on the assumption that the English language is finite. While this assumption is technically correct, it is not realistic or feasible. We would have to iterate over all possible labels until one was found that did not cause an error. This validates the intuition that the two problems are not equivalent, but are nonetheless related.

6.3.2.2 Algorithmic Comparison

The previous reduction implies that it should be possible to compute the output of Polarity by using only calls to the ESP Game. Recall that the goal of Polarity is to return a partition over a collection of images into those that match the input label ℓ and those that do not. As indicated above, we can accomplish this same task by running the ESP Game on each image independently. We can repeatedly invoke the Oracle-ESP algorithm on each image until either the algorithm returns a matching label or we exhaust the possible labels for the image without finding a match. Because (1) there are finitely many valid labels for any image, and (2) the Oracle-ESP algorithm will eventually return all valid labels, we can be assured that this process eventually terminates. In this section, we demonstrate that with respect to the number of images, the maximum number of queries to the human oracle to solve this problem the ESP Game grows only linearly faster than solving the problem using Polarity.

Theorem 6.3.7 *The ESP Game and Polarity both require $O(k)$ queries to partition a set of k images.*

Proof: The number of queries required by Polarity to label a collection of k images is always $2k$, regardless of the outcome. When solving the classification problem using the ESP Game, the total number of rounds is determined by k, the number of images we will need to evaluate. For the ESP Game, the highest cost would be incurred when none of the images in the collection are matches to ℓ. Each image would require at n executions the Oracle-ESP algorithm to fully label where n is the maximal (finite) number of possible reasonable labels for a single image. Adding each returned label to the `TABOO` list and thereby removing it from the pool of potential guesses for subsequent rounds, we could could incur a maximal cost of $n - (k - 1) + 1$ on the k^{th} round. The maximum cost per image is therefore given by the sum:

$$\sum_{k=1}^{n}(n - (k - 1)) + 1 = \sum_{k=0}^{n-1}(n - k + 1) = \frac{n^2 + 3n}{2}$$

Thus, the maximum cost incurred to partition a collection of k images with at most n potential labels per image using only the ESP Game is $k * \frac{n^2+3n}{2} = O(k)$. $\square$

This analysis comes with a caveat: while this model enables us to demonstrate the two algorithms are similar under a standard measure for algorithmic complexity, the role of the constant factor does not go unnoticed. Using the ESP game as indicated above would require significantly more queries to its Human Oracles because unlike in Polarity, we haven't pruned the initial search space by telling the Human Oracles where to focus their attention. Because of this, they may have to exhaustively explore all of the (finitely many) possible labels before they can give us the answer we seek. In practice, the number of valid labels for most images is relatively small. Results from the original publication on the ESP Game indicate that after 4 months of near constant play with over 13,600 players, only 0.3% of the images in their dataset had more than 5 validated labels [VAD04]. KissKissBan reports a higher average of about 14 labels per image with 78.8% recall [HCL$^+$09], indicating that some of these labels may be noise. Because the size of n is reasonably small for most images, demonstrating this similarity in asymptotic growth illustrates that knowing the label we're trying to match in advance yields only a modest advantage in solving this problem. In the following section, we will explore the relationship between two algorithms whose complexity with respect to the input size differ on a much deeper level.

6.3.3 Comparing ESP and Peekaboom

On the surface, the ESP Game and Peekaboom appear similar in many ways. Both compute on a single image, and both leverage a Human Oracle to the `DESCRIBE` problem to perform some part of their computation. In this section, we illuminate some fundamental differences between these algorithms and their underlying problems.

6.3.3.1 Problem Reduction

To demonstrate that:

$$\texttt{BOUND\_OBJECT} \leq \texttt{DESCRIBE}$$

we will add a second layer of nondeterminism to the argument from the previous reduction showing that CLASSIFY $\leq$ DESCRIBE.

Proof: Given an image I and a description ℓ_{obj} of an object to bound, nondeterministically select one of a subimage I'. On I', nondeterministically query DESCRIBE for all possible descriptions. If any of the returned descriptions matches ℓ_{obj}, return the boundary of the subimage as the bounding box. □

The number of possible subimages is limited by the size of the image. As before, the number of possible valid descriptions for any input image is also finite due to the limitations of both image resolution and language. As part of the reduction, we assume we have access to an efficient method for solving DESCRIBE that may return any one of these finitely many descriptions, but won't return anything else. Thus, if ℓ_{obj} is a valid description for the subimage, it will eventually show up as one of the suggested descriptions returned by DESCRIBE. We are therefore guaranteed that this nondeteriministic "guess-and-check" method will eventually yield a correct bounding box.

As in the previous section, this reduction only holds in one direction; it is not possible to reduce from DESCRIBE to BOUND_OBJECT without iterating over all possible words in the English language. This confirms that the two problems are not equivalent, but are nonetheless related.

6.3.3.2 Algorithmic Comparison

The previous reduction implies that it should be possible to compute the output of Peekaboom by using only calls to the ESP Game. In this section, we demonstrate that the maximum number of queries to the Human Oracle using either Peekaboom or the ESP Game required are both polynomially bounded in the size of the image, these polynomial bounds are not equivalent.

Proof: Recall that the goal of Peekaboom is to return a minimal $w \times h$ bounding box surrounding the described object in the image, and that this is accomplished by having Boom sequentially reveal parts of the image to Peek. Assume without loss of generality that the size of the image is $m \times m$, and that the size of each revealed region is $r \times r$, where $0 < r < m$. The smallest possible bounding box, where $w = h = r$, would be returned in the case that Peek was able to guess the word after a single reveal. In the worst case $w = h = m$, because Peek may not be able to guess the word before seeing the entire image, which could require at most $2 * (m \div r)^2 = O(m^2)$ queries to the Human Oracles.

As indicated above, we can repeatedly invoke the Oracle-ESP algorithm on each subimage in ascending order of size until either the algorithm returns a matching label or we exhaust the possible labels for the subimage without finding a match, and move on to the next one. Because (1) the label given as input to Peekaboom was validated a priori, (2) there are finitely many valid labels for any image, and (3) the Oracle-ESP algorithm will eventually return all valid labels, we can be assured that this process eventually terminates. Because we evaluate subimages in increasing order of size, this process guarantees that the first subimage on which we find a match is minimal.

The total number of times we must execute the Oracle-ESP algorithm is determined by the number of subimages that must be evaluated. The smallest bounding box that could be returned by Peekaboom is the size of one revealed region, and so we need only evaluate subimages that are at least $r \times r$, and that are at most the entire image. The number of possible subimages ranging from size $r \times r$ to $m \times m$ is:

$$\sum_{w=r}^{m} \sum_{h=r}^{m} (m - w + 1)(m - h + 1) = O(m^4)$$

thus requiring on the order of $O(m^4)$ queries to the Human Oracles across all executions of the algorithm. Thus, the worst-case number of queries needed to bound an object using only the ESP Game grows asymptotically faster than the number of queries needed using Peekaboom. □

In the proofs above, we used brute force to illustrate the relationship between the ESP Game and Polarity, as well as between the ESP Game and Peekaboom. This demonstrates the relationship between *labeling an image*, *classifying an image* and *locating an object in an image* in an intuitive manner, but we reiterate that this is not intended as a prescription for future design. In practice, because this method requires an exhaustive search, this approach would not be an effective use of Human Computation resources. The average case performance could likely be significantly improved by making more intelligent decisions about which subimages to send to the subroutine for validation. We could, for example, start with a well-distributed subset of subimages of each size. This has the potential to greatly reduce the number of calls to the subroutine because it could de-prioritize redundant rounds on uninteresting regions of the image without sacrificing accuracy. We could also select regions according to content by preprocessing using an image segmentation algorithm. This would increase the amount of machine computation, in exchange for a reduction in the number of queries to the Human Oracles. However, these heuristics would not alter the underlying differences between these algorithms.

6.4 Probabilistic Performance

The Human Oracle Model enables us to bound the best and worst case performance in terms of the number of queries to the oracle. Under this model, we can also describe the *average cost* of Human Computation algorithms in terms of the number of queries needed to solve the problem most of the time. For example, consider running the Oracle-ESP algorithm with randomized Human Oracles: both Human Oracles have identical predefined sets of n labels for the input image, but they appear in a random order in their response lists. In other words, both Human Oracles know all n possible labels for the image, but they may have drastically different ideas of which ones are the most important. In the worst case, their lists are in exactly the opposite order. When this is the case, the `while` loop in line 3 of the Oracle-ESP algorithm will iterate $\frac{n}{2}$ times without making a match, making a total of $n + 2$ queries before a match is guaranteed.

However, this case is highly unlikely due to the conditional probability of each selection. Assume that we have executed k iterations of the `while` loop without a match. This means that in $2k$ queries to the Human Oracles, $2k$ of the possible labels have been guessed so far; half by H_1 and the other half by H_2. The (conditional) probability that the $(k+1)^{st}$ iteration results in a match is:

$$\frac{(n-2k)(n-2k-1)}{(n-k)^2}$$

Trivially, the (conditional) probability that the $(k+1)^{st}$ iteration does **not** result in a match is:

$$1 - \frac{(n-2k)(n-2k-1)}{(n-k)^2}$$

Conditional probabilities can be multiplied to find the probability that related events occur in sequence. In this case, the probability that a game of ESP goes m iterations without a match is:

$$\prod_{k=0}^{m} 1 - \frac{(n-2k)(n-2k-1)}{(n-k)^2}$$

On an image with 10 acceptable labels ($n = 10$), this indicates that the game terminates with 3 or fewer iterations 71% of time time, and in 4 or fewer iterations 93% of the time.

In many cases, probabilistic performance is more telling than worst-case analysis, as it provides a clearer picture of the how the algorithm can be expected to perform over the long-term. Information about the long-term expected performance of an algorithm could be used to help refine the practice of tuning parameters such as *timeout*, which could improve efficiency over static settings under certain conditions. We believe that establishing bounds (best, worst, and average-case) on algorithmic processes involving human computation and deepening our understanding of the relationships between the problems were trying to solve, as well as identifying areas of redundancy, will enable us to design more efficient algorithms in the future. In addition, reporting bounds on the complexity of human computation algorithms along with the observed performance of the system would improve study reproducibility.

Chapter 7

Classification Dimensions and Induced Complexity Classes

7.1 Introduction

As demonstrated in the previous chapter, the number of required operations is one intuitive metric by which we may order a collection of algorithms for solving a problem. Indeed, this is analogous to traditional notions of computational work. Because we lack a mechanism for converting between units of *human work* and units of *machine work*, the $\langle \Phi_H, \Phi_M \rangle$ notation introduced by Shahaf and Amir [**?**] can prove useful. Recall for example the two techniques discussed in Chapter 6 for identifying the location of an object in an $m \times m$ image. Using Peekaboom, the number of queries to the Human Oracles is bounded by $O(m^2)$. The cost to the machine is a simple comparison between each of Peek's guesses and the input label, and so $\Phi_M = \Phi_H = O(m^2)$ as well. In the second approach using the ESP Game as a subroutine, the number of queries to the Human Oracles could be as high as $O(m^4)$ in the event that all n labels needs to be guessed before we arrive at the input label. The machine must then compare the value returned by each query to the collection of previous guesses for that round to determine if there has been a match. Assuming an efficient data structure, each lookup or insertion would incur a cost on the order of $\log(n)$. In addition, each of the n possible returned labels must be compared against the input label. This results in a total machine cost of $O(m^4 * \log(n) + n)$. Comparing these two tuples, it is clear that Peekaboom is a more efficient method for bounding an object in and image than the ESP Game approximation in terms of both human and machine computation.

Perhaps more interestingly, the examples given in Chapter 6 demonstrate that an algorithm requiring **more information** (Peekaboom requires a predefined label) as well as **interactivity** (Boom must be able to touch the image) can be simulated using a polynomial number of calls to an algorithm with limited interactivity and unrestricted search space (ESP). This sheds important light on the *value* of this additional information and power, and the scale of the cost incurred if we are forced to solve the same problem using an intuitively "weaker" machine. This notion of "stronger" and "weaker" suggests that the way in which human computation is leveraged as part of an algorithmic process may be used to induce *complexity classes* that partition the space of human computation. In the remainder of this chapter, we will discuss several additional dimensions along which human computation may be classified and compared.

7.1.1 Problem Instance

To begin, we consider whether or not the Human Oracle can influence the specific instance of the problem that it is being asked to solve. In many existing human computation algorithms, the human or collection of humans is asked to perform a *specific* computational process as a subroutine to a larger algorithm, such as labeling a particular image. Under this restriction, the Human Oracle does not have any power to determine which problem it will solve. Algorithms where the Human Oracle is not allowed to alter the problem instance are leveraging a relatively restrictive use of human processing power. We call these **fixed instance** human computation algorithms. This is consistent with the standard Oracle Turing Machine model; the Oracle is simply a set, and can only answer questions about membership in that set. All of the algorithms under consideration in Chapter 6 and in many other *Games* *with a Purpose*-style human computation systems are fixed instance algorithms.

Alternatively, the Human Oracle may be given some autonomy in deciding which problems it will solve. We call these **variable-instance algorithms**. The Human Oracle may generate and solve its own problems such as in the use of Visual Analytics systems, or it may select problem instances based on some economic model as seen in generalized task markets like Amazon Mechanical Turk. In computationally challenging problems, the Human Oracle may be able to solve only specific problem instances, such as in the crowdsourced protein folding game Fold.it [CKT+10]. General protein folding is known to be NP-Complete [BL98], and there is good reason to believe that humans cannot efficiently solve general instances of NP-Complete problems [Aar12]. However, there is high value in the discovery of lower-energy foldings for individual proteins, and so an algorithm enabling the Human Oracle to select and solve specific instances is useful.

7.1.2 Query Order

Next, we can consider whether or not the sequence of queries to the Human Oracle can be determined in advance. In an algorithm with **predetermined query order**, we claim that there exists some function:

$$f : I \rightarrow (q_1, \ldots, q_n)$$

that takes as input a problem instance I and generates a finite sequence of queries $(q_1, \ldots, q_n)$ that will be passed in order to the Human Oracle. Because the sequence can be generated *a priori*, it follows that the position of any query q_i must not depend on the Human Oracle's previous answers. In these systems, the Human Oracle cannot influence the order in which it receives and responds to specific queries. A general example of a process that uses predetermined queries is semi-supervised machine learning. In these techniques, the Human Oracle is asked to label a set of training data which is then used to infer a classification function. While the resulting classification function is dependent on how the training data is labeled, the points to be used as training data are determined in advance.

Relaxing this restriction yields a slightly more powerful querying model: algorithms whose future queries are contingent on the Human Oracle's answers to previous queries. In an algorithm with **adaptive query order**, we claim that there exists some function:

$$f : \{I, (a_1, \ldots, a_n)\} \to q_{n+1}$$

that takes as input a problem instance I as well as $(a_1, \ldots, a_n)$, the sequence of responses from the Human Oracle so far, and generates the next query (q_{n+1}). An excellent example of processes that utilize adaptive querying is *active learning*. In active learning algorithms, the Human Oracle is first asked to label some small set of training data. Based on their responses, the algorithm reclusters the data. It then selects a new subset of points about which it is least confident and submits these to the Human Oracle to label. The selection of each subsequent collection of points is directly related of the labels provided in the previous round. This process continues iteratively until some confidence threshold is met.

7.1.3 Oracle Responses

Finally, we can consider whether or not the Human Oracle is able to compute on its previous responses in order to generate a future response. Mirroring the previous dimension, some systems assume that the Human Oracle's responses to queries are **independent** of one another. An example of a real-world system where this is true is reCAPTCHA [VAMM$^+$08], a human computation system for optical character recognition (OCR). If the same human is asked to pass several reCAPTCHA instances, she will not gain any information in the process of solving a single instance that would change her answers to new instances further down the line. Thus, we can presume that each of her responses is independent. When this is the case, there is no discernible difference between asking the same Oracle n questions, or asking n different Oracles one question each. Because of this, processes leveraging Human Oracles whose responses are independent can be parallelized.

	Problem Instance		Query Order		Oracle Responses	
	Fixed	Variable	Predetermined	Adaptive	Independent	Adaptive
reCAPTCHA	✓		✓		✓	
Semi-supervised Learning	✓		✓		✓	
the ESP Game	✓		✓			✓
Active Learning	✓			✓	✓	
HBGA	✓			✓		✓
Fold.it		✓	✓			✓
Visual Analytics		✓		✓		✓

Table 7.1: Three-dimensional classification of various techniques leveraging human computation.

As mentioned in Chapter 5, it is sometimes useful to endow the Human Oracle with some amount of persistent memory regarding the query sequence. In these systems, the Human Oracle may be able to **modify its future behavior based on previous events**. In the simplest sense, this could be used to ensure that the Human Oracle does not return the same answer twice. In other scenarios, we may wish to enable computation on this sequence in order to model more nuanced adaptations, such as learning or fatigue. For example, complex systems such as visual analytics tools require that the Human Oracle be able to learn from a prior sequence of actions and responses and subsequently modify its future behavior. Note that while we continue to develop more robust models of human memory and its limits, we may elect to abstract the specifics of *how* the Human Oracle remembers and instead include the cost of accessing this memory in the query cost.

7.2 Describing the Space of Human Computation

In contrast to previous schema for comparing human computation systems which rely on nominal classification dimensions [QB11], each of the dimensions introduced here has an implicit notion of magnitude that induces an partial ordering on different algorithms and problems. For example, allowing the Human Oracle to decide which instances of a problem to solve enables us to make progress in many more challenging areas than is possible with Human Oracles that simply follow orders. In this sense, fixed instance algorithms are comparatively weaker than variable instance algorithms. Equivalently, the problems that can be solved using fixed instance algorithms are comparatively simpler than problems that cannot. Similarly, an algorithm that must predetermine its set of queries is weaker than an algorithm that can choose its next query based on previous responses, and a Human Oracle unable to use history to its advantage is not as strong as one that can.

Consider the systems and techniques listed in Table 7.1. Categorizing along these three dimensions, we see that many of our intuitions about the relative strength of these techniques are captured. For example, we see the the algorithm underlying reCAPTCHA can be computed with a less powerful Human Oracle than the ESP Game. In reCAPTCHA, the human is simply a visual decoder, and the machine takes care of compiling and aggregating the results. In the ESP Game, more responsibility is placed on each individual human to avoid performing redundant computation in order to generate a selection of unique labels. Similarly, we see that active learning requires a more powerful use of human computation than semi-supervised learning. We presume that by enabling more careful selection of user constraints or labels, the set of datasets that can be classified using active learning is broader than the set of datasets that could be classified using semi-supervised learning, given the same amount of supervision.

Because they define a partial ordering between problems, these dimensions can be used to establish preliminary *complexity classes* within the space of human computation (see Fig. 7.1). Note that we believe the class defined by the use of *independent oracle queries* to be entirely contained within the class defined by fixed problem instances. This follows from the intuition that a Human Oracle with no memory would have no mechanism for evaluating its preference for one instance over another. However, at present this is only deductive speculation; a rigorous proof is beyond the scope of this dissertation.

We suggest that these classes are complementary to those in traditional computational complexity. Indeed, we may consider these hierarchies to be orthogonal. That is, there could exist human-computer collaborative systems with all combinations of $\langle \Phi_H, \Phi_M \rangle$. For example, the ESP Game lives at the intersection of *fixed problem instance* and $O(n^2) \in \mathtt{P}$, while Fold.it exists where *predetermined query*

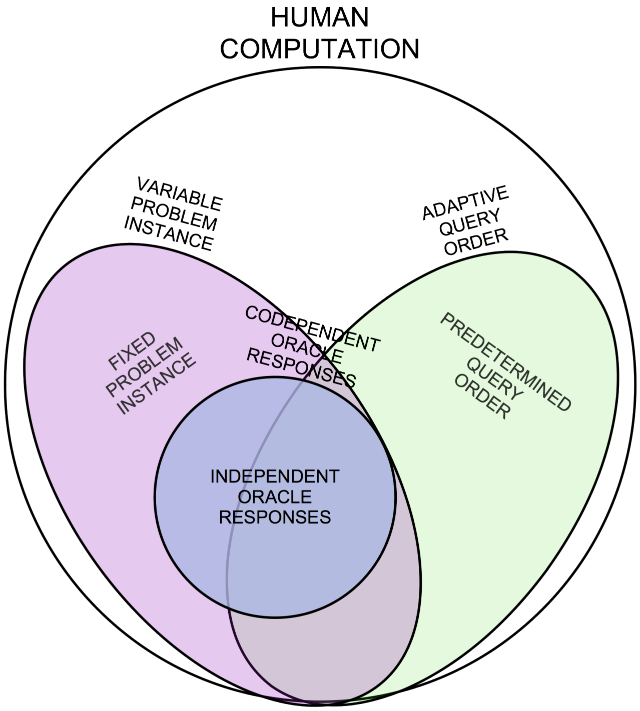

HUMAN COMPUTATION
VARIABLE PROBLEM INSTANCE
ADAPTIVE QUERY ORDER
FIXED PROBLEM INSTANCE
CODEPENDENT ORACLE RESPONSES
PREDETERMINED QUERY ORDER
INDEPENDENT ORACLE RESPONSES

Figure 7.1: Preliminary proposed hierarchy of complexity classes in human computation.

order meets `NP`.

In the following chapter, we will discuss some of the limitations of this model, as well as advocate for the continued pursuit of complexity measures for systems involving human computation. Developing more nuanced models for the use of human computation within larger computational systems will enable us to further refine these classes. Exploring the boundaries and intersections of these spaces, as well as the efficiency with which problems can be solved by various combinations of human and machine computation, is a primary goal of our future work in this area.

Chapter 8

Discussion

The affordance-based framework presented in this dissertation sheds important light on the ways in which the human mind can be harnessed as a computational resource. In addition, the Human Oracle Model provides a critical first step in quantifying the use of these resources, and helps us to better understand the intricate relationships between different problems and problem families when viewed through the lens of human computation. That said, this dissertation only just scratches the surface of this potentially rich area for future research. This model ignores some very real factors present in any system involving the variability of biological computation. In the following sections, we will discuss some of the limitations of this model, as well as motivate our continued research in this area.

8.1 Limitations of the Human Oracle Model

8.1.1 Imperfect Oracles

Under this model, there is an explicit assumption the Human Oracle will always be able to provide the correct answer at a fixed cost. In reality, humans don't work this way. Intuition and experience indicate that humans eventually get tired or bored, and as a consequence their speed and accuracy suffer. In addition, individual differences in ability and cost are absent. In the real world, not all humans are equal in their capacity to answer the questions we ask. Some are more skilled or have better training, and their expertise comes (we presume) at a higher cost.

Similar issues have arisen in the area of Active Learning, which has historically assumed a single tireless, flawless, benevolent oracle was always available to provide labels for its training data. *Proactive learning* relaxes these assumptions, adopting a decision-theoretic approach to match one of a collection of (possibly imperfect) oracles to each instance [DC08]. More recent work in the area of *multiple expert active learning* (MEAL) improves upon this model by incorporating load balancing to ensure that no worker has to shoulder an inequitable share of the burden [WSBT11]. These methods assume there exists some method to model both how hard any single problem instance is, as well as how costly and effective a given worker is.

8.1.2 Quantifying the Human Brain

This highlights another problem: as of this writing, there does not exist any reliable method for quantifying how hard a human has to work in order to accomplish a given task. Because we don't fully understand fundamental operations of the human brain or how they assemble to perform computation, it is not yet possible to calculate a precise per-operation cost. As such, at present this model cannot actually tell us *how much work* the human is doing; it only tells us how many times the human is working. When the task is comparable, such as when comparing various image labeling algorithms, this does not pose a significant problem. However, identical algorithms leveraging different affordances can have drastically different success rates.

Consider the seemingly reasonable assumption that the successful process employed the ESP Game [VAD04] in image labeling could be directly reapplied to labeling other stimuli such as audio. This was in fact the very assumption made by the designers of TagATune [LVADC07]. Despite being identical to the ESP Game in nearly every way with the exception of the human affordance, this first iteration failed miserably; people simply couldn't agree on labels for most of the input. In a second iteration, the designers found that asking the users to decide whether or not they thought they were listening to the same thing was far more successful for audio labeling than explicit matching [LVA09], although this introduces significantly more noise into the resulting dataset. This would indicate that though the human is superficially being asked to perform a similar task, the underlying information processing is fundamentally different for images versus audio.

Our lived experience might lead us to speculate that the human might be sampling their responses from a much larger search space; after all, audio lacks the same tangible, concrete concepts like *chair* and *grass* upon which we often anchor our labels for images. In addition, one might suggest that the fact that the input is continuous rather than discrete might play some role. While cognitive modeling techniques can help us to understand the interplay between stimulus and response, existing architectures are not designed to determine the "complexity" of the model itself. Though the number of nodes and interactions in two models may be different, we do not have evidence to support (or refute) that this relates to problem hardness. While unobtrusive brain sensing methods are currently under development and have shown promise in detecting mental workload [HSG+09] and task difficulty [GSH+09], the information revealed is not yet refined to a per-operation granularity. Thus, from a cognitive science perspective, there is presently no mechanism for quantifying the computation performed by the human brain. In order to form a complete model of human computation, it is critical that we continue to develop more nuanced models the human brain and to incorporate these models into the evaluation of algorithmic complexity and performance in human-machine collaborative systems.

8.2 Why Develop Complexity Measures for Human Computation?

To date, human computation has concerned itself almost entirely with questions of *computability.* That is, can using human computation make it possible to solve problems which are otherwise thought to be unsolvable? Using experiential knowledge regarding the kinds of processing that we humans are "better" at, such as recognizing objects and speaking naturally, we build systems that capitalize on these skills and offer them as constructive proof: tangible evidence that the problems are in fact solvable using human computation, even when other methods have failed.

In other areas of the computational sciences, theoretical arguments paved the way for the designs that made provably correct solutions tractable. In human computation, the development of real-world implementations has far outpaced the development of theoretical measures. Many of these implementations have demonstrated unparalleled success at previously intractable problems. However, in the absence of a rigorous theory in which to ground the development of new algorithms, researchers must rely on intuition and some deeply-rooted assumptions about the differences between human and machine computation in order to design new systems.

There is an implicit assumption that the use of human processing power in such systems will be judicious. After all, we have observed that there is a point at which human "processors" will simply refuse to perform any more computation. Much effort has been put into learning how to best incentivize human processors to perform computation through financial [MW10a, SM11] and social [CH11] mechanisms. *Games with a Purpose* try to make participation more entertaining for the human [VA06], thereby increasing their willingness to contribute. However, to date there has been little progress toward measures for describing how the computational tasks are being allocated, and few mechanisms have been developed for comparing the algorithmic processes underlying human computation systems independent of the details of their implementation.

Computational complexity theory takes the study of solvable problems to a deeper level by asking about the **resources** needed to solve them in terms of time and memory. It enables us to ask questions that get at the fundamental nature of the problem and how me might go about solving it more effectively. Does randomization help? Can the process be sped up using parallelism? Are approximations easier? By understanding the resources required, we can begin to group algorithms and problems into *complexity classes*, with members of the same class requiring similar kinds or quantities of resources. It also enables us to investigate the effect limiting these resources has on the classes of tasks that can still be solved.

At a low level, there is significant interest in establishing concrete lower bounds on the complexity of computational problems. That is, what is the *minimum amount of work* that must be done in order to guarantee the solution is correct? Most research in areas such as circuit complexity fall this category. At a higher level, complexity theory also explores the connections between different computational problems and processes, such as in NP-completeness. This kind of analysis can yield fruitful comparisons that deepen our understanding of the nature of a problem space, even when we are unable to provide absolute statements regarding the individual problems or notions.

We argue that developing these analytical tools, establishing bounds on our algorithmic processes and deepening our understanding of the relationships between the problems we're trying to solve are of critical importance to the study and design of systems involving human computation. Drawing parallels at the algorithmic level rather than at the implementation level will enable us to compare solutions more effectively than using simple A-B testing. In human computation as with other branches of computational science, identifying areas where existing algorithms are redundant or inefficient will enable us to design more efficient algorithms in the future. In addition, reporting bounds on the complexity of human computation algorithms along with the observed performance of the system would improve study reproducibility, as well as help isolate the effects of interface design and other implementation details.

8.3 Broader Impact

The importance of understanding human computation as part of a larger computational complexity system is not limited to improving algorithm design. Augmenting computational complexity models to incorporate human computation can expand our understanding of what can be computed, as did the development of probabilistic and parallel computation. Indeed, this is a forte of the field of computational complexity:

> "The mark of a good scientific field is its ability to adapt to new ideas and new technologies. Computational complexity reaches this ideal. As we have developed new ideas... the complexity community has not thrown out the previous research, rather they have modified the existing models to fit these new ideas and have shown how to connect [their] power... to our already rich theory." [FH03]

The development of complexity measures for human computation may play a significant role in the broader adoption of human computational methods. Robust models of *how humans fit* into the grand scheme of computational tools is essential to promoting wider understanding of human effort as a legitimate and measurable computational resource.

Chapter 9

Conclusion

In this dissertation, we have explored the complementary nature of human and machine computation through three interrelated research thrusts. First, we described the development of two successful visual analytics systems for exploring complex behavioral simulations in political science. Through in situ expert analysis, we demonstrated the utility of these human-computer collaborative analytics systems and their superior performance relative to preexisting manual practices. Second, we presented a framework for comparing human-computer collaborative systems according to relative strengths of human and machine collaborators on which they rely. Finally, we introduced the Human Oracle Model as a method for characterizing and quantifying the *use of human processing power as part of an algorithmic process.* We demonstrated the utility of this model for comparing and analyzing several well-known human computation systems for image labeling and describe how this model can be used to characterize the space of human computation. In closing, we discussed the model's limitations and its potential for broader impact. Through this research, we hope to form a more holistic picture of the interrelationship between human and machine computation, and to develop a robust theoretical model for the analysis of systems involving their collaboration.

Bibliography

[AAR$^+$09] G. Andrienko, N. Andrienko, S. Rinzivillo, M. Nanni, D. Pedreschi, and F. Giannotti. Interactive visual clustering of large collections of trajectories. In *Visual Analytics Science and Technology (VAST) 2009, Symposium on*, pages 3–10. IEEE, 2009.

[Aar12] Scott Aaronson. *Why Philosophers Should Care About Computational Complexity.* MIT Press, 2012.

[Ada11] E. Adar. Why i hate mechanical turk research (and workshops). In *Proc. of the 29th SIGCHI Conf. on Human factors in computing systems: Workshop on Crowdsourcing and Human Computation.* ACM, 2011.

[AED$^+$02] R.L. Axtell, J.M. Epstein, J.S. Dean, G.J. Gumerman, A.C. Swedlund, J. Harburger, S. Chakravarty, R. Hammond, J. Parker, and M. Parker. Population growth and collapse in a multiagent model of the Kayenta Anasazi in Long House Valley. *Proceedings of the National Academy of Sciences of the United States of America*, 99(Suppl 3):7275, 2002.

[AHG11] B. Alcorn, A. Hicken, and M. Garces. VirThai: A PS-I Implemented Agent-Based Model of Thailand in 2010 as a Predictive and Analytic Tool. 2011.

[AIP11] J. Attenberg, P.G. Ipeirotis, and F. Provost. Beat the machine: Challenging workers to find the unknown unknowns. In *Workshops at the Twenty-Fifth AAAI Conf. on Artificial Intelligence*, 2011.

[Ama96] T.M. Amabile. *Creativity in context: Update to "the social psychology of creativity"*. Westview press, 1996.

[AWD11] A. Anand, L. Wilkinson, and T.N. Dang. Using random projections to identify class-separating variables in high-dimensional spaces. In *Visual Analytics Science and Technology (VAST), Conf. on*, pages 263–264. IEEE, 2011.

[Axe97] R. Axelrod. *The Complexity of Cooperation*, volume 159. Princeton University Press, 1997.

[BB00] R. Bhavnani and D. Backer. Localized Ethnic Conflict and Genocide. *Journal of Conflict Resolution*, 44(3):283, 2000.

[BC94] D. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *AAAI-94 workshop on knowledge discovery in databases*, pages 229–248, 1994.

[BJJ+10] J.P. Bigham, C. Jayant, H. Ji, G. Little, A. Miller, R.C. Miller, R. Miller, A. Tatarowicz, B. White, S. White, et al. Vizwiz: nearly real-time answers to visual questions. In *Proc. of the 23rd symposium on User interface software and technology*, pages 333–342. ACM, 2010.

[BJVH11] I. Bowman, S.H. Joshi, and J.D. Van Horn. Query-based coordinated multiple views with feature similarity space for visual analysis of mri repositories. In *Visual Analytics Science and Technology (VAST), Conf. on*, pages 267–268. IEEE, 2011.

[BKAA11] F.A. Boujarwah, J.G. Kim, G.D. Abowd, and R.I. Arriaga. Developing scripts to teach social skills: Can the crowd assist the author? In *Workshops at the Twenty-Fifth AAAI Conf. on Artificial Intelligence*, 2011.

[BKW+11] David Bermbach, Robert Kern, Pascal Wichmann, Sandra Rath, Christian Zirpins, David Bermbach, and Christian Zirpins. An extend-able toolkit for managing quality of human-based electronic services. *Human Computation*, 11:11, 2011.

[BL98] B. Berger and T. Leighton. Protein folding in the hydrophobic-hydrophilic (hp) model is np-complete. *Journal of Computational Biology*, 5(1):27–40, 1998.

[BL10] E. Bertini and D. Lalanne. Investigating and reflecting on the integration of automatic data analysis and visualization in knowledge discovery. *ACM SIGKDD Explorations Newsletter*, 11(2):9–18, 2010.

[BLM+10] M.S. Bernstein, G. Little, R.C. Miller, B. Hartmann, M.S. Ackerman, D.R. Karger, D. Crowell, and K. Panovich. Soylent: a word processor with a crowd inside. In *Proc. of the 23rd symposium on User interface software and technology*, pages 313–322. ACM, 2010.

[BOTL09] L. Barrington, D. O'Malley, D. Turnbull, and G. Lanckriet. User-centered design of a social game to tag music. In *Proc. of the ACM SIGKDD Workshop on Human Computation*, pages 7–10. ACM, 2009.

[BRB+09] M. Bell, S. Reeves, B. Brown, S. Sherwood, D. MacMillan, J. Ferguson, and M. Chalmers. Eyespy: supporting navigation through play. In *Proc. of the 27th SIGCHI Conf. on Human factors in computing systems*, pages 123–132. ACM, 2009.

[CAB+11] Y. Chen, J. Alsakran, S. Barlowe, J. Yang, and Y. Zhao. Supporting effective common ground construction in asynchronous collaborative visual analytics. In *Visual Analytics Science and Technology (VAST), Conf. on*, pages 101–110. IEEE, 2011.

[CBY10] Y. Chen, S. Barlowe, and J. Yang. Click2annotate: Automated insight externalization with rich semantics. In *Visual Analytics Science and Technology (VAST), Symposium on*, pages 155–162. IEEE, 2010.

[CC12] R.J. Crouser and R. Chang. An affordance-based framework for human computation and human-computer collaboration. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2859–2868, 2012.

[CCH11] Tao-Hsuan Chang, Cheng-wei Chan, and Jane Yung-jen Hsu. Musweeper: An extensive game for collecting mutual exclusions. In *Human Computation*, 2011.

[CCXC09] K. Chung, C. Chiu, X. Xiao, and P.Y.P. Chi. Stress outsourced: a haptic social network via crowdsourcing. In *Proc. of the 27th SIGCHI Conf. on Human factors in computing systems*, pages 2439–2448. ACM, 2009.

[CEH+09] M. Chen, D. Ebert, H. Hagen, R.S. Laramee, R. Van Liere, K.L. Ma, W. Ribarsky, G. Scheuermann, and D. Silver. Data, information, and knowledge in visualization. *Computer Graphics and Applications, IEEE*, 29(1):12–19, 2009.

[CH11] Dana Chandler and John Joseph Horton. Labor allocation in paid crowdsourcing: Experimental evidence on positioning, nudges and prices. In *Human Computation*, 2011.

[CKJC12] R.J. Crouser, D.E. Kee, D.H. Jeong, and R. Chang. Two visualization tools for analyzing agent-based simulations in political science. *IEEE Computer Graphics and Applications*, pages 67–77, 2012.

[CKT+10] S. Cooper, F. Khatib, A. Treuille, J. Barbero, J. Lee, M. Beenen, A. Leaver-Fay, D. Baker, Z. Popovic, et al. Predicting protein structures with a multiplayer online game. *Nature*, 466(7307):756–760, 2010.

[CKY09] Kam Tong Chan, Irwin King, and Man-Ching Yuen. Mathematical modeling of social games. In *Computational Science and Engineering,* *2009. CSE'09. International Conference on*, volume 4, pages 1205–1210. IEEE, 2009.

[CLG$^+$08] Remco Chang, Alvin Lee, Mohammad Ghoniem, Robert Kosara, William Ribarsky, Jing Yang, Evan Suma, Caroline Ziemkiewicz, Daniel Kern, and Agus Sudjianto. Scalable and interactive visual analysis of financial wire transactions for fraud detection. *Information visualization*, 7(1):63–76, 2008.

[CLKP10] J. Choo, H. Lee, J. Kihm, and H. Park. ivisclassifier: An interactive visual analytics system for classification based on supervised dimension reduction. In *Visual Analytics Science and Technology (VAST), Symposium on*, pages 27–34. IEEE, 2010.

[CLZ11] Y. Chen, B. Liem, and H. Zhang. An iterative dual pathway structure for speech-to-text transcription. In *Human Computation: Papers from the AAAI Workshop (WS-11-11). San Francisco, CA*, 2011.

[COB10] S. Chernova, J. Orkin, and C. Breazeal. Crowdsourcing hri through online multiplayer games. In *Proc. Dialog with Robots: AAAI fall symposium*, 2010.

[CPK09] Jon Chamberlain, Massimo Poesio, and Udo Kruschwitz. A demonstration of human computation using the phrase detectives annotation game. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pages 23–24. ACM, 2009.

[CS11] Otto Chrons and Sami Sundell. Digitalkoot: Making old archives accessible using crowdsourcing. In *Human Computation*, 2011.

[Dai11] Daniel S Weld Mausam Peng Dai. Human intelligence needs artificial intelligence. 2011.

[Dav10] Colin J Davis. The spatial coding model of visual word identification. *Psychological Review*, 117(3):713, 2010.

[DC08] Pinar Donmez and Jaime G Carbonell. Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 619–628. ACM, 2008.

[Dou04] P. Dourish. *Where the action is: the foundations of embodied interaction.* The MIT Press, 2004.

[DSC10] Pedro Demasi, Jayme L Szwarcfiter, and Adriano JO Cruz. A theoretical framework to formalize agi-hard problems. In *The Third Conference on Artificial General Intelligence, Lugano, Switzerland*, 2010.

[DSG07] B.N. Da Silva and A.C.B. Garcia. Ka-captcha: An opportunity for knowledge acquisition on the web. In *Proc. of the national Conf. on Artificial Intelligence*, volume 22, page 1322, 2007.

[DW02] S.W.A. Dekker and D.D. Woods. Maba-maba or abracadabra? progress on human-automation coordination. *Cognition, Technology & Work*, 4(4):240–244, 2002.

[DWCR11] W. Dou, X. Wang, R. Chang, and W. Ribarsky. Paralleltopics: A probabilistic approach to exploring document collections. *Visual Analytics Science and Technology (VAST), Conf. on*, 2011.

[FA98] G. Ferguson and J.F. Allen. Trips: An integrated intelligent problem-solving assistant. In *Proc. of the National Conf. on Artificial Intelligence*, pages 567–573. JOHN WILEY & SONS LTD, 1998.

[FH03] Lance Fortnow and Steve Homer. A short history of computational complexity. *Bulletin of the EATCS*, 80:95–133, 2003.

[FHI11] Siamak Faradani, Bj¨orn Hartmann, and Panagiotis G Ipeirotis. What's the right price? pricing tasks for finishing on time. In *Human Computation*, 2011.

[Fit51] P.M. Fitts. Human engineering for an effective air-navigation and traffic-control system. 1951.

[FWG09] R. Fuchs, J. Waser, and M.E. Groller. Visual human+ machine learning. *Visualization and Computer Graphics, Transactions on*, 15(6):1327–1334, 2009.

[FZ07] H. Fastl and E. Zwicker. *Psychoacoustics: facts and models*, volume 22. Springer-Verlag New York Inc, 2007.

[Gav91] W.W. Gaver. Technology affordances. In *Proc. of the SIGCHI Conf. on Human factors in computing systems: Reaching through technology*, pages 79–84. ACM, 1991.

[Gib77] JJ Gibson. The theory of affordances. *Perceiving, acting, and knowing*, pages 67–82, 1977.

[Gib86] J.J. Gibson. *The ecological approach to visual perception.* Lawrence Erlbaum, 1986.

[Gri11] David Alan Grier. Error identification and correction in human computation: Lessons from the wpa. In *Human Computation*, 2011.

[GRM10] S. Garg, IV Ramakrishnan, and K. Mueller. A visual analytics approach to model learning. In *Visual Analytics Science and Technology (VAST), Symposium on*, pages 67–74. IEEE, 2010.

[GSH$^+$09] Audrey Girouard, Erin Treacy Solovey, Leanne M Hirshfield, Krysta Chauncey, Angelo Sassaroli, Sergio Fantini, and Robert JK Jacob. Distinguishing difficulty levels with non-invasive brain activity measurements. In *Human-Computer Interaction–INTERACT 2009*, pages 440–452. Springer, 2009.

[GVGH12] Yotam Gingold, Etienne Vouga, Eitan Grinspun, and Haym Hirsh. Diamonds from the rough: Improving drawing, painting, and singing via crowdsourcing. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[HB12] Harry Halpin and Roi Blanco. Machine-learning for spammer detection in crowd-sourcing. In *Workshop on Human Computation at AAAI, Technical Report WS-12-08*, pages 85–86, 2012.

[HBRK11] C. Hu, B.B. Bederson, P. Resnik, and Y. Kronrod. Monotrans2: A new human computation system to support monolingual translation. In *Proc. of the 29th SIGCHI Conf. on Human factors in computing systems*, pages 1133–1136. ACM, 2011.

[HCL$^+$09] C.J. Ho, T.H. Chang, J.C. Lee, J.Y. Hsu, and K.T. Chen. Kisskissban: a competitive human computation game for image annotation. In *Proc. of the SIGKDD Workshop on Human Computation*, pages 11–14. ACM, 2009.

[Hor99] E. Horvitz. Principles of mixed-initiative user interfaces. In *Proc. of the SIGCHI Conf. on Human factors in computing systems: the CHI is the limit*, pages 159–166. ACM, 1999.

[HSG$^+$09] Leanne M Hirshfield, Erin Treacy Solovey, Audrey Girouard, James Kebinger, Robert JK Jacob, Angelo Sassaroli, and Sergio Fantini. Brain measurement for usability testing and adaptive interfaces: an example of uncovering syntactic workload with functional near infrared spectroscopy. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2185–2194. ACM, 2009.

[HZVvdS12] Chien-Ju Ho, Yu Zhang, J Vaughan, and Mihaela van der Schaar. Towards social norm design for crowdsourcing markets. In *AAAI Workshops*, 2012.

[IMI$^+$10] S. Ingram, T. Munzner, V. Irvine, M. Tory, S. Bergner, and T. Mller. Dimstiller: Workflows for dimensional analysis and reduction. In *Vi-*

sual Analytics Science and Technology (VAST), Symposium on, pages 3–10. IEEE, 2010.

[IMO08] S. Ingram, T. Munzner, and M. Olano. Glimmer: Multilevel MDS on the GPU. *IEEE Transactions on Visualization and Computer Graphics*, pages 249–261, 2008.

[IPW10] Panagiotis G Ipeirotis, Foster Provost, and Jing Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 64–67. ACM, 2010.

[IV11] C.Y. Ip and A. Varshney. Saliency-assisted navigation of very large landscape images. *Visualization and Computer Graphics, Transactions on*, 17(12):1737–1746, 2011.

[JL11] Hyun Joon Jung and Matthew Lease. Improving consensus accuracy via z-score and weighted voting. In *Human Computation*, 2011.

[JL12] Hyun Joon Jung and Matthew Lease. Improving quality of crowdsourced labels via probabilistic matrix factorization. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[Jor63] N. Jordan. Allocation of functions between man and machines in automated systems. *Journal of applied psychology*, 47(3):161, 1963.

[KBBN09] Neeraj Kumar, Alexander C Berg, Peter N Belhumeur, and Shree K Nayar. Attribute and simile classifiers for face verification. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 365–372. IEEE, 2009.

[KBGE09] S. Koch, H. Bosch, M. Giereth, and T. Ertl. Iterative integration of visual insights during patent search and analysis. In *Visual Analytics Science and Technology (VAST), Symposium on*, pages 203–210. IEEE, 2009.

[KCD$^+$09] N. Kadivar, V. Chen, D. Dunsmuir, E. Lee, C. Qian, J. Dill, C. Shaw, and R. Woodbury. Capturing and supporting the analysis process. In *Visual Analytics Science and Technology (VAST), Symposium on*, pages 131–138. IEEE, 2009.

[KCH11] Anand P Kulkarni, Matthew Can, and Bjoern Hartmann. Turkomatic: automatic recursive task and workflow design for mechanical turk. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, pages 2053–2058. ACM, 2011.

[KDHL08] Robert Kosara, Fritz Drury, Lars Erik Holmquist, and David H Laidlaw. Visualization criticism. *Computer Graphics and Applications, IEEE*, 28(3):13–15, 2008.

[KJB12] A Kumaran, Sujay Kumar Jauhar, and Sumit Basu. Doodling: A gaming paradigm for generating language data. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[KKEM10] Daniel A Keim, Jörn Kohlhammer, Geoffrey Ellis, and Florian Mansmann. *Mastering The Information Age-Solving Problems with Visual Analytics*. Florian Mansmann, 2010.

[KLC$^+$09] Y. Kuo, J.C. Lee, K. Chiang, R. Wang, E. Shen, C. Chan, and J.Y. Hsu. Community-based game design: experiments on social games for commonsense data collection. In *Proc. of the ACM SIGKDD Workshop on Human Computation*, pages 15–22. ACM, 2009.

[KPHH11] S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer. Wrangler: Interactive visual specification of data transformation scripts. In *Proc. of the 2011 Conf. on Human factors in computing systems*, pages 3363–3372. ACM, 2011.

[Kul11] A. Kulkarni. The complexity of crowdsourcing: Theoretical problems in human computation. In *CHI Workshop on Crowdsourcing and Human Computation*, 2011.

[LA11] Edith Law and Luis von Ahn. Human computation. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 5(3):1–121, 2011.

[LAGR10] I.S. Lustick, B. Alcorn, M. Garces, and A. Ruvinsky. From Theory to Simulation: The Dynamic Political Hierarchy in Country Virtualization Models. In *American Political Science Association (APSA) 2010 Annual Meeting*, 2010. Available at SSRN: http://ssrn.com/abstract=1642003.

[LALUR12] Boyang Li, Darren Scott Appling, Stephen Lee-Urban, and Mark O Riedl. Learning sociocultural knowledge via crowdsourced examples. In *Proc. of the 4th AAAI Workshop on Human Computation*, 2012.

[Lea11] Matthew Lease. On quality control and machine learning in crowdsourcing. In *Human Computation*, 2011.

[LKWL07] J. Lin, E. Keogh, L. Wei, and S. Lonardi. Experiencing SAX: a novel symbolic representation of time series. *Data Mining and Knowledge Discovery*, 15(2):107–144, 2007.

[LLKM10] Z. Liu, B. Lee, S. Kandula, and R. Mahajan. Netclinic: Interactive visualization to enhance automated fault diagnosis in enterprise networks. In *Visual Analytics Science and Technology (VAST), Symposium on*, pages 131–138. IEEE, 2010.

[LMSR08] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[LSD$^+$10] H.R. Lipford, F. Stukes, W. Dou, M.E. Hawkins, and R. Chang. Helping users recall their reasoning process. In *Visual Analytics Science and Technology (VAST), Symposium on*, pages 187–194. IEEE, 2010.

[LST07] H. Lieberman, D. Smith, and A. Teeters. Common consensus: a web-based game for collecting commonsense goals. In *ACM Workshop on Common Sense for Intelligent Interfaces*, 2007.

[Lus02] I. Lustick. PS-I: A user-friendly agent-based modeling platform for testing theories of political identity and political stability. *Journal of Artificial Societies and Social Simulation*, 5(3), 2002.

[LVA09] E. Law and L. Von Ahn. Input-agreement: a new mechanism for collecting data using human computation games. In *Proc. of the 27th SIGCHI Conf. on Human factors in computing systems*, pages 1197–1206, 2009.

[LVADC07] E.L.M. Law, L. Von Ahn, R. Dannenberg, and M. Crawford. Tagatune: A game for music and sound annotation. *Proc. of ISMIR (Vienna, Austria)*, 2007.

[LW$^+$12] Christopher H Lin, Daniel Weld, et al. Crowdsourcing control: Moving beyond multiple choice. *arXiv preprint arXiv:1210.4870*, 2012.

[M$^+$98] G.E. Moore et al. Cramming more components onto integrated circuits. *Proc. of the IEEE*, 86(1):82–85, 1998.

[MCQG09] H. Ma, R. Chandrasekar, C. Quirk, and A. Gupta. Page hunt: using human computation games to improve web search. In *Proc. of the ACM SIGKDD Workshop on Human Computation*, pages 27–28. ACM, 2009.

[ME08] M.I. Mandel and D.P.W. Ellis. A web-based game for collecting music metadata. *Journal of New Music Research*, 37(2):151–165, 2008.

[MPC12] Andrew Mao, Ariel D Procaccia, and Yiling Chen. Social choice for human computation. *Proc. of 4th HCOMP*, 2012.

[MS12] Subhransu Maji and Gregory Shakhnarovich. Part annotations via pairwise correspondence. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[MvGW11] MA Migut, JC van Gemert, and M. Worring. Interactive decision making using dissimilarity to visually represented prototypes. In *Visual Analytics Science and Technology (VAST), Conf. on*, pages 141–149. IEEE, 2011.

[MW10a] Winter Mason and Duncan J Watts. Financial incentives and the performance of crowds. *ACM SigKDD Explorations Newsletter*, 11(2):100–108, 2010.

[MW10b] M. Migut and M. Worring. Visual exploration of classification models for risk assessment. In *Visual Analytics Science and Technology (VAST), Symposium on*, pages 11–18. IEEE, 2010.

[NGR$^+$11] Prayag Narula, Philipp Gutheim, David Rolnitzky, Anand Kulkarni, and Bjoern Hartmann. Mobileworks: A mobile crowdsourcing platform for workers at the bottom of the pyramid. In *Human Computation*, 2011.

[NO03] Kenneth A Norman and Randall C O'Reilly. Modeling hippocampal and neocortical contributions to recognition memory: a complementary-learning-systems approach. *Psychological review*, 110(4):611, 2003.

[Nor02] D.A. Norman. *The design of everyday things*. Basic books, 2002.

[OSL$^+$11] David Oleson, Alexander Sorokin, Greg P Laughlin, Vaughn Hester, John Le, and Lukas Biewald. Programmatic gold: Targeted and scalable quality assurance in crowdsourcing. *Human computation*, 2011.

[PG11] Devi Parikh and Kristen Grauman. Interactively building a discriminative vocabulary of nameable attributes. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1681–1688. IEEE, 2011.

[Pri85] H.E. Price. The allocation of functions in systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 27(1):33–45, 1985.

[QB09] Alexander J Quinn and Benjamin B Bederson. A taxonomy of distributed human computation. *Human-Computer Interaction Lab Tech Report, University of Maryland*, 2009.

[QB11] A.J. Quinn and B.B. Bederson. Human computation: a survey and taxonomy of a growing field. In *Proc. of the 29th SIGCHI Conf. on Human factors in computing systems*, pages 1403–1412. ACM, 2011.

[RBBV11] M. Ruiz, A. Bardera, I. Boada, and I. Viola. Automatic transfer functions based on informational divergence. *Visualization and Computer Graphics, Transactions on*, 17(12):1932–1941, 2011.

[RJ99] J. Rickel and W.L. Johnson. Animated agents for procedural training in virtual reality: Perception, cognition, and motor control. *Applied Artificial Intelligence*, 13(4-5):343–382, 1999.

[RTMF08] B.C. Russell, A. Torralba, K.P. Murphy, and W.T. Freeman. Labelme: a database and web-based tool for image annotation. *Int'l journal of computer vision*, 77(1):157–173, 2008.

[Rub15] Edgar Rubin. *Synsoplevede figurer: studier i psykologisk analyse. 1. del.* Gyldendalske Boghandel, Nordisk Forlag, 1915.

[Run07] M.A. Runco. *Creativity: Theories and themes: Research, development, and practice.* Academic Press, 2007.

[RV12] Gireeja Ranade and Lav R Varshney. To crowdsource or not to crowdsource? In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[SCVAT10] Nitin Seemakurty, Jonathan Chu, Luis Von Ahn, and Anthony Tomasic. Word sense disambiguation via human computation. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pages 60–63. ACM, 2010.

[SDFF12] Hao Su, Jia Deng, and Li Fei-Fei. Crowdsourcing annotations for visual object detection. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[Seb02] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.

[SGL09] Y.B. Shrinivasan, D. Gotzy, and J. Lu. Connecting the dots in visual analysis. In *Visual Analytics Science and Technology (VAST), Symposium on*, pages 123–130. IEEE, 2009.

[SGM11] Siddharth Suri, Daniel G Goldstein, and Winter A Mason. Honesty in an online labor market. In *Human Computation*, 2011.

[She00] T.B. Sheridan. Function allocation: algorithm, alchemy or apostasy? *Int'l Journal of Human-Computer Studies*, 52(2):203–216, 2000.

[Shn96] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proc.., IEEE Symposium on*, pages 336–343. IEEE, 1996.

[SM05] P. Shah and A. Miyake. *The Cambridge handbook of visuospatial thinking*. Cambridge Univ Pr, 2005.

[SM11] Yaron Singer and Manas Mittal. Pricing tasks in online labor markets. In *Human Computation*, 2011.

[SPRK03] A. Srbljinovic, D. Penzar, P. Rodik, and K. Kardov. An agent-based model of ethnic mobilisation. *Journal of Artificial Societies and Social Simulation*, 6(1):1, 2003.

[SRIT10] M Six Silberman, Joel Ross, Lilly Irani, and Bill Tomlinson. Sellers problems in human computation markets. In *Proceedings of the acm sigkdd workshop on human computation*, pages 18–21. ACM, 2010.

[SRL11] Yu-An Sun, Shourya Roy, and Greg Little. Beyond independent agreement: A tournament selection approach for quality assurance of human computation tasks. In *Human Computation*, 2011.

[SRSJ11] Ruben Stranders, Sarvapali D Ramchurn, Bing Shi, and Nicholas R Jennings. Collabmap: Augmenting maps using the wisdom of crowds. In *Human Computation*, 2011.

[SSJKF09] C.A. Steed, JE Swan, TJ Jankun-Kelly, and P.J. Fitzpatrick. Guided analysis of hurricane trends using statistical processes integrated with interactive parallel coordinates. In *Visual Analytics Science and Technology (VAST), Symposium on*, pages 19–26. IEEE, 2009.

[ST08] P. Shenoy and D.S. Tan. Human-aided computing: Utilizing implicit human processing to classify images. In *Proc. of the 26th SIGCHI Conf. on Human factors in computing systems*, pages 845–854. ACM, 2008.

[TAE$^+$09] A. Tatu, G. Albuquerque, M. Eisemann, J. Schneidewind, H. Theisel, M. Magnork, and D. Keim. Combining automated analysis and visualization techniques for effective exploration of high-dimensional data. In *Visual Analytics Science and Technology (VAST), Symposium on*, pages 59–66. IEEE, 2009.

[TC05] J.J. Thomas and K.A. Cook. *Illuminating the path: The research and development agenda for visual analytics*, volume 54. IEEE, 2005.

[Ter95] L.G. Terveen. Overview of human-computer collaboration. *Knowledge-Based Systems*, 8(2-3):67–81, 1995.

[TSK11] Y. Tanaka, Y. Sakamoto, and T. Kusumi. Conceptual combination versus critical combination: Devising creative solutions using the sequential application of crowds. In *Proc. of the 33rd Conf. of the Cognitive Science Society*, 2011.

[Tur38] Alan Mathison Turing. *Systems of logic based on ordinals: a dissertation*. PhD thesis, Cambridge, 1938.

[VA05] Luis Von Ahn. *Human Computation*. PhD thesis, Carnegie Mellon University, 2005.

[VA06] Luis Von Ahn. Games with a purpose. *Computer*, 39(6):92–94, 2006.

[VAD04] L. Von Ahn and L. Dabbish. Labeling images with a computer game. In *Proc. of the 22nd SIGCHI Conf. on Human factors in computing systems*, pages 319–326. ACM, 2004.

[VAGK$^+$06] L. Von Ahn, S. Ginosar, M. Kedia, R. Liu, and M. Blum. Improving accessibility of the web with a computer game. In *Proc. of the 24th SIGCHI Conf. on Human factors in computing systems*, pages 79–82. ACM, 2006.

[VAKB06] L. Von Ahn, M. Kedia, and M. Blum. Verbosity: a game for collecting common-sense facts. In *Proc. of the 24th SIGCHI Conf. on Human factors in computing systems*, pages 75–78. ACM, 2006.

[VALB06] L. Von Ahn, R. Liu, and M. Blum. Peekaboom: a game for locating objects in images. In *Proc. of the 24th SIGCHI Conf. on Human factors in computing systems*, pages 55–64. ACM, 2006.

[VAMM$^+$08] L. Von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum. recaptcha: Human-based character recognition via web security measures. *Science*, 321(5895):1465–1468, 2008.

[VP99] R.E. Valdés-Pérez. Principles of human-computer collaboration for knowledge discovery in science. *Artificial Intelligence*, 107(2):335–346, 1999.

[Vyg62] L.S. Vygotsky. Thought and word. 1962.

[WBPB10] Peter Welinder, Steve Branson, Pietro Perona, and Serge J Belongie. The multidimensional wisdom of crowds. In *Advances in Neural Information Processing Systems*, pages 2424–2432, 2010.

[Win08] J.M. Wing. Five deep questions in computing. *Communications of the ACM*, 51(1):58–60, 2008.

[WSBT11] Byron C Wallace, Kevin Small, Carla E Brodley, and Thomas A Trikalinos. Who should label what? instance allocation in multiple expert active learning. In *SDM*, pages 176–187, 2011.

[WY10] Jun Wang and Bei Yu. Sentence recall game: a novel tool for collecting data to discover language usage patterns. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pages 56–59. ACM, 2010.

[WY12] Jun Wang and Bei Yu. Collecting representative pictures for words: A human computation approach based on draw something game. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[XKS92] Lei Xu, Adam Krzyzak, and Ching Y Suen. Methods of combining multiple classifiers and their applications to handwriting recognition. *Systems, Man and Cybernetics, IEEE Transactions on*, 22(3):418–435, 1992.

[Yam11] Roman V Yampolskiy. Ai-complete captchas as zero knowledge proofs of access to an artificially intelligent system. *ISRN Artificial Intelligence*, 2012, 2011.

[Yam12] Roman V Yampolskiy. Ai-complete, ai-hard, or ai-easy–classification of problems in ai. In *Midwest Artificial Intelligence and Cognitive Science Conference*, page 94. Citeseer, 2012.

[Yam13] Roman V Yampolskiy. Turing test as a defining feature of ai-completeness. In *Artificial Intelligence, Evolutionary Computing and Metaheuristics*, pages 3–17. Springer, 2013.

[YCK09] M.C. Yuen, L.J. Chen, and I. King. A survey of human computation systems. In *Computational Science and Engineering, Int'l Conf. on*, volume 4, pages 723–728. IEEE, 2009.

[YKG10] T. Yan, V. Kumar, and D. Ganesan. Crowdsearch: exploiting crowds for accurate real-time image search on mobile phones. In *Proc. of the 8th Int'l Conf. on Mobile systems, applications, and services*, pages 77–90. ACM, 2010.

[YN11] L. Yu and J.V. Nickerson. Cooks or cobblers?: crowd creativity through combination. In *Proc. of the 29th SIGCHI Conf. on Human factors in computing systems*, pages 1393–1402. ACM, 2011.

[YZG+08] Yang Yang, Bin B Zhu, Rui Guo, Linjun Yang, Shipeng Li, and Nenghai Yu. A comprehensive human computation framework: with application to image labeling. In *Proceedings of the 16th ACM international conference on Multimedia*, pages 479–488. ACM, 2008.

[ZAM11] Z. Zheng, N. Ahmed, and K. Mueller. iview: A feature clustering framework for suggesting informative views in volume visualization. *Visualization and Computer Graphics, Transactions on*, 17(12):1959–1968, 2011.

[ZSS11] Liyue Zhao, Gita Sukthankar, and Rahul Sukthankar. Robust active learning using crowdsourced annotations for activity recognition. In *Human Computation*, 2011.