Title
Geometric Models for Collaborative Search and Filtering

Author
Bitton, Ephrat

Publication Date
2011

Peer reviewed|Thesis/dissertation

Geometric Models for Collaborative Search and Filtering

by

Ephrat Bitton

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering – Industrial Engineering and Operations Research
and the Designated Emphasis

in

New Media

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Ken Y. Goldberg, Co-Chair
Professor Dorit Hochbaum, Co-Chair
Professor Coye Cheshire
Professor Ilan Adler

Spring 2011

Geometric Models for Collaborative Search and Filtering

Abstract

Geometric Models for Collaborative Search and Filtering

by

Ephrat Bitton

Doctor of Philosophy in Engineering – Industrial Engineering and Operations Research

with a Designated Emphasis in New Media

University of California, Berkeley

Professor Ken Y. Goldberg, Co-Chair

Professor Dorit Hochbaum, Co-Chair

This dissertation explores the use of geometric and graphical models for a variety of information search and filtering applications. These models serve to provide an intuitive understanding of the problem domains and as well as computational efficiencies to our solution approaches.

We begin by considering a search and rescue scenario where both human and automated agents share control over a fleet of unmanned aerial vehicles (UAVs) with the goal of locating a missing subject as quickly as possible. We describe a new interface and search framework, Hydra, which merges the intuition, reasoning, and vision capabilities of humans with the computational power of machines to reduce the expected time to locate the subject. The interface allows participating human agents to collaboratively decide where to send the UAVs via spatial dynamic voting, a geometric method for aggregating regional selections (votes) on a map. Via extensive simulation and theoretical analysis, we show that our method can be an effective component of search and rescue operations.

In the next chapter, we present a new graph-theoretical model for filtering a large set of genes to identify those that exhibit the most significant change in expression values between a series of control and test experiments; this is known as the Gene Selection Problem. Although not a geometric model in the traditional sense, graph theory allows us to organize data in abstract geometric spaces, where similarity metrics are used to define relative distances between nodes of data as opposed to working with an absolute coordinate system. Our algorithm first pre-processes the data using statistical hypothesis testing to filter out statistically irrelevant genes, and then we analyze the expression levels recorded for each gene by modeling them on a graph and evaluating the capacity of the cut between the test and control experiments. The capacity of a cut on a graph is a measure of the separation between two disjoint sets of nodes, and we use this value to rank the genes. We evaluated our

model on a rich data set assessing the success of embryo implantation in mice in the presence or absence of uterine dendritic cells. A thorough biological analysis of our results enabled the discovery of significant factors that were not identified by more traditional, statistical methods.

In the remaining chapters of this dissertation, we transition to a series of algorithms and models for filtering information in a collaborative, social context. We begin by presenting a new, constant-time recommender system for jokes that adapts in real-time to changes in user preferences or mood. We also present an extension of this system that makes personalized recommendations on where participants might wish to donate their money.

Chapters 5 and 6 consider the domain of collaborative opinion and idea sharing in an online setting. We present a new tool, Opinion Space, that we are developing for visualizing and crowdsourcing a diversity of insights collected via textual responses to a discussion question. Opinion Space projects participants onto a two-dimensional plane using Principal Component Analysis based on their levels of agreement with a series of statements. The projection is specifically designed so that participants with similar opinions will be near each other in the space; this allows participants to easily navigate the diversity of opinions shared by others.

Over the last two years, we have released multiple versions of Opinion Space and collected several rich data sets for analysis. In Chapter 5 we describe the interface and design decisions made when building the site. We also present results from a controlled user study comparing user engagement with Opinion Space versus more traditional models of online opinion sharing (specifically, linear comment lists). Not only did we find that participants were significantly more engaged with Opinion Space, but they had significantly higher levels of agreement with and respect for the responses that they read.

In Chapter 6 we present several models, both geometric and statistical, for ranking the contributions of our participants based on how insightful they are. Our primary model considers the spatial relationships between users in addition to the ratings they give each other; the intuition behind the model can be described as follows. By giving users the opportunity to rate the responses they read, we allow for the very likely possibility that users will only promote their own interests and rate opposing opinions poorly, even if it is a well-written and pointed response. We claim that this behavior is of little value towards our objective of identifying insightful ideas, because users are simply reinforcing their own opinions. Visually, one can imagine that the space of users is partitioned into subgroups or smaller spheres of agreement, and we are interested in emphasizing the comments where these spheres intersect. In this scenario, we have identified users of different viewpoints that have potentially found a legitimate middle ground.

Chapter 7 provides concluding remarks on our work with Opinion Space from a New Media and social responsibility perspective, and we present preliminary results on future work in the area.

To my incredible family, who gave me the strength and courage
to pursue my dreams.

Acknowledgements

First and foremost, I would like to thank my advisor, Professor Ken Goldberg, for his invaluable guidance and continued support over the course of my graduate career. Under his mentorship I learned to push the limits of research, both technically and philosophically. He taught me the importance of understanding the sociological impacts of the technology we build, and he showed me that technology can be communicated and understood in beautiful, artistic ways.

I would also like to extend my immense gratitude to my co-advisor, Professor Dorit Hochbaum, for taking so much time to share with me her incredible wealth of knowledge. She equipped me with a powerful set of tools for solving difficult graphical problems, and perhaps more importantly, she served as an inspiring role model. Her rigor and high expectations pushed me to become a better researcher, and It has been a true honor to learn from and work with her.

Thank you to my dissertation and qualifying exam committee members: Professor John Canny, Professor Ilan Adler, and Professor Coye Cheshire. They're feedback and support were tremendously helpful.

I would like to acknowledge my deepest appreciation for the incredible amount of support I received from Professor Gail de Kosnik. I thoroughly enjoyed our countless discussions about the social consequences of technology, and she showed me that it's possible to write about mathematically theoretical work within the context of the Humanities. This newfound awareness has undoubtedly made me a better researcher.

I would also like to thank the wonderful members of the Automation Sciences Lab, who were always eager to brainstorm and try out new research ideas. Specific thanks go to the Opinion Space team, who made a large part of my dissertation research possible: Tavi Nathanson, David Wong, Siamak Faridani, and Sanjay Krishnan.

I am especially thankful for my wonderfully supportive friends, whom I could count on for anything. To Anand Kulkarni: you always found a way to help me through even the most desperate situations. I am continually in awe of your ambition and dedication towards making the world a better place; it has been such an honor to share this journey with you. To Melissa Goldstein, Orly Perlstein, Mira Leytes, Evan Davidson, and Timmy Siauw: I don't know how I would have made it without you!

And finally, I would like to thank the members of my immediate family for their unconditional love and support. To my sisters Dafna, Ayelet, and Joanna: you're my best friends and I love you more than anything. To my parents, Rakefet and Moshe, I can't begin to thank you enough for your unwavering encouragement; without you this wouldn't have been possible. To my grandmother, Sylvia: you never hesitated to do anything within your reach to help me along; thank you for everything.

Contents

1

Introduction

This dissertation explores the use of geometric models and techniques for a variety of search and filtering problems, from spatial search with robotic systems to information search and retrieval both through gene expression data and over a social network. One of the primary advantages of geometric methods is that they rely on an intuitive understanding of the structure of problem, whether it be in Euclidean space or some abstract space.

In this thesis, we investigate four different application domains. The first (Chapter 2) is a geometric framework for allowing humans and automated agents to share control over a set of robotic aerial vehicles with the task of locating a missing subject. The idea is to create a system that takes advantage of what humans and computers do best: humans bring intuition and fast image processing capabilities to the table, and computers are able to manage and crunch through extraordinary amounts of data in a fraction of the time as humans.

We then move on to information filtering models using network flows. While technically these models fall under the domain of Graph Theory and not geometry, they can be thought of as geometric models in an abstract sense. Network flow models are structured on graphical models consisting of nodes and directed arcs or undirected edges between nodes. These arcs or edges may be weighted, and hence can be thought of as existing in a non-Euclidean geometric space. The space is no defined by a formal coordinate system, but rather relative distances between nodes. Further, the distances or relationship between some pairs of nodes may not be well-defined. However, we ask many of the same questions on graphical structures in this space that we do in Euclidean space such as finding the shortest path between two points in the space. In both spaces, we are also concerned with questions about volume. In the geometry defined by a graph, one analogy to volume is the amount of flow that can be pushed from one node or set of nodes to another; this is purely defined by the constraints implied by the weights on the arcs or edges of the graph.

The next four chapters (4-7) consider geometric models for information filtering and ranking problems on social data sets. Chapter 4 presents new constant-time models for making personalized recommendations of a) jokes and b) non-profits to donate to.

Chapters 5 and 6 investigate geometric models for visualizing the diversity of opinions expressed in textual responses to a discussion question and ranking those responses according to how insightful they are. In Chapter 7 we provide concluding remarks from a New Media perspective and discuss future work. The remainder of this section summarizes each chapter in greater detail.

Chapter 2: A Geometric Framework for Mixed-Initiative Search. In this chapter we demonstrate a framework and algorithms for collaborative human and automated (or *mixed-initiative*) decision making within the context of outdoor search and rescue. Hydra is a networked simulation tool that allows n human and k automated agents operating under different assumptions to share control over m unmanned aerial vehicles (UAVs) with cameras, with the goal of locating a hidden subject θ as quickly as possible. The agents are modeled on a pre-defined hierarchy of authority, and the search space is characterized by varying degrees of obstructions.

Search is based on iterating the following cycle of four steps: 1) all agents generate image requests based on their individual probability density functions (pdfs), 2) Hydra collects requests and computes an optimal assignment of images to the UAVs, 3) Hydra processes the resulting image data and specifies whether or not the subject was detected, and 4) all agents update their pdfs. We propose initial models and algorithms under this framework, and we show via simulations of a scenario with three agents and one UAV that our method performs 57.7 percent better than a theoretical upper bound for a single agent and UAV.

Key technical challenges and contributions include a geometric and graphical model for optimizing the control of automated resources that would maximize an information theoretic measure of the search space. Another challenge involves the development of an adequate filtering algorithm for determining individual posterior probability distributions as information is gathered.

Chapter 3: Information Filtering with Network Flows. In this chapter we present a new mathematical model,"Graph Cut" (GC), for identifying the most differentially expressed genes as evidenced between a set of test experimental versus control repetitions. We evaluate the model on a gene expression data set designed to study the effects of uterine dendritic cell (uDC) depletion on embryo implantation in mice. The GC model is a hybrid approach based on statistical analysis and on graph theory, which is a sub-field of theoretical computer science. Statistical analysis is used to filter the genes that do not exhibit a statistically significant difference in expression levels between the test and control repetitions, and graph theory is used to define the degree of separation between the repetitions.

Results with GC were compared with those of the commonly used LIMMA algorithm and the CLICK algorithm, the latter is a method also based on graph theory. Our findings indicate that while GC and CLICK often return significant results, GC is more robust, as it tends to find genes that exhibit a significantly greater separation between test and control groups and bare greater biological significance. In this respect, the results yielded by GC revealed a distinct group of pro-inflammatory chemokines

differentially upregulated in embryonic day (E) 4.5 in uDC-depleted implantation sites, more extended than with CLICK and completely absent from the LIMMA analysis. Other genes found to be significantly downregulated by GC upon uDC depletion, were mostly related to embryo implantation or uDC regulation and this group was further extended on E5.5. While LIMMA picked up some of these genes on E5.5, CLICK was unable to differentially detect any of them. This study serves a dual purpose of illustrating both the GC mathematical model for gene expression array analysis as well as sheds light on uDC action in mouse embryo implantation.

Chapter 4: Constant-time, Adaptive Collaborative Filtering. Recommender systems strive to recommend items to users that they will appreciate and rate highly, often presenting items in order of highest predicted ratings first. In this working chapter we present Eigentaste 5.0, a constant-time recommender system that dynamically adapts the order that items are recommended by integrating user clustering with item clustering and monitoring item portfolio effects. This extends the Eigentaste 2.0 algorithm, a geometric method that uses principal component analysis (PCA) to cluster users offline. In preliminary experiments we backtested Eigentaste 5.0 on data collected from Jester, our online joke recommender system. Results suggest that it will perform better than Eigentaste 2.0. The new algorithm also uses item clusters to address the cold-start problem for introducing new items.

We also present Donation Dashboard, a system that recommends non-profit organizations to users in the form of a portfolio of donation amounts. Recommendations are made using Eigentaste 2.0 in combination with a new method for generating a weighted portfolio of recommendations. The key challenge is to generate a customized portfolio that does not necessarily exclude items already rated by the user. Under our method, the weights for items in the portfolio that have not yet been rated by the user are normalized factors of their predicted ratings, and the weights for items previously rated by the user are normalized factors of the actual ratings. Donation Dashboard 1.0 launched in April 2008, and as of May 8 2009 we have collected over 59,000 ratings of 70 nonprofit organizations from over 3,800 users.

We include a description of our experience developing Donation Dashboard, including the design of the system and our new method for portfolio generation. We use Normalized Mean Absolute Error (NMAE) to measure the accuracy of Eigentaste using our dataset of non-profit organization ratings and we compare that with the global mean algorithm. We analyze the data collected since the launch of the site, and we have made our dataset available to the public. Donation Dashboard and the Donation Dashboard dataset are accessible at:

`http://dd.berkeley.edu`
`http://dd.berkeley.edu/dataset`

Chapter 5: Visualizing the Diversity of Online Textual Responses. Internet users are increasingly inclined to contribute textual responses to online news articles, videos, product reviews, and blogs. The most common interface for navigating these responses is a linear list, sorted by time of entry or by binary ratings. It is

widely recognized that such lists do not scale well and can lead to "cyberpolarization," which serves to reinforce extreme opinions. In this chapter we present Opinion Space: a new online interface incorporating ideas from deliberative polling, geometry (specifically, dimensionality reduction), and collaborative filtering that allows participants to visualize and navigate through a diversity of responses. We also report results of a controlled user study, in which we found that when Opinion Space was compared with a chronological List interface, participants read a similar diversity of responses. However, they were significantly more engaged with the system, and they had significantly higher agreement with and respect for the responses they read.

Chapter 6: A Spatial Model for Ranking Textual Responses. In this chapter we present a series of methods, both geometric and statistical, for ranking the textual responses in Opinion Space according to how insightful they are. The first method we propose is a spatial approach that considers the physical location of participants in the space and the corresponding biases of opinion we expect to see. Specifically, we assume that participants with similar opinions (i.e. those that are near each other in the Space) will tend to rate each other higher than would participants with differing opinions (i.e. those that are farther from each other in the Space). This assumption stems from the core assumption made by recommender and collaborative filtering systems: that participants with similar preferences or taste are likely to agree on the quality of the same item. We then extend this model to account for the inherent uncertainty in the ratings collected using a statistical modeling approach. We also present a recursive model that considers that reliability of the ratings provided by each participant, which is measured in terms of how often they agree with the majority on the quality of a response. We evaluate the methods on empirical data collected from two data sets that contain structurally different properties, namely in terms of sparsity.

Chapter 7: Concluding Remarks from a New Media Perspective. This chapter serves as concluding remarks on the design of online opinion-sharing spaces. It begins with an expository discussion from a New Media perspective of deliberative democracy and the public sphere, particularly within the context of an online setting. We discuss the importance and challenges of upholding the ideals of these theories, both from a philosophical and technical point of view.

We follow this discussion with a description of two new improvements to Opinion Space that serve to facilitate deliberative democracy in an online setting. The first is a new dimensionality reduction technique for building the Opinion Space map, Canonical Correlation Analysis (CCA), which considers the content of the participant's textual response in addition to her ratings. This results in an opinion map that yield stronger spatial relationships and hence provides more significant meaning. Consequently participants are given more reliable control when navigating the space in terms of the diversity of responses they see.

The second extension we consider is a new user interface feature and recommendation algorithm that we call the "Diversity Donut." This extension is aimed at giving participants explicit control over the diversity of the responses recommended to them,

so as to cater to different preferences or personalities when considering the opinions of others. We report on results from a preliminary pilot study, in which we were unable to establish a statistical advantage over other recommendation methods. However, subject self-reported data indicated that the Diversity Donut yielded the most diverse set of comments and the highest satisfaction in regards to diversity.

2

A Geometric Framework for Mixed-Initiative Search and Control

2.1 Introduction

In this chapter we present a series of geometric algorithms and a framework for *mixed-initiative* control of a fleet of robotic vehicles for search applications in the physical world. We define a mixed-initiative system to be one where both human and automated agents simultaneously share control over a limited set of resources. The problem differs from traditional search problems in that it requires the ability for humans to participate in the search decision process. The idea is to create a system that takes advantage of the unique strengths of both humans and computers. Humans are highly visual beings in that they can process incredible amounts of visual information extraordinarily quickly; they also have reasoning and intuition skills that are difficult to quantify, but they extremely slow at processing quantitative information. Computers, on the other hand, struggle with vision and intuitive reasoning tasks but excel at computation. To create a system where both humans and computers can contribute and share control on a relatively equal playing field, we require an intuitive way to accept input and commands from human agents in a framework that is largely quantitative. Catering to the visual strengths of human agents, we propose a visual interface with which humans can request regions to search on a planar map. This motivates the design of geometric algorithms for processing the spatial information collected and computing the optimal search strategy. In the remainder of this section, we provide further motivation of the problem and describe a geometric framework that allows both humans and computers to work together to complete a search task.

Recent technological advances in unmanned flight have provided equipment useful in designing automated search and rescue systems that allow searchers to cover ground more quickly without putting human operators at risk. Because such missions are too complex to be either fully automated, we seek to understand how humans and

automation should share authority over complex command and control operations in order to maximize a measure of information about the system as quickly as possible. By integrating human intuition and reasoning capabilities (which can be difficult to quantify) with computational resources, we provide a robust framework for collaborative, mixed-initiative decision making that can accommodate different agent authority structures. Our objective is to experiment with problem solving strategies for search operations where multiple agents collaborate to control a single or multiple robots to explore regions of a map.

Search and rescue missions for lost people in the wilderness and at sea are often long, costly, and dangerous processes. In the wilderness, several teams of people are sent out on foot with the hope of covering as much ground as possible. Severe weather conditions and dangerous terrain (such as the threat of an avalanche) can force search teams to call off their efforts until the dangers subside; for example, in a rescue effort to find three missing men on Oregon's Mount Hood in December 2006, searchers lost several critical days due to extreme weather conditions (48), resulting in the death of all three climbers. Time is most critical when searching for a missing person, both because they are less likely to survive the longer they are exposed to the elements, and because if they are moving on foot the search radius will have to increase significantly with respect to time. Hence, if a mixed-initative approach can reduce the amount of time required to locate a missing person by even a few hours, it can mean the difference between life and death.

Unmanned aerial vehicles are robotic aircraft capable of gathering, processing, and relaying data such as images, temperature, and other sensed information. While this may be very useful in the search and rescue domain, there are several challenges yet to be addressed. We are primarily concerned with enabling UAVs to *assist* search and rescue teams by enabling them to collaborate with each other and to direct the UAVs to gather more useful information; the idea is to combine the powers of cognitive- and computer-based processing so that search can be carried out most intelligently and efficiently. Algorithmic challenges in doing so include the design of a collaborative graphical user interface that allows searchers to easily state which information they seek from the UAVs (i.e. which region of the search domain they wish to investigate further), and to convert that information into appropriate and efficient directions to the UAVs.

Hydra is a game-based simulation and visualization tool we developed that enables distributed human and automated agents to collaborate via a scalable spatial dynamic voting, networked interface (Figure 2.1). Agents request updated images of areas of interest by specifying rectangular subregions of the search domain, which provides a unified representation for visualization and coordination. Each image request is subsequently treated as a *spatial* vote for that information, and because search missions are limited in both time and resources, Hydra uses this information to determine the set of images that will accommodate the greatest number of searchers (weighted by authority level).

A prototype of the simulation tool is accessible via the internet and runs on Java-

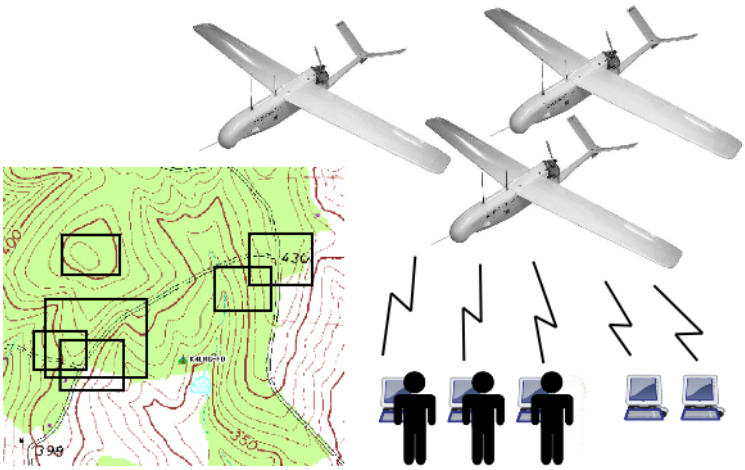


Figure 2.1: Three UAVs with mounted cameras are controlled by a sequence of frame requests from distributed human and automated agents.

enabled browsers. In the simulated scenario, searchers are asked to work together to find a missing person by sharing control over a single camera-mounted unmanned aerial vehicle, which is capable of taking and relaying photographs of specified regions of the search domain. Searchers may operate under varying search strategies that may be based on or translated to different spatial probability distributions of the search domain.

2.2 Related Work

In 2007, the plane that businessman Steve Fossett was flying solo went missing somewhere over the Nevada desert. Despite countless attempts by various search and rescue parties and air patrols, his plane was nowhere to be found. Several days into the search, high-resolution satellite images were uploaded to Amazon.com's Mechanical Turk engine, along with a request for volunteers to review the images and search for any evidence of Fossett's plane. (See Figure 2.2.) Within three days, up to 50,000 people had volunteered and reviewed over 300,000 images, eac covering 278 square feet of land. (54) Although the collaborative effort was ultimately unsuccessful and rescue crews were unable to locate either Fossett or the wreckage of his plane, it is an inspiring example of the willingness for individuals to contribute to search efforts, even when asked to perform menial tasks. It also motivates the need for a more intelligent system that fully harnesses the power of mixed-initiative, collaborative search.

The Automation Sciences Lab has an extensive history of projects related to collaborative robotic control, starting with the Telegarden project that launched in 1995 and ran until 2004. (58; 61) Recognized as world's first robot controllable over the internet, the Telegarden was a community garden that allowed members to remotely plant seeds and tend to plants by collaboratively controlling a robotic arm and camera.

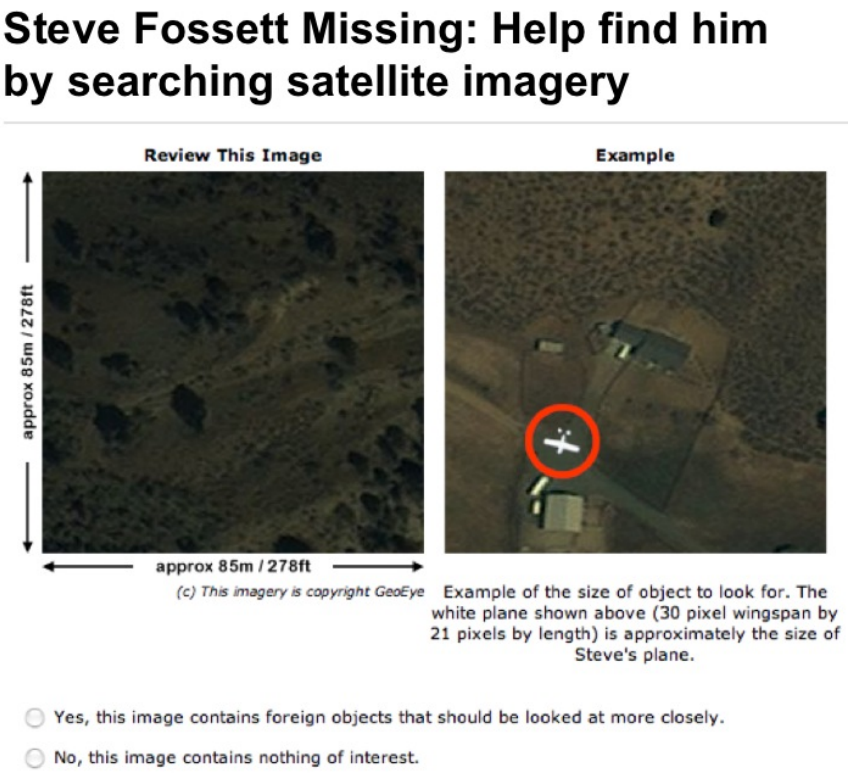


Figure 2.2: Use of Amazon.com's Mechanical Turk to search for Steve Fossett, whose plane crashed at an unknown location in the Nevada desert on 3 September 2007. Users were asked to review a series of aerial images and flag any images that could potentially contain Fossett's plane.

Users had to share control over the robot and coordinate planting and watering in order for the garden to be successful. (See Figure 2.3 for an image illustrating the setup of the garden.) The Teleactor (62) project came out in 2001 and was designed to allow groups of users to collaboratively and remotely explore a location via a designated person (the "teleactor") with cameras and microphones capable of receiving commands from a central server. Users would vote on images for the teleactor to explore, and votes were collected and processed using a Spatial Dynamic Voting algorithm.

This algorithm led to the development of Song et. al's model for shared camera control where human users request sensor data corresponding to a specific rectangular region of a shared image. (136; 137) By treating these image requests as spatial votes, the authors leverage the geometric properties of these votes to formulate the model as an optimization problem, which they term *spatial dynamic voting*. Given a set of image requests, they provide both exact and approximate algorithms to determine the single rectangular region that maximizes user satisfaction. The authors make use of these algorithms in (135), where they describe a system that allows multiple users to

simultaneously share control over a single robotic camera. They consider the problem of sharing control over more than one camera in (157), and we provide an alternate formulation and algorithm in section 2.4.2.2. The original single-camera algorithm was employed in the Collaborative Observatories for Natural Environments (CONE) project, a remote, collaboratively-controlled observatory for bird watching via a robotic camera. (40) The project was first deployed in the Sutro Forest of San Francisco, California in April 2007 and was moved one year later to the Welder Wildlife Preserve in Texas.

Early work in collaborative control for multiple human operator, single robot systems includes a project by Cannon that enabled remote waste cleanup by having users specify locations in a shared image for a robot to excavate. (20) In this scenario, the author demonstrated an improvement in cleanup time, although he does not address conflict resolution between users.

McDonal et. al study protocols and interfaces for Virtual Collaborative Control (VCC) of robots, though their work does not allow for simultaneous control (95). In (59), Goldberg and Chen present a theoretical framework for the simultaneous control of an online robot by an ensemble of sources, which may include a combination of sensors, control processes, and human operators. They demonstrate their model with a system that averages multiple vector inputs to control the position of a moving point resource and show that it is robust to source malfunctions.

At the Center for Robot Assisted Search and Rescue, Murphy led several studies in human-robot interaction (23; 101), where the authors examined the use of robots in urban search and rescue settings to understand the workflow of such operations and the types of errors encountered. Murphy has also led work in cooperative control of mobile robots based on modeling and simulating societal behavior (102).

In designing a UAV control framework for search and rescue applications, two important considerations are collision avoidance (safety) and ground coverage (efficiency). In (123), Ryan and Hedrick give a control algorithm for a set of UAVs flying in formation to sweep the search space using four basic path rules. Hu et al study the optimal formation constrained multi-agent coordination problem in (79) and identify geometric properties of its solutions. Ryan, Nguyen, and Hedrick consider a Coast Guard search and rescue scenario where two UAVs assist a manned helicopter by expanding the range of visual data available to the pilot. (124) The authors present a decentralized controller for maneuvering the UAVs safely.

Baum and Passino present a search-theoretic approach for fully automated cooperative control of UAVs tasked with locating stationary targets. (6) They extend classic search theory techniques to incorporate trajectory generation and to allow for multiple information seekers. Although we do not explicitly consider the trajectory generation problem in this paper, we provide a framework for coordinated control of multiple UAVs that can incorporate such models.

Chaimowicz and Kumar study the problem of using a set of UAVs to coordinate and control a swarm of ground vehicles in urban environments. (24) They develop probabilistic and behavioral models for *shepherding* based on an hierarchical frame-

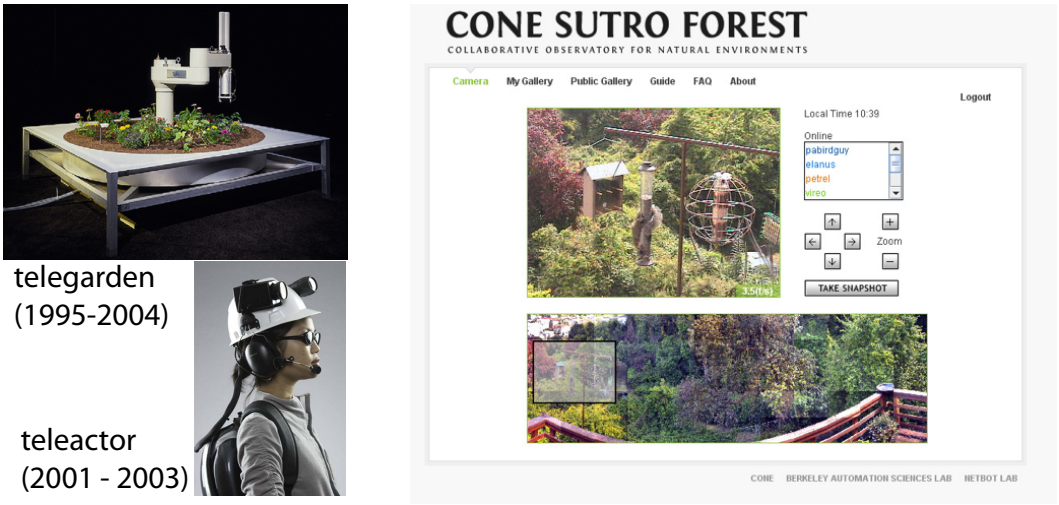


Figure 2.3: The Telegarden, Teleactor, and CONE projects are early examples of collaborative robotic control projects in the Automation Sciences Lab at UC Berkeley.

work. In (19) Caffarelli et al present algorithms for directing UAVs to monitor known stationary targets by paying them a minimum number of visits per unit of time while in consideration of the energy consumption of the UAVs and the uncertainty of the trajectories.

The following two groups consider the complete automation of a set of UAVs tasked with locating a target. We expand on their work by proposing image frames as a unified vocabulary by which both humans *and* automation can easily request sensor information from the system, in accordance with a user-defined hierarchy of agent authority. We also characterize the underlying search space with varying degrees of obstructions, which in turn affects the quality of information collected. We then present a model for extracting data from the sensors that provides a tradeoff between the size of the sampled image and the quality (reliability) of the information.

In (12), Bourgault et al describe a decentralized Bayesian approach for locating a single target by coordinating multiple autonomous agents. In their framework, automated search agents make individual decisions based only on their knowledge (prior probability distribution), and the information gathered by the different sensing platforms. Information is combined using a fully decentralized Bayesian data fusion technique, and controls are given using a decentralized coordinated control scheme. Furukawa expands on this work in (55) with the development of a coordinated control method for autonomously searching for and tracking multiple targets using multiple vehicles. Hoffmann et al (75) also consider the automation of a set of networked UAVs and develop a non-parametric technique based on particle filtering to determine in real-time the optimal control sensor locations to minimize the number of future observations required to determine the state of a target. In their setup, each UAV maintains its own estimate of the target's current state and uses an onboard particle filter to ap-

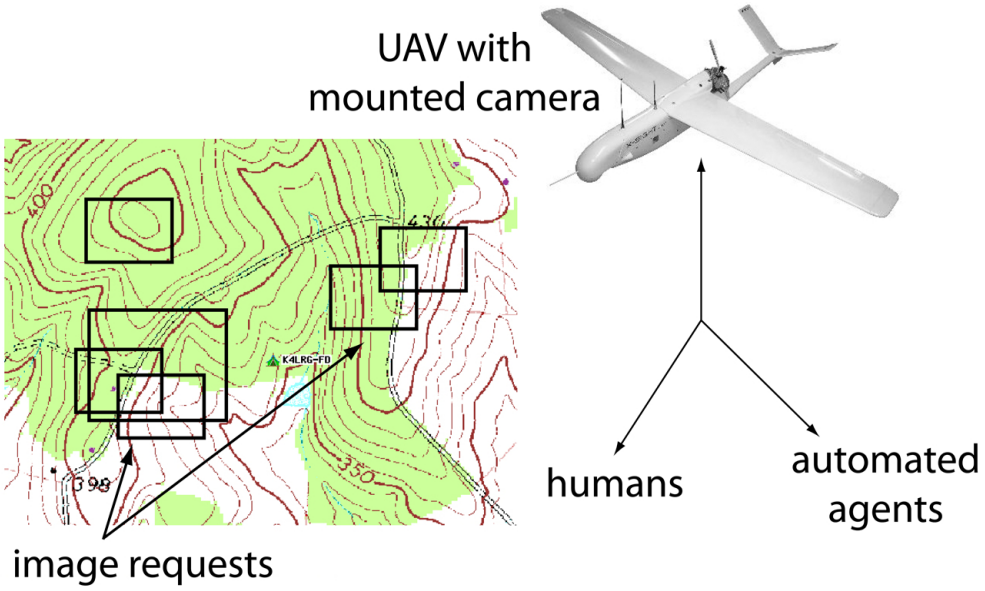


Figure 2.4: Illustration of the Mixed-Initiative Search and Rescue problem. UAVs with mounted cameras are collaboratively controlled by a mixture of both human and automated agents. Each agent makes frame requests to a central server, which uses spatial dynamic voting to identify the best course of action for each UAV.

proximate the posterior distribution once sensor data has been gathered. The authors present two polynomial-time approximation algorithms, making the network scalable while maintaining a high degree of descriptiveness.

2.3 Problem Formulation

In this section we give a formal mathematical formulation of the mixed-initiative search and rescue problem and outline a framework of steps for its solution. In Section 2.4 we give our proposed solution in mathematical detail, which we then evaluate via simulation in Section 2.6.

Every search is initialized with the following inputs. We limit the search space to a bounded area of the plane Θ that contains a hidden, stationary subject with location $\theta \in \Theta$. We assume that the search space is characterized by a pre-specified parameter $c\,(x, y)$, which corresponds to the density of obstructions in Θ at point (x, y). (These could be buildings, fog, vegetation, etc.) A distributed group of n human and k automated agents collaboratively control a set of m UAVs with mounted cameras for data collection. Each agent has an associated authority level $\alpha_i \in [0, 1]$ and maintains a pdf $P_{i,t}$ over Θ of the subject's location.

In each iteration of a session, every agent specifies a rectangular frame to investigate

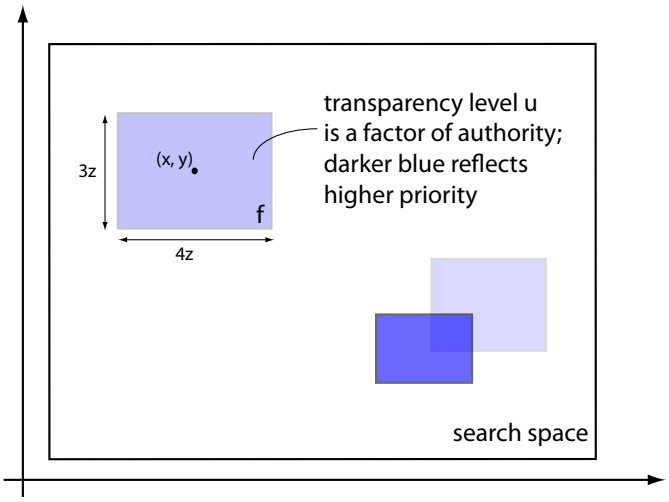


Figure 2.5: Illustration of a frame request as defined by the Hydra framework. Frames are defined within a fixed region of the two-dimensional plane. Depending on camera constraints, frames may be restricted to a fixed aspect ratio such as 3:4.

further. (See Figure 2.5.)

Definition. A frame $f\,(x, y, z, t)$ corresponds to a rectangular subregion of the search space, centered at point (x, y), with zoom level z, and indexed by time t.

Both human and automated agents submit requests in this unified format. We assume that the combined number of agents is larger than the number of UAVs available. In each iteration, the system must compute a set of m frames that maximize total "satisfaction" among the agents.

Once data for a frame f is collected, the information is processed and a binary value $B(f)$ is returned indicating whether or not the subject is detected within f. $B(f)$ is a Bernoulli random variable that is more likely to return a correct answer when a frame is of high resolution (i.e. covers a small area) and the density of obstructions is low.

We assume that as the area spanned by an image and/or the density of obstructions increases, the quality of information decreases. Let a be a pre-specified termination threshold corresponding to the maximum acceptable probability of a false positive in a candidate frame. Then, a search session terminates when the sensor detects the subject in a frame with small enough area so that we can ascertain with probability $1 - a$ that the sensor information is accurate. We determine the maximum area of a terminating frame solely as a function of a and the average density of obstructions in the frame,

$$c_f = \frac{1}{Area\,(f)} \int_f c\,(x, y)\,dy\,dx. \tag{2.1}$$

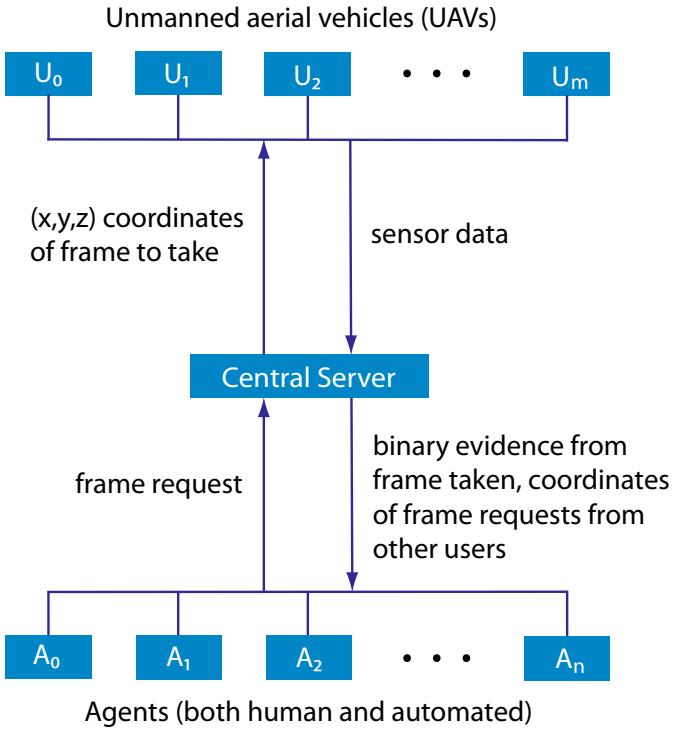


Figure 2.6: Illustration of information flow in the Hydra framework.

Since time is a major factor determining the success of a search and rescue operation and hence the system, *the goal of each agent is to minimize t, the number of iterations required to locate the subject.* For this project, we have broken down our problem solution into the following four steps, for each of which we have developed models and algorithms. (The flow of information in the system is illustrated in Figure 2.6, and a breakdown of the problem formulation is given in Table 2.1.)

1. **Agent Frame Request:** All agents generate frame requests based on their individual pdfs of the subject's location.

2. **UAV Frame Allocation:** Hydra collects requests and computes an optimal frame assignment to the UAVs.

3. **Sensor Data Extraction:** Hydra processes the resulting image data and specifies whether or not the subject was detected.

4. **Prior Distribution Update:** All agents update their pdfs to incorporate the new data.

Problem Formulation
Input: Search domain Θ; number of UAVs m; density of obstructions $c(x, y)$; termination threshold a
Goal: Minimize t, the number of iterations to termination
Steps in Each Iteration: 1) Agent frame request 2) UAV frame allocation 3) Sensor data extraction 4) Prior distribution update
Termination Condition: A sampled frame f such that $B_t(f) = 1$ and $\Pr(\theta \in f \mid B_t(f) = 1) \geq 1 - a$

Table 2.1: Summary of the system inputs, goal, steps, and termination condition.

2.4 Framework and Algorithms for Mixed-Initiative Search

In this section we present our approach to each of the above stated tasks in technical detail.

2.4.1 Step 1: Agent Frame Request

At the beginning of each iteration, participating agents submit frame requests corresponding to the rectangular subregions of the search space they wish to investigate further. We consider different strategies for each class of agent.

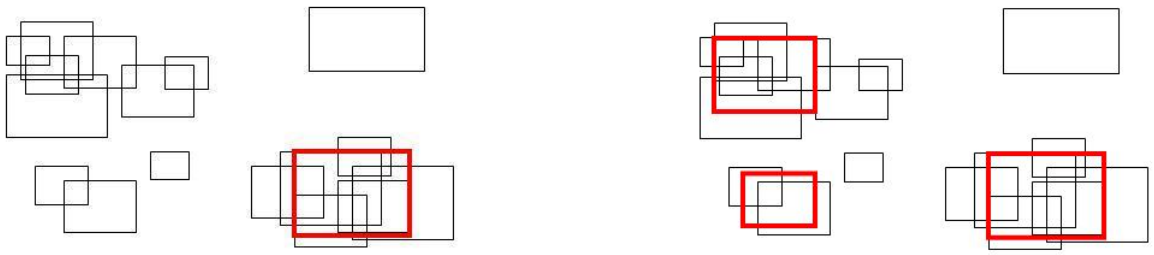
2.4.1.1 Human Agents

To facilitate rapid decision-making, the Hydra interface maintains for each agent a visual representation of his or her probability distribution $P_{i,t}$ of the subject's location. Each cell of the search space is filled with a shade of blue, where a darker shade corresponds to a higher likelihood that the subject is located within that cell.

With this visual representation system, human agents can quickly get a feel for what regions of the search space have higher probabilities of finding the subject. The agents can then decide which frame to request based on intuition.

2.4.1.2 Automated Agents

We consider a search strategy for a single automated agent using results from information theory. The information entropy of a probability distribution is a measure



(a) Single Frame Allocation **(b)** Multiple Frame Allocation

Figure 2.7: A sample snapshot of the frame requests in the queue at a given moment in time. The solution to the single frame allocation problem is given by the thick rectangle in (a), and the corresponding solution to the multi-frame allocation problem is shown in (b).

of uncertainty, where higher entropy corresponds to greater uncertainty regarding the outcome of a random variable. The entropy of agent i's distribution is given by

$$H(\theta) = -\int_x \int_y \mathrm{P}_{i,t}(x,y) \log_2 \mathrm{P}_{i,t}(x,y)\, dy\, dx \tag{2.2}$$

As shown by Shannon in (128), the uncertainty in the agent's distribution is minimized by sampling the frame that minimizes the expected information entropy of the posterior distribution. This is equivalent to maximizing the expected log-likelihood of the posterior, known as *information gain*.

Let $p_1 = \mathrm{P}(B(f) = 1)$ be the probability that the sensor data for frame f indicates that $\theta \in f$, and let $p_0 = \mathrm{P}(B(f) = 0)$ be the probability that the subject was not detected in f. We assume a general probability model for now and give an explicit one below. The information entropy conditioned on the sensor data is defined as

$$\begin{aligned} H(\theta|B(f)) \quad = \quad & -(p_1 \log_2 p_1) H(\theta|B(f) = 1) \\ & -(p_0 \log_2 p_0) H(\theta|B(f) = 0) \end{aligned} \tag{2.3}$$

Thus, the frame f_i^* that maximizes the information gained for agent i is

$$f_i^* \quad = \quad \arg\max_f \; H(\theta) - H(\theta|B(f)) \tag{2.4}$$

as given by (88).

While choosing the frame that maximizes information gain helps concentrate the agent's pdf, it is not designed to zero in on the areas with highest probability and hence the subject's most likely location. We thus propose a two-state search process for automated agents. In the first stage, the agent's strategy is to request the frame that minimizes the expected entropy of his or her posterior distribution. In the second stage,

the agent employs a greedy strategy by requesting the frame of maximum acceptable size for termination (given a and c) that holds the greatest expected probability.

2.4.2 Step 2: UAV Frame Allocation

In each cycle, we have a queue of frame requests initiated by distributed human and automated agents. Due to limited resources, not all of these requests can be met within a reasonable amount of time, since the UAVs can take several seconds to physically adjust their positions, focus, and record data. Consequently, we require a method that considers certain user attributes to determine which frames to record and in what order; we call this the *UAV frame allocation problem.*

We present a geometric approach that uses agent authority coupled with cumulative dissatisfaction to prioritize frame requests. Since both human and automated agents submit requests in the format of a frame, we do not distinguish between the two. We first consider the case where the agents share control over a single UAV (i.e. $m = 1$) by formulating a spatial dynamic voting optimization problem. We then provide a heuristic that extends this solution to consider multiple available resources.

2.4.2.1 Single Frame Allocation

We adopt the model given in (136; 137) to mathematically define a user's satisfaction with a proposed frame. Our objective then becomes to maximize the priority-weighted sum of the users' individual satisfaction measures, which we denote as the *global satisfaction function.*

Let $F = \{f_1, \ldots, f_n\}$ be a set of axis-parallel rectangles with fixed aspect ratio that represents the frame requests currently on the queue. We define agent i's *individual satisfaction* $s(f, f_i)$ as a measure of the similarity or overlap between candidate frame f and frame request $f_i \in F$. We use the *intersection over maximum* to measure the similarity between two rectangles, given by:

$$s(f, f_i) = \frac{Area\,(f \cap f_i)}{\max\,(Area\,(f)\,,\, Area\,(f_i))} \tag{2.5}$$

This function exhibits the following property: $0 \leq s(f, f_i) \leq 1$. The agent's satisfaction is therefore 0 when the intersection of f_i with f is the empty set (i.e. they are disjoint), and 1 when $f_i = f$. A sample solution to the problem is illustrated in Figure 2.7 (a). Furthermore, the function is piecewise linear, allowing the use of computationally efficient optimization algorithms.

Let $s_{i,t}$ be the i^{th} agent's satisfaction with the frame allocated at time step t, and let the agent's dissatisfaction with the frame be $\bar{s}_{i,t} = 1 - s_{i,t}$. We model the priority ρ_i of agent i's frame request by taking the product of the agent's authority level and a normalized exponentially decaying function of the agent's dissatisfaction with the three previously allocated frames:

$$\rho_i = \frac{\alpha_i}{0.875}\left(\frac{1}{2}\bar{s}_{i,t-1} + \frac{1}{4}\bar{s}_{i,t-2} + \frac{1}{8}\bar{s}_{i,t-3}\right) \tag{2.6}$$

Hence, the more dissatisfied an agent is with the three previously allocated frames, the higher his priority will be. Furthermore, the model is constructed so that an agent's priority can never exceed his authority, and greater weight is given to dissatisfaction from more recently allocated frames.

We desire an axis-parallel rectangle f^* that maximizes total satisfaction for all agents, weighted by priority:

$$f^* = \arg\max_{f} \sum_{f_i \in F} \rho_i \, s(f, f_i), \tag{2.7}$$

$$\text{subject to} \qquad f \leq \max\{\text{Area}\,(f_i) : f_i \in F\}$$

We constrain the maximum area of an allocated frame to be less than or equal to the area of the largest frame request in the queue; this prevents the algorithm from selecting larger frames in an attempt to satisfy more agents, which would in turn significantly delay the time until the search can terminate successfully.

Polynomial-time exact and approximation algorithms for identifying a single optimal frame f^* in this context are given in (136; 137). In the following section we extend this to find the m best frames.

2.4.2.2 Multiple Frame Allocation

Given that there are m UAVs available for use, we construct an optimization problem that seeks to determine a sequence of m frames that maximize the sum of all users' individual satisfaction. A graphical example of this problem is given in Figure 2.7 (b). It can be likened to the p-center or facility location problem in Operations Research, in which a set of facilities must be chosen and located to minimize the distance between customers and their nearest facility. (136)

We seek to determine a set of n frames $F = \{f_1, \ldots, f_n\}$ that maximizes total expected reward, where reward is a pre-defined function. The behavior of this objective is entirely dependent on the chosen reward model for capturing images of each species. For example, if the reward for a species s is inversely proportional to the frequency at which the bird is naturally observed, then this objective translates to finding the set of frames that will maximize the likelihood of capturing images of more elusive species. If, on the other hand, rewards are directly proportional to the frequency at which birds are naturally observed, then the objective becomes to maximize the probability that *any* bird is found.

We begin by defining the following decision variable for selecting a frame:

$$f_i = \begin{cases} 1 & \text{if frame frame } i \text{ is selected} \\ 0 & \text{otherwise} \end{cases}$$

The number of times we expect to observe species s in a set of frames $F = \{f_1, \ldots, f_n\}$ is $\sum_{f_i \in F} P_{si}$. Hence, the expected reward for taking frame f_i is given by

$$E\left[\text{reward for taking frame } f_i\right] \;=\; \sum_{s \in S} \alpha_s P_{si} \tag{2.8}$$

Our objective is therefore to find a set of frames $f_1, \ldots, f_n$ that maximizes the total expected reward, as follows.

$$\begin{aligned} \max \quad & \sum_{i=1}^{m} \sum_{s \in S} \left[\alpha_s P_{si}\right] f_i \\ \text{subject to} \quad & \sum_{i=1}^{m} f_i = n \\ & f_i \in \{0, 1\} \end{aligned}$$

Observe that if we maximize the stated objective function without any additional constraints, the solution will always be of the form $f_1 = f_2 = \cdots = f_n$. To avoid this, we will need to place bounds on the amount of acceptable overlap between any pair of candidate frames. In the simplest case, we can require that all frames are mutually exclusive; that is, the area of intersection between any two frames must be 0. To keep the problem general, we allow for the maximum area of overlap to be a, and we introduce limits on the size of a frames. The multiple frame selection problem (MFS) now becomes the following constrained optimization model.

$$\begin{aligned} \text{(MFS)} \;\; \max \quad & \sum_{i=1}^{m} \sum_{s \in S} \left[\alpha_s P_{si}\right] f_i \\ \text{subject to} \quad & \sum_{i=1}^{m} f_i = n \\ & \text{Area}\left(f_i \cap f_j\right) \leq a \qquad \forall\, i \neq j \\ & l\left(f_i\right) \leq \text{Area}\left(f_i\right) \leq u\left(f_i\right) \\ & f_i \in \{0, 1\} \end{aligned}$$

Reduction to Independent Set

We now show that MFA reduces to the Independent Set (IS) problem, and is therefore NP-complete. Let $\{f_1, \ldots, f_m\}$ be the set of all possible frames in the search space Θ. The decision version of MFA is to find a feasible set of n frames such that the total expected reward is at least r. A polynomial-time verifiable certificate that there exists such a solution would be a corresponding set of n frames.

Theorem. *Independent Set* $\leq_p$ *Multiple Frame Allocation*

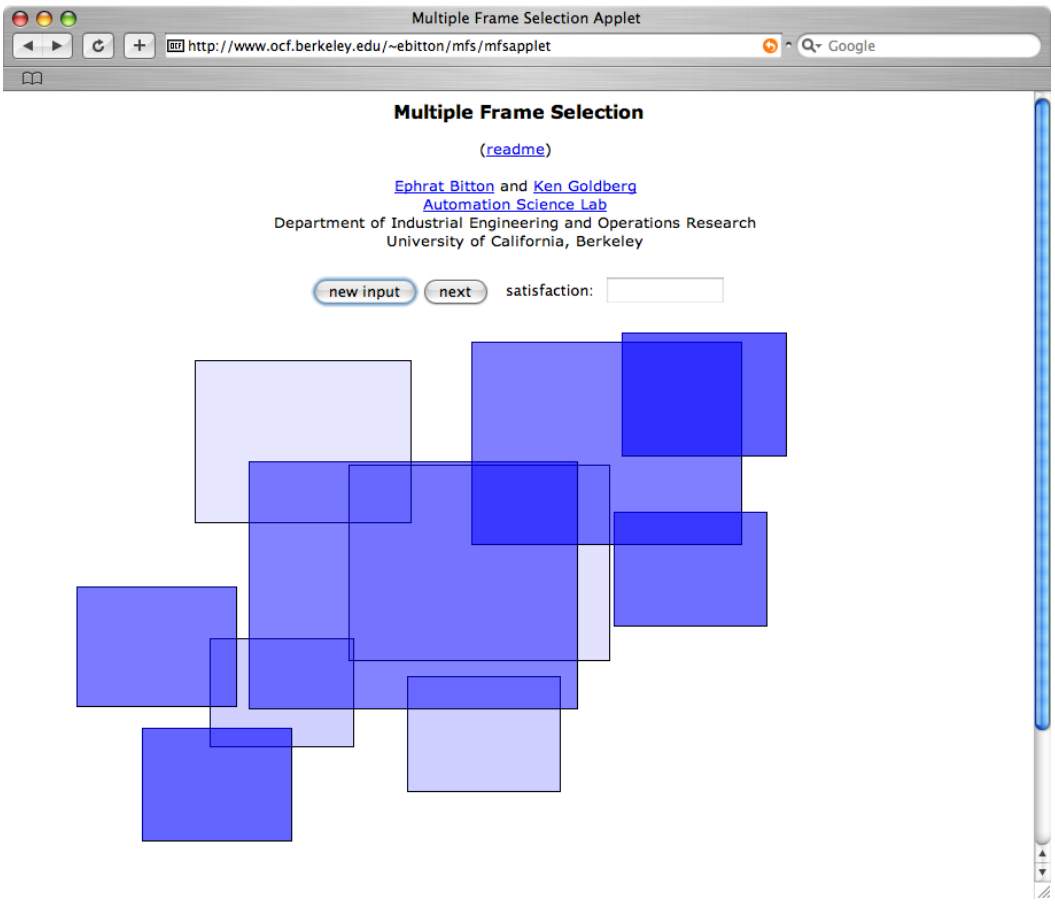


Figure 2.8: Screenshot of online simulator for the multiple frame allocation problem.

Proof. Given an arbitrary instance of IS $G = (V, E)$, we construct the following instance of the MFA problem. For every vertex $v_i \in V$ add a candidate frame f_i with a reward of 1 to F. For every edge $e_{ij} \in E$, we say that the area of intersection between f_i and f_j is greater than a, implying that the frames cannot both be part of a final solution to MFA.

We now claim that G has an independent set of size at least n if and only if there is a feasible set of n frames (whose total weight is at least n). For if G has an independent set with at least n vertices, then the corresponding frames in the MFA problem form a feasible set, and the sum of their weights is at least n. Conversely, suppose there is a feasible set F_n of n frames. Then the vertices in G corresponding to these frames will form an independent set with cardinality n, since no two frames f_i and f_j can be in F_n if there exists a corresponding edge $e_{ij} \in E$. □

Unfortunately, the maximization (optimization) version of Independent Set is not only NP-hard, it is also NP-hard to approximate. (41) This motivates the design and use of a heuristic solution for this problem that can be found quickly with reasonable results.

$a \setminus c$	0	0.02	0.04	0.06	0.08	0.10	0.12	0.14
0.02	6.47							
0.06	4.67	4.20	7.20					
0.10	4.83	5.17	5.00	6.17	11.03			
0.14	4.17	4.43	5.10	5.40	5.97	6.47	8.30	
0.18	4.30	3.57	5.40	4.87	7.13	5.40	5.10	9.27

Table 2.2: The average number of steps required for five agents to locate the subject using three UAVs and with termination threshold a and obstruction density c.

Heuristic Solution

To extend the single frame allocation algorithm to find the m best frames, we prioritize the frame requests using an m^{th} order exponentially decaying function of the agent's dissatisfaction as follows:

$$\rho_i = \frac{\alpha_i}{1 - 2^{-m}} \sum_{j=1}^{m} \left(\frac{1}{2}\right)^j \bar{s}_{i,t-j} \tag{2.9}$$

We then run the single frame allocation algorithm m sequential times, updating the priority of each frame request appropriately with each newly allocated frame. Figure 2.8 is a screenshot of an online simulator of the MFA problem, designed to give a visual feel for how the algorithm works. The "new input" button generates a random set of frame requests with randomly assigned authority levels. Once the frame requests have been generated, the "next" button is used to compute the next (approximate) best frame given the inputs. After the frame is computed and displayed to the user in the form of a red rectangle, the user can choose to accept the frame, and the corresponding priority levels of the frame requests are adjusted according to the MFA algorithm.

2.4.3 Step 3: Sensor Data Extraction

The sensor is a camera and image processing system. Given a frame specification, the UAV flies to the appropriate height and location and takes a photo with the camera. The photo is analyzed and a binary value {0,1} is returned, indicating 1 if the subject is detected in the frame and 0 otherwise. Since the size of the frame is related to the level of detail/resolution available to the image processing system, the sensor output value is based on two factors: 1) whether or not the subject is in fact located inside the frame, and 2) the accuracy of the sensor, which corresponds to the size of the frame and the density of obstructions in the underlying scene.

Let $r(f) = \frac{Area(f)}{Area(\Theta)}$ be the ratio of the area of a frame f to the size of the search space, so that $r(f) = 1$ if the frame is maximally large, and $0 < r(f) < 1$ for smaller frame requests. f either contains the subject or it does not. Let c_f be a value between 0 and 1 that corresponds to the average density of obstructions in frame f, and let

$B(f)$ be the binary sensor output. Conditioned on the frame containing the subject, we model $B(f)$ as a Bernoulli random variable, where the evidence is more likely to be accurate when the frame is small (the image is of high resolution) and the density of obstructions is low. According to agent i's distribution, the probability that the subject will be detected in frame f at time t is given by:

$$\begin{aligned} \mathrm{P}\left[B(f)=1\right] &= \left(1-c_f\right)\left(1-r\left(f\right)\right)\mathrm{P}_{i,t}(f) \\ &+ \left[1-\left(1-c_f\right)\left(1-r\left(f\right)\right)\right]\left(1-\mathrm{P}_{i,t}(f)\right) \end{aligned} \tag{2.10}$$

That is, if f contains the subject, the sensor returns 1 with probability $\left(1-c_f\right)\left(1-r\left(f\right)\right)$, and if f does not contain the subject, the sensor returns 1 with probability $1-\left(1-c_f\right)\left(1-r\left(f\right)\right)$. With this sensor model, we determine an upper bound on the acceptable frame size for termination $\bar{r}\left(f\right)$ by solving the following:

$$\begin{aligned} 1-a &\leq \left(1-c_f\right)\left(1-\bar{r}\left(f\right)\right) \\ \Rightarrow \bar{r}\left(f\right) &\leq 1-\frac{1-a}{1-c_f} \end{aligned} \tag{2.11}$$

Observe that frame f meets the size requirement for termination only when $a > c_f$.

2.4.4 Step 4: Updating Priors

As evidence is collected during each cycle, every searcher's individual, spatial probability distribution for the subject's location must be updated to account for new information.

Let f_t^* be the frame sampled at time t, and let B_t be the corresponding evidence collected. For agent k, we compute the probability that $\theta \in f_t^*$ by integrating over the marginals:

$$\mathrm{P}_{k,t}(f_t^*) := \mathrm{P}_{k,t}(\theta \in f_t^*) = \int_{f_t^*} \mathrm{P}_{k,t} \tag{2.12}$$

Bayes' rule can then be used to obtain the posterior probability that the subject is located within f_t^*, conditioned on the new evidence as follows:

$$\mathrm{P}_{k,t}(f_t^*|B_t) = \frac{\mathrm{P}_{k,t}(B_t|\theta \in f_t^*)\,\mathrm{P}_{k,t}(f_t^*)}{\mathrm{P}_{k,t}(B_t)} \tag{2.13}$$

Once each searcher's posterior distribution has been updated, we can incorporate the evidence from the next frame (f_{t+1}^*) in a similar fashion. Our prior distribution is now given by the posterior from the first step, and we use Bayes' rule and the information quality model to find the new posterior distribution.

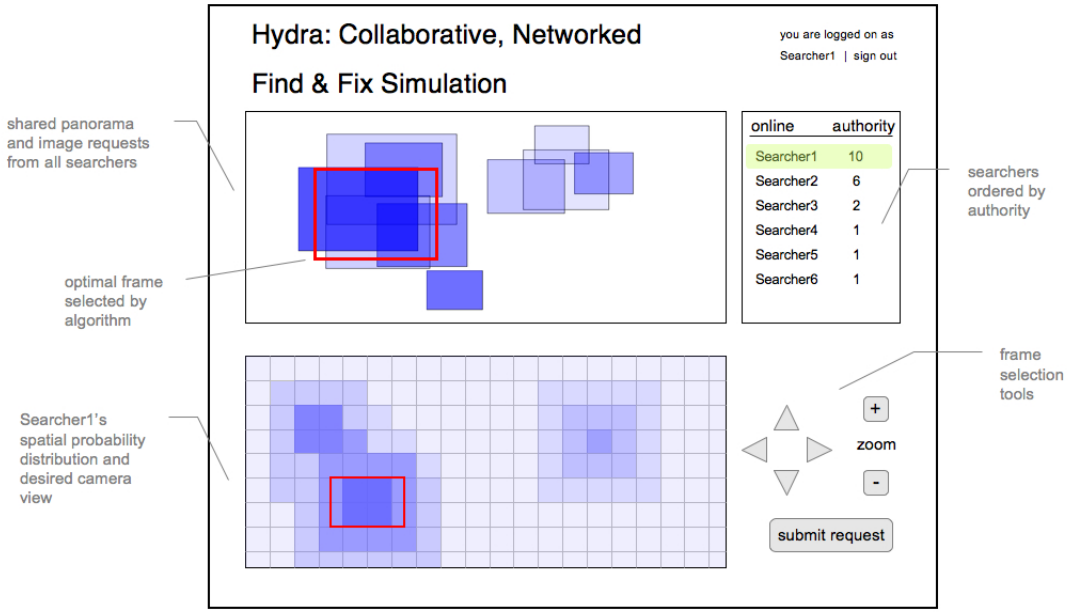


Figure 2.9: Mockup of graphical user interface for the Hydra mixed-initiative search and rescue framework.

2.5 User Interface for Collaborative Control

A crucial component to the success of the Hydra system is the design of an effective graphical user interface (GUI) for allowing both human and automated agents to collaboratively control the actions of the fleet of UAVs. Figure 2.9 illustrates a mockup of the user interface we designed to allow for this. The main screen is partitioned into an upper portion and a lower portion. In the upper portion of the screen is a shared panorama view of all the image requests received from all of the agents (both human and automated). The opacity of a frame is directly proportional to the authority of the agent who made the request. Since opacity is cumulative, the darkest regions of the panorama indicate the "hottest" or most requested areas (weighted by authority). A red rectangle is used to indicate the frame that was selected by Hydra to explore next based on all of the frame requests received.

The lower portion of the screen is unique to each agent and represents that agent's spatial probability distribution of the subject's location in the search space. Once initialized, this distribution is automatically updated by Hydra based on any new evidence or information received via the UAVs. Agents can use the control panel to the right of this panorama to zoom in or out and navigate the space. They can submit a new frame request by highlighting a rectangular subregion of interest on the probability distribution panorama. Upon clicking on the "submit request" button the frame request is sent to the central server and processed by the Hydra system. To the right of the panorama screen is a list of all agents actively using the system and their associated

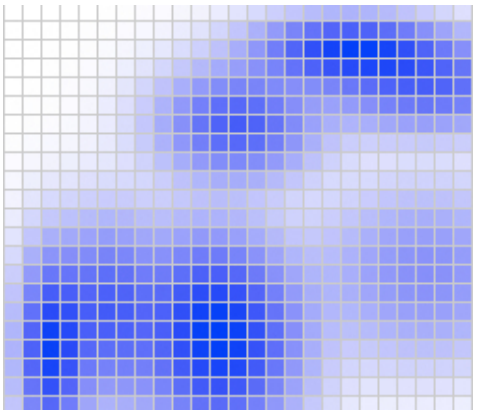


Figure 2.10: Example of a spatial prior distribution to be maintained by either a human or automated agent. The darker the cell the higher the probability that the subject is located within.

authority levels.

A possible addition to this basic mockup is an option for agents to review the images and data captured by the UAVs out in the field. A fully functional system would also require an easy way for human agents to initialize their spatial probability distributions. Figure 2.10 illustrates an example spatial probability distribution with which a human agent would work. For ease of visualization, darker cells indicate a higher probability that the hidden subject is at that location. We believe that this sort of visualization is more effective for humans to work with than actual numbers, as it is significantly easier to determine which areas require further investigation.

2.6 Experimental Results

We are interested in determining how quickly the search strategy described for automated agents can locate the hidden subject. We created a scenario with a subject hidden uniformly at random in a 25×25 grid with five automated agents of equal authority and three UAVs; additionally, we randomly generated a different prior distribution for each agent, which we used to seed each run of the simulation. We tested the agents' combined performance 30 times each for a range of termination thresholds (a) and constant obstruction densities (c). We limited the first (information-seeking) phase of the automated agent frame request algorithm to three steps and set an upper bound of 200 iterations. Table 2.2 and Figure 2.11 reflect the average number of iterations required for the runs that successfully terminated within 200 iterations. Approximately 6.93 percent of the simulation runs diverged and were unable to locate the subject; this behavior was particularly pronounced when the difference between a and c was small (i.e. when we require a smaller frame to terminate).

We derive an upper bound on the expected number of steps for a single automated agent to detect the subject by following a naïve frame request algorithm. Let a be the pre-specified termination threshold, and let c be the constant density of obstructions

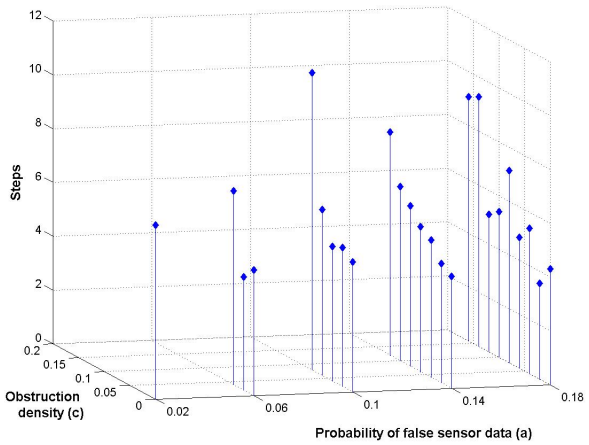


Figure 2.11: Average number of steps to termination as given in Table 2.2. The number of steps increases when we require a smaller frame to terminate.

across Θ. Then the largest acceptable frame for termination has area $\bar{r} = 1 - \frac{1-a}{1-c}$. If we only consider frames with area equal to $\bar{r}$, then we can sweep the entire search space with $1/\bar{r}$ frames, and we terminate with the first frame in which the subject is detected. To find an upper bound on the number of frames we must take before the subject is detected, assume that we continue search until the termination conditions are met *and* the subject is truly located. Let f^* be the region of the search space in the partition of $1/\bar{r}$ frames that contains the subject. Sampling from the frame is equivalent to sampling from a geometric distribution with a probability of success equal to $(1-c)(1-\bar{r}) = 1-a$. Hence, the expected number of samples of just frame f^* required before the subject is detected is $\frac{1}{1-a}$, and

$$\mathrm{E}\left[\text{time to truly locate subject}\right] = \frac{1}{\bar{r}}\left(\frac{1}{1-a}\right) \tag{2.14}$$

To compare the maximum information gain algorithm against the naïve search strategy, we ran each simulation until an appropriately sized frame was found that truly contained the subject. In our experiments we observed that on average the number of steps taken by the maximum information gain frame request algorithm is 65% fewer than the expected number of steps required by the sweep strategy, with a standard deviation of 15.8%. In experiments with three automated agents and a single UAV, we observed a 57.7% improvement over the naïve search strategy.

2.7 Conclusion and Future Work

In this chapter we describe a framework for collaborative control for visual search applications that is designed to accommodate different models for sensor data extraction, agent authority hierarchies, prior distributions, and termination conditions.

Future work in this area will require extensive experiments with both automated and human agents to verify our simulation results. We will also seek to extend the frame allocation algorithm to account for the current positions of the UAVs and the cost of travel when computing the optimal set of frames for the UAVs to explore. This will require the incorporation of path planning and scheduling algorithms for the UAVs.

We will also investigate other applications for the the Hydra framework, including an automated version of the Collaborative Observatories for Natural Environments (CONE) project. CONE (`http://cone.berkeley.edu`) is a framework that allows the general public to contribute to scientific research (a newly popular activity otherwise known as "citizen science") by tele-operating a robotic camera to observe, record, and index animal activity in a remote environment. The most recent installment of CONE is stationed at the Welder Wildlife Refuge in Texas. Excluding tropical regions, this refuge sees the greatest variety of bird species in the North American continent.

The system is accessible for free to the general public. Similar to search agents using Hydra, a user can send an image or frame request tot he camera by highlighting the corresponding rectangular region of interest on a panoramic image of the scene. In the event that more than on requests are received at the same time, the single frame allocation algorithm is used to determine the frame that minimizes the mean and variance of time-dissatisfaction across all requests, as described in (29).

Although CONE has proven to be a popular service and often has a handful of users online at any given time, there are periods of time where the camera is not in use and not actively recording data. It is during these times that we require the use of an automated procedure to intelligently capture images of the scene that contain wildlife activity. Formally, we seek to determine a set of n frames $F = \{f_1, \ldots, f_n\}$ that maximizes total expected reward, where reward is a pre-defined function. The behavior of this objective is entirely dependent on the chosen reward model for capturing images of each species. For example, if the reward for capturing an image of a particular species is inversely proportional to the frequency at which the species is naturally observed, then this objective translates to finding the set of frames that will maximize the likelihood of capturing images of more elusive species. If, on the other hand, rewards are directly proportional to the frequency at which birds are naturally observed, then the objective becomes to maximize the probability that *any* bird is found.

Let S be the set of all possible bird species that may be observed by the CONE camera, and let α_s be the reward for capturing an image of a bird from species $s \in S$. P_{is} is the probability that frame f_i contains a bird of species $s \in S$. We assume that birds of any species can enter and leave the search space at any time. Let binary variable f_i be defined as follows.

$$f_i = \begin{cases} 1 & \text{if frame frame } i \text{ is selected} \\ 0 & \text{otherwise} \end{cases}$$

The number of times we expect to observe species s in a set of frames $F = \{f_1, \ldots, f_n\}$ is $\sum_{f_i \in F} P_{si}$. Hence, the expected reward for taking frame f_i is given by

$$E\left[\text{reward for taking frame } f_i\right] \;=\; \sum_{s\in S} \alpha_s P_{si} \tag{2.15}$$

Our objective is therefore to find a set of frames $f_1, \ldots, f_n$ that maximizes the total expected reward, as follows.

$$\begin{aligned} \max \quad & \sum_{i=1}^{m} \sum_{s\in S} \left[\alpha_s P_{si}\right] f_i \\ \text{subject to} \quad & \sum_{i=1}^{m} f_i = n \\ & f_i \in \{0,1\} \end{aligned}$$

Observe that if we maximize the stated objective function without any additional constraints, the solution will always be of the form $f_1 = f_2 = \cdots = f_n$. To avoid this, we will need to place bounds on the amount of acceptable overlap between any pair of candidate frames. In the simplest case, we can require that all frames are mutually exclusive; that is, the area of intersection between any two frames must be 0. To keep the problem general, we allow for the maximum area of overlap to be a, and we introduce limits on the size of a frames. The multiple frame selection problem (MFS) now becomes the following constrained optimization model.

$$\begin{aligned} \text{(MFS)}\ \max \quad & \sum_{i=1}^{m} \sum_{s\in S} \left[\alpha_s P_{si}\right] f_i \\ \text{subject to} \quad & \sum_{i=1}^{m} f_i = n \\ & \text{Area}\left(f_i \cap f_j\right) \le a \qquad \forall\, i \neq j \\ & l\left(f_i\right) \le \text{Area}\left(f_i\right) \le u\left(f_i\right) \\ & f_i \in \{0,1\} \end{aligned}$$

This problem is nearly identical to the problem considered by Hydra, with the exception that the feedback loop is open instead of closed. Since it has proven to be extremely difficult to identify and classify birds in an image using supervised learning algorithms, we are unable to analyze the images collected and give reliable information back to the system regarding the content of the images. Hence, a realistic solution to this problem would require the collection of a large amount of images, which would be analyzed by human volunteers at a later point in time. Turning the classification process into a game has been shown to be a successful strategy for motivating people to volunteer their time (e.g. Games with a Purpose), as has small monetary awards on Mechanical Turk.

3

Information Filtering with Network Flows

3.1 Introduction

This chapter explores information filtering methods based on graph theoretical techniques, with specific application to gene expression analysis. We begin by motivating the problem and defining it mathematically. We then discuss traditional methods used by biologists to solve the problem, and then present and analyze a new graph-theoretical method.

Although powerful microarray technology has provided scientists with the ability to study large-scale changes in the expression levels of tens of thousands of genes simultaneously, the major barrier still remains: how to identify which handful of genes are the major participants in the biological phenomenon studied, in that they exhibit a significant difference in expression levels, in response to pre-determined test conditions. This is otherwise known as the Gene Selection Problem. The difficulty in solving this problem stems from the incredibly large amount of noise in the data. Some noise can be attributed to the position of the probe along the gene, which could cause two probes of the same gene to report significantly different expression levels. Even more noise is caused by biological factors unique to individual tissue samples or specimen, which cannot be modeled or predicted. To illustrate the extent of this problem, if a method for identifying differentially expressed genes from a set of 30,000 genes was to yield a false positive rate of one percent, then on average around 300 false positives will be expected. While for many applications a false positive rate on this scale would be considered quite excellent, in this case this rate nearly prevents the ability to identify the top 50 or 100 significant genes.

A myriad of statistical techniques have been developed to combat this problem, many relying on hypothesis testing. In a typical *class comparison* study, one set of tissue samples or specimen are subjected to pre-determined controlled conditions, and a second set is subjected to some experimental conditions. The gene expression values are then measured across all biological replications of both conditions. For each gene

measured, the *null hypothesis* is that the gene did not show a significant difference in expression levels, and the research hypothesis is that the difference is in fact significant.

In this chapter, we present a new approach to the Gene Selection problem that combines statistical hypothesis testing methods with partitioning models from graph theory, which is a subfield of computer science. Graph theory allows us to consider the intricacies of the structure of the data and relationships between samples beyond what hypothesis testing can do alone. It provides extremely flexible and customizable models for finding partitions of data sets that meet user-specified criteria. Here, we consider one of the simplest graph partitioning models, known as the Minimum Cut, for ranking genes according to the degree to which they are differentially expressed between the control and test experiments.

We evaluate this model on an expression microarray data set collected from a series of experiments on female transgenic mice. Specifically, we are interested in shedding light on the mechanism by which uterine dendritic cells contribute to embryo implantation in mice. Uterine dendritic cells (uDC) can be depleted using a transgenic mouse model by which cells expressing CD11c, a marker known to be selectively expressed on dendritic cells, can become sensitive to Diphtheria toxin (DTx) which will cause their elimination. In these mice, the transgenic expression of the human diphtheria toxin receptor (DTR) under the CD11c promoter will allow depletion of the CD11c- positive mouse cells after intraperitoneal administration of DTx. DC are depleted within 8 hours after administration of the DTx and will remain absent from the tissue for two days (83), which is the time span of the embryo implantation process in mice. uDC were previously shown to be critical for mouse embryo implantation (110). They were shown to have a role in the uterine tissue remodeling that will allow the uterus to be receptive towards the embryo and will therefore allow the embryo to implant into the uterine wall. In this work, the role of uDC was shown to be independent of their classical immunological role as antigen presenting cells to T cells and therefore these cells were not involved in regulation of maternal tolerance towards the embryo. One proposed role was the contribution of uDC to the uterine decidualization process, which is characterized by the proliferation and differentiation of the uterine stroma into a spongy mass of cells called decidua, which will allow the uterus to be receptive towards the embryo and will sustain the embryo until the placenta is formed. It was also suggested that uDC could possibly exert their critical role in the uterine decidulaization process by induction of uterine angiogenesis (i.e. the development of new capillaries from existing ones). However, the mechanism has yet to be fully elucidated. An attempt to shed light on the exact mechanism by which uDC promote the decidualization process was exerted by generating an expression array data comparing uterine samples of normal mouse embryo implantation sites to uDC-depleted ones and was subsequently analyzed using Graph Cut (GC).

In order to validate the significance of the biological results yielded by the Graph Cut model, as well as examine the robustness of this model versus others, we also analyzed these data using the CLICK algorithm and the popular LIMMA software package. The expression array data set was analyzed using all three methods for the

20 most significantly differentially expressed genes between experimental and control groups. Overall, analysis of this expression array data set using GC allowed us to discover factors that were specifically down and upregulated upon uDC depletion during the two days of embryo implantation in mice more reliably than with the other two models examined. Specifically, the results yielded that upon uDC depletion just prior to embryo implantation in mice (achieved by DTx administration on embryonic day, E3.5), a significant upregulation of inflammatory cytokines was exhibited on E4.5, along with downregulations of genes related to embryo implantation or uDC regulation. On E5.5, some of the genes shown to be downregulated using GC on E4.5, also persist on E5.5 along with other genes, all downregulated and mostly related to embryo implantation, both in humans and in mice.

3.2 Problem Formulation

In this section we give a formal description and discussion of the gene selection problem and its connection to graph theory. We then provide a review of related literature, including both traditional (statistical) methods for identifying differentially expressed genes and grapth theoretical approaches. We conclude with a description of our proposed Graph Cut (GC) method, which builds on graphical models of separation, and the biological experimental setup describing the expression array data analyzed using GC as compared to other algorithms.

3.2.1 The Gene Selection Problem

Our goal is to identify the genes that exhibit the most significant change of expression levels between the control groups and the test groups of mice. We assume that we know which tissue samples correspond to which groups, and we assume we have the following data available as input to our problem:

- m groups of various test and control experiments
- n_i experimental repetitions from group i; let $n = \sum_{i=1}^{m} n_i$
- $\mathcal{G}$ is the set of genes probed
- $\mathcal{E}$ is a $|\mathcal{G}| \times n$ matrix of gene expression levels for all genes in $\mathcal{G}$ and n tissue samples

Our desired output is a ranking of the genes in $\mathcal{G}$ according to some measure of significance. Ideally, the most differentially expressed genes will help better understand the factors at play when uterine dendritic cells are present or not during embryo implantation.

To evaluate our results and compare them with the outputs of other algorithms, we consider the top 20 genes as a representative sample. Theoretically, these are the genes that should be the most differentially expressed, and a robust algorithm should be able

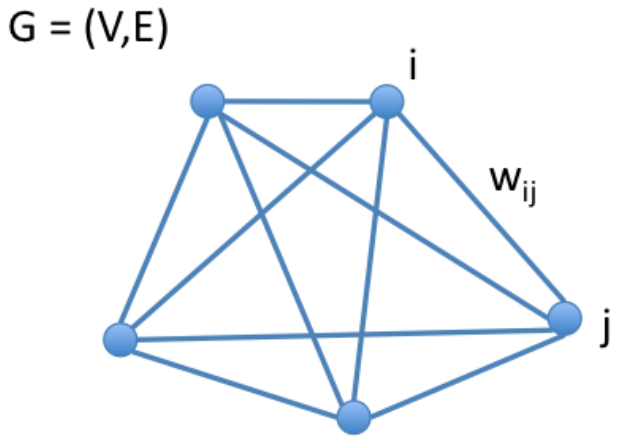


Figure 3.1: Example of a complete, undirected graph with edge weights. A separate graph is built for every gene $g \in \mathcal{G}$, where each node corresponds to a different tissue sample, and edge weights reflect the similarity in expression levels between two samples.

to determine the most significant genes. From the biological viewpoint, we are looking for genes that are biologically relevant to the mechanisms at play in the experiment. This may include groups of genes that are part of the same family that are either up or downregulated *together*. Since the effect of the DTx is to deplete dendritic cells, which are part of the immune system, we are also looking for genes that are immunologically relevant. Moreover, since we already know from previous work that uDC are critical for the succsess of embryo implantation, we will also be expecting to find downregulation of genes which promote adequate implantation upon uDC depletion. The ultimate goal when looking at an array is to find a pathway (or a few pathways) that helps explain the experimental results . In this particular study, we seek to determine the impact of depleting uDC in mice. In terms of the phenotypes observed when uDC are depleted from the uterus, we know that the uterus does not respond correctly to the embryo. Hence, our goal is to identify any genes that could be related to or explain that phenotype and could ultimately guide us to the critical pathways by which uDC exert their action on the uterus to allow adequate embryo implantation that can be later pursued and confirmed by experimental biological work.

3.2.2 Modeling Gene Expression Data on a Graph

For each gene, we can model relationships between tissue samples on a graph $G = (V, E)$, where each tissue sample corresponds to a different vertex $v \in V$ on the graph. There is an edge $e_{ij} \in E$ between every pair of samples i and j; that is to say, the graph is *complete*. Each edge $e_{ij} \in E$ carries a weight w_{ij} that reflects the quantified similarity S_{ij} between sample i and sample j. (See Figure 3.1.)

The similarity between two vertices i and j in a graph is determined by a function

that takes as input a feature or observation vector x_i for vertex i and another, x_j, for vertex j. In our application, x_i contains the observed gene expression level for tissue sample i. The function outputs a single real-valued number, where larger numbers indicate a higher degree of similarity between i and j. Depending on the properties of the feature vectors, a variety of similarity functions can be used. The most commonly used similarity metrics are measures of correlation, such as Pearson's correlation or Kendall's tau rank correlation (85); these functions require feature vectors of length greater than one. Euclidean distance or l^2-norm is a common measure of dissimilarity, where larger distances correspond to greater degrees of dissimilarity. The Gaussian similarity function transforms the Euclidean distance measure to a similarity function as follows:

$$S_{ij} = S(x_i, x_j) = \alpha \exp\{-\beta ||x_i - x_j||^\gamma\} \tag{3.1}$$

We opted to use this similarity function due to its intuitive behavior and flexibility, however, any alternative monotonically increasing function in $||x_i - x_j||$ may be used in its place.

3.2.3 Cuts on a Graph

We now provide a formal definition for cuts or partitions on a graph, which is a major component of our proposed gene selection method.

Let $G = (V, E)$ be an undirected graph, where V is the set of nodes and E is the set of edges connecting nodes in the graph. Using common notation, let $n = |V|$ be the number of nodes and $m = |E|$ the number of edges in G. The weights of the edges in the graph are denoted by w_{ij} for every $[i, j] \in E$.

A bipartition of the graph is called a *cut*, $\left(S, \bar{S}\right) = \{[i, j] \mid i \in S, j \in \bar{S}\}$, where $\bar{S}$ is the complement of S ($\bar{S} = V \setminus S$). See Figure 3.2 an illustration. We define the *capacity of a cut* $\left(S, \bar{S}\right)$ as

$$C\left(S, \bar{S}\right) = \sum_{i \in S, j \in \bar{S}} w_{ij} \tag{3.2}$$

More generally, for any pair of sets $A, B \subseteq V$, we define the set of edges going between these two sets as $(A, B) = \{[i, j] \mid i \in A, j \in B\}$, and the capacity of (A, B) is $C(A, B) = \sum_{i \in A, j \in B} w_{ij}$. We define the *capacity of a set* $A \subset V$ to be $C(A, A) = \sum_{i,j \in A} w_{ij}$, denoted by $C(A) = C(A, A)$.

The problem of partitioning a graph into two nonempty components that minimize the capacity of the cut is called the *minimum 2-cut* and is polynomial time solvable (103). Given a partition of a graph into k disjoint components $\{V_1, \ldots, V_k\}$, the k-cut value is $C(V_1, \ldots, V_k) = \frac{1}{2} \sum_{i=1}^{k} C\left(V_i, \bar{V}_i\right)$. The problem of partitioning a graph into k nonempty components that minimize the k-cut value is called the minimum k-cut and is polynomial time solvable for fixed k (63; 72; 73).

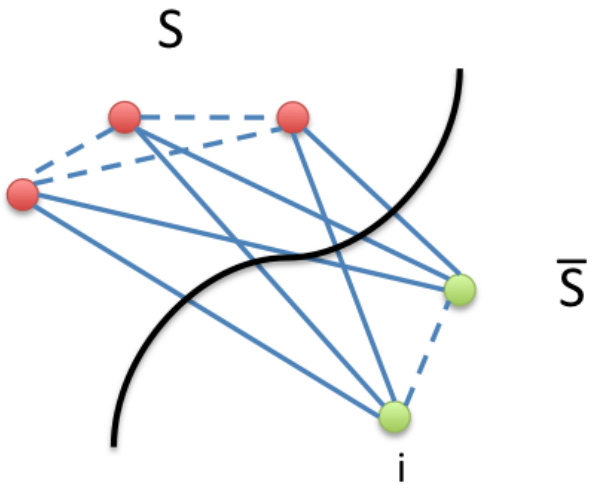


Figure 3.2: Example of a cut on an undirected, complete graph. The cut is indicated by the dark black line that partitions the node set V into two disjoint sets: S and $\bar{S}$. The capacity $C\left(S,\bar{S}\right)$ of the cut is the sum of the weights of the edges that cross the cut (that is, the sum of the weights of all edges that have exactly one endpoint in S and one in $\bar{S}$). $d\left(i\right)$ is defined as the sum of the weights of the edges adjacent to node i.

Let $d_i = \sum_{[i,j]\in E} w_{ij}$ denote the sum of edge weights adjacent to node i. The weight of a subset of nodes $B \subseteq V$ is denoted by $d\left(B\right) = \sum_{j\in B} d_j$ is referred to as the *volume* of B.

The *normalized cut* (NC) problem was introduced by Shi and Malik (131) in their work on image segmentation. This model is an alternative to finding a minimum cuts on a graph, and it is designed to find a bipartition that is more balanced. Formally, the normalized cut is defined as

$$\min_{S\subset V} \frac{C\left(S,\bar{S}\right)}{d\left(S\right)} + \frac{C\left(S,\bar{S}\right)}{d\left(\bar{S}\right)} \tag{3.3}$$

By construction of this objective function, the ratio with the smaller value of $d\left(\right)$ will dominate the objective value; therefore, this objective function drives the segment of S and its complement to be approximately of equal size. This problem was shown to be NP-hard by reduction from the set partitioning problem (131).

In this chapter, we make use of the 2-cut model for identifying the most differentially expressed genes, though it can be interchanged with other cut models such as NC or those discussed in (73).

3.3 Related Work

In this section we discuss both traditional methods used to solve the Gene Selection problem as well as recently developed techniques based on graph theoretical models.

3.3.1 Traditional Methods

One of the first models developed for identifying differentially expressed genes is known as the *fold change* (142). Let $\bar{\mathcal{E}}_g^c$ be the average expression level recorded for gene $g \in \mathcal{G}$ under the control conditions, and let $\bar{\mathcal{E}}_g^t$ be the average expression level recorded for gene g under the test condition. Then the fold change is defined as:

$$FC = \frac{\bar{\mathcal{E}}_g^t}{\bar{\mathcal{E}}_g^c}. \tag{3.4}$$

In many cases, the result is log-transformed to obtain a more symmetric distribution and is referred to as the *log fold change*. It was decided, albeit somewhat arbitrarily, that a gene exhibiting a fold change greater than 2 (either up or down) satisfies the research hypothesis that the difference in its expression levels is significant. For many years, genes were therefore ranked and filtered according to their fold change values. However, there are serious problems with this method resulting from the fact that the variances of the expression values are not considered; this can result in an undesirable increase of false negatives as well as false positives.

In recent years, more sophisticated techniques relying on statistical hypothesis testing have been employed. Each gene is considered independently with a corresponding null and research hypothesis. The most straight-forward statistical test used is the t-test, which compares the means of the test and control expression values and simultaneously considers the variability in the data. As pointed out in (142), if the standard deviation happens to be very close to zero, the t-statistic can be artificially inflated and result in a false positive finding. The reader is referred to (134; 142) for a more in-depth discussion of these techniques.

Furthermore, it cannot be assumed that expression levels across genes are independent due to the inherent complexities of biological systems. Multiple comparison correction methods such as Holm's (77) and the False Discovery Rate (7) may be used to account for these dependencies; these techniques have the added advantage of letting the user specify an acceptable level of false positives.

3.3.2 Spectral and Graph-Based Methods

In this section we give a detailed description and analysis of the CLICK method for clustering genes based on their expression values, against which we compare our own method in our experimental analysis. We also provide a brief description of alternative graph-based methods that have been proposed and the problems with these models.

CLuster Identification via Connectivity Kernels (CLICK). The CLICK algorithm (129; 130) is a cut-based method designed for clustering gene expression

data. Generally, the objective of a clustering problem in this domain is to identify groups or *clusters* of genes that exhibit similar behavior across different test and control scenarios. Since in this application the authors assume some prior knowledge on the structure of the correct clustering, they build this information into an edge weighting scheme that uses probability models. The graph is then recursively bi-partitioned with minimum cuts until no further change can be made without violating a pre-specified stopping criterion. The result is a set of tight clusters or *kernels* that are then merged in the second phase of the algorithm until certain additional criteria are met.

CLICK can be adopted for use in the gene selection problem by using its similarity model to determine when the expression levels of one group of experimental repetitions is sufficiently different from the expression levels of a second group. Specifically, a weighted similarity graph $G_g = (V, E)$ is constructed for each gene $g \in \mathcal{G}$. Each node in V corresponds to a different tissue sample from the set of control and test observations, and each pair of nodes is connected by an edge. When two samples or nodes i and j belong to the same experimental condition (i.e. either both are from the control group or both are from the test group), then i and j are said to be *mates*. Let p_{mates} be the probability that any two randomly selected samples are mates.

The CLICK clustering method makes five key assumptions: 1) The similarity between mates is normally distributed with mean μ_T and variance σ_T. 2) The similarity between non-mates is normally distributed with mean μ_F and variance σ_F. 3) The nodes in the graph have pairwise independent mate relationships. 4) For a cut $C(A, B) \in G$, the similarity random variables $\{S_{ij}\}_{i \in A, j \in B}$ are pairwise independent conditioned on the fact that all element pairs are either mates or non-mates. 5) Kernels must have at least k elements, where k is a parameter set by the user.

Let $f(S_{ij}|\mu_T, \sigma_T)$ be the mates probability density function evaluated at S_{ij}, and let $f(S_{ij}|\mu_F, \sigma_F)$ be the non-mates probability density function evaluated at S_{ij}. The weight given to edge $[i, j] \in E$ considers the probability that sample i and sample j are mates and is defined as

$$\begin{aligned} w_{ij} &= \log \frac{\Pr(i,j \text{ are mates } |S_{ij})}{\Pr(i,j \text{ are non-mates } |S_{ij})} \\ &= \log \frac{p_{\text{mates}} f(S_{ij}|\mu_T, \sigma_T)}{(1 - p_{\text{mates}}) f(S_{ij}|\mu_F, \sigma_F)} \\ &= \log \frac{p_{\text{mates}} \sigma_F}{(1 - p_{\text{mates}}) \sigma_T} + \frac{(S_{ij} - \mu_F)^2}{2\sigma_F^2} - \frac{(S_{ij} - \mu_T)^2}{2\sigma_T^2} \end{aligned} \tag{3.5}$$

The CLICK algorithm uses this weighting scheme to recursively bipartition vertices of the graph into disjoint sets S and $\bar{S}$ using minimum cut; they show mathematically that when the capacity of the minimum weight cut is greater than 0, then all elements in the current subgraph are most likely to be mates and belong in the same cluster or group.

Under their assumptions, we can apply the CLICK framework to the gene selection problem by creating a separate graph for each gene and computing the cost of the cut

between the test and control groups. Sorting the genes in ascending order according to their cut capacities will yield the genes with the greatest probability of belonging to separate groups at the top of the list; that is, these genes will have exhibited the greatest difference in expression levels as determined by the CLICK model.

There are two primary issues with the CLICK algorithm that can affect the quality of the result.

1. While constructing the graph to have negative edge weights by taking the log-likelihood of the probability ratio makes the mathematics cleaner, it is not immediately clear why this step is necessary other than that it is standard practice in genetics. This decision alone makes their model NP-hard to solve completely, and thus introduces the need for a crude approximation scheme.

2. The CLICK algorithm requires the use of several thresholds for the processes of adopting singletons into kernels and merging kernels into clusters. Such thresholds can drastically increase the difficulty of finding a good clustering in practice, as much fine-tuning is required. While it may not always be possible to completely eliminate the need for thresholds, we might be able to minimize the number required and their added sensitivity to the final outcome.

Other Graph-Based Methods. Recently, there have been a number of other graph-based techniques developed for interpreting gene expression data, particularly as a way to identify groups or clusters of genes that behave similarly. Xu et al. (156) use minimum spanning trees (MST) to identify clusters of genes, where a minimum spanning tree of a graph is defined as a subset of the edges of minimum total weight such that all nodes in the graph are still connected by a path. These trees can be found very efficiently in polynomial time. Xu et al. define a cluster C to be a subset of nodes (or in this case, genes) such that in any arbitrary partition of C into subsets C_1 and C_2, the closest node $v \notin C_1$ to any node in C_1 belongs to C_2. They then show that for any two nodes v_1 and v_2 in the same cluster, the nodes along the unique path from v_1 to v_2 in the MST are in the same cluster. Hence, their method reduces to a graph partitioning problem, where the resulting partitions correspond to clusters of genes. However, MSTs are a very crude model of the clustering problem: defining a cluster in terms of distance to the nearest neighbor in a cluster allows two items in the same cluster to be arbitrarily far apart from each other and can often lead to undesirable outcomes.

JointCluster (104) is another graph-based gene clustering algorithm. It uses normalized cuts (NC) to recursively bipartition the graph to form a hierarchy of clusters. The algorithm is designed to cluster several graphs simultaneously in a way that provides reasonable and consistent results in each graph as well. This is analogous to clustering on a single graph with multi-dimensional data. Since finding a normalized cut is NP-hard, this method relies heavily on heuristic techniques. Specifically, the authors use a variation of common heuristic for solving the NC problem, which is a *spectral* or eigenvector-based technique (84).

There are several other graph-based gene clustering algorithms that have been developed using the Normalized Cut graph-partitioning model proposed in (131). CLIFF (155) was one of the first of such methods; it works by iterating through a two-part process of filtering irrelevant features and clustering on the remaining features until the clusters converge. Fowlkes et. al (47) extend the bi-partitioning technique used by CLIFF to a recursive k-way partitioning algorithm, where k is chosen at each iteration to minimize the k-way normalized cut objective. Verma and Meila (144) give a comparison study of several spectral clustering techniques on three different data sets, one of which is gene expression data. They find that the algorithms behave quite similarly in both artificially constructed scenarios and on real data.

Dhillon et. al (36) show that the normalized cut objective function can be derived as a special case of the weighted kernel k-means objective function; subsequently, they show that eigenvector-based approximations are not necessary for minimizing normalized cuts. JointCluster (104) uses normalized cuts to recursively bipartition the graph to form a hierarchy of clusters. The algorithm is designed to cluster several graphs simultaneously in a way that provides reasonable and consistent results in each graph as well. This is analogous to clustering on a single graph with multi-dimensional data. Since finding a normalized cut is NP-hard, this method relies heavily on heuristic techniques. Hu et. al (78) use normalized cuts as part of a scalable algorithm for mining dense, large subgraphs.

All of the above methods using normalized cut for clustering gene data are spectral methods; that is, due to the computational complexity challenges of normalized cut, these algorithms use eigenvalue-based techniques as a way to find an approximate solution. An excellent survey on both kernel and spectral methods for clustering, including the use of normalized cuts, can be found in (43).

3.4 Graph Cut: A Hybrid Graph and Statistical Algorithm for Gene Selection

In this section we describe a new hybrid graphical and statistical method for identifying genes that are differentially expressed between a set of control and a set of test experiments.

3.4.1 Preprocessing

Before we can analyze the raw expression data, we preprocess it in two steps. First, we normalize the data so that the changes in expression levels can be compared across genes. The second step involves filtering the genes so that we only consider those that show a statistically significant difference in expression levels between the test and control groups.

3.4.1.1 Step 1: Normalizing Raw Expression Values.

Looking at the raw expression profiles for each gene, the numbers can vary wildly. The expression levels for some genes can be several orders of magnitude larger than the expression levels for others; consequently, in their raw form expression levels are incomparable across genes. To correct for this, we first normalize the raw expression levels for each gene $g \in G$ as follows. Let $\mathcal{E}_g^{max}$ be the largest expression value recorded for gene g across all tissue samples. Then the normalized expression value for gene g of tissue sample j becomes

$$\mathcal{E}_{g,j} \leftarrow \frac{\mathcal{E}_{g,j}}{\mathcal{E}_g^{max}}. \tag{3.6}$$

According to this normalization scheme, the expression levels for every gene $g \in G$ now range between 0 and 1, and every gene will have an expression level equal to 1 for at least one tissue sample (across all m groups and N samples).

3.4.1.2 Step 2: Filtering Insignificant Genes with Hypothesis Testing.

Once the gene expression data has been normalized, we remove the genes from the data set that do not have a statistical difference in expression levels between the test and control experimental groups. Formally, for each gene we separate the tissue samples into two groups of experimental repetitions: test and control. We then run an unpaired Student's t-test on these groups to determine whether the null hypothesis that the expression values were drawn from the same distribution should be accepted. This test assumes that the two groups being compared have the same variance and that each expression value was sampled independently. Recall that $\bar{\mathcal{E}}_{g,T}$ and $\bar{\mathcal{E}}_{g,C}$ are the average expression values for the test and control groups sampled for gene $g \in \mathcal{G}$, respectively. Let $\sigma_{g,T}^2$ and $\sigma_{g,C}^2$ be the respective variances of the test and controls groups for g, and let n_T and n_C be the number of repetitions in each group. Then the t-value for the Student's t-test can be found as follows.

$$t_g = \frac{\bar{\mathcal{E}}_{g,T} - \bar{\mathcal{E}}_{g,C}}{\sqrt{\frac{\sigma_{g,T}^2}{n_T} + \frac{\sigma_{g,C}^2}{n_C}}} \tag{3.7}$$

Given t_g, the corresponding p-value can be found using a look-up table of the Student's t-distribution. As is common practice, we accept the null hypothesis when the p-value returned by the t-test is greater than 0.05; in this case, it is determined that there is no significant difference in expression values between the test and control repetitions. Every gene for which the null hypothesis is accepted is subsequently removed from $\mathcal{G}$, the set of genes to be considered. This process prunes the set of genes to consider by about 92 percent.

3.4.2 The Graph Cut (GC) Method

We are given as input data from a class comparison study, where the goal is to identify the subset of genes that exhibit the greatest change in expression values between a controlled scenario and test scenario. The Graph Cut (GC) model we propose in this section uses the 2-cut model described above to compare genes based on their respective separations between test and control repetitions. Note, however, that this can be replaced with any other cut-based model, such as Normalized Cut. Before running our model, however, the user must select an appropriate similarity function for comparing expression values as well as a graph partitioning or cut model.

Let A be the known set of repetitions corresponding to the control experiments, and let B be the known set of repetitions corresponding to the test experiments. Our method is described as follows.

1. Preprocess the data by first normalizing the expression values for each gene so that they range between 0 and 1, and then filter out the genes that do not show statistically significant differences between the control and test scenarios. Let $\mathcal{G}'$ be the set of remaining genes.

2. For each gene $g \in \mathcal{G}'$, define a complete graph $G_g = (A \cup B, E)$ with edge weights defined by the similarity function $S(x_i, x_j)$. Compute the capacity of the cut between the control and test nodes: $C_g(A, B)$.

3. Sort genes in ascending order according to the cut capacities from the previous step.

3.5 Case Study: Effect of Dendritic Cells on Embryo Implantation in Mice

Since embryo implantation occurs between embryonic day (E) 4 and E5.5 post mating (the morning after mating is considered E0.5), the DTx was intraperitoneally injected at E3.5 for both the experimental and control groups, as described in (110). Then, gene expression levels for one experimental group were sampled at E4.5, and expression levels for the other experimental group were sampled at E5.5. For each of these two experiments, there was a corresponding control group of mice that did not receive DTx. The remaining three control groups are baseline measurements of virgin mice (have never been pregnant), 1 and 2 days after DTx administration or without the DTx.

We studied a rich gene expression data set detailing the experiments on the effects of uterine dendritic cell depletion on embryo implantation in mice. We used Illumina MouseWG-6 chips, v2.0. It details gene expression levels as measured by 38,715 different probes on 24 different tissue samples (experimental repetitions) divided among 7 different experimental conditions. Each experimental condition had 3-4 biological repetitions. In each repetition there was RNA from 3-5 implantation sites retrieved from

Group	Type	DTx	Days Post Treatment	Pregnant	# Reps
1	Experiment	Yes	1	Yes	4
2	Control	No	1	Yes	4
3	Experiment	Yes	2	Yes	4
4	Control	No	2	Yes	3
5	Control	No	N/A	No	3
6	Control	Yes	1	No	3
7	Control	Yes	2	No	3

Table 3.1: Description of data used for the study.

Groups Compared	p_{mates}	μ_T	σ_T	μ_F	σ_F
(1,3), (2,4)	0.46667	0.88631	0.090264	0.88454	0.093785
1, 2	0.42857	0.89079	0.082772	0.8835	0.09262
3, 4	0.42857	0.87716	0.098379	0.8795	0.098205
5,6	0.4	0.86789	0.10902	0.85022	0.1221
5, 7	0.4	0.88292	0.092649	0.83136	0.13157
5, (6,7)	0.5	0.86924	0.10598	0.84921	0.12219
(1,3), (6,7)	0.47253	0.88258	0.095768	0.85796	0.11551
1,6	0.42857	0.87808	0.097965	0.85205	0.12174
3, 7	0.42857	0.88225	0.09432	0.839	0.1242

Table 3.2: Calculated parameters for CLICK algorithm.

1-2 pregnant female mice or 1-2 uteri from non- pregnant female mice. Specifically, the data is partitioned as described in Table 1. The groups are numbered 1 through 7, and each is defined as either an experimental condition or a control. Four of the groups were given the DTx treatment as previously reported (110), of which two groups were pregnant (injected E3.5) and two were not (virgin mice). In Group 1, pregnant mice were probed 1 day following treatment, and in Group 2 pregnant mice were probed 2 days following treatment (as indicated by the Time Post Treatment column). Group 5 is a pure control case of non-pregnant mice that have not received the treatment. The last column of the table indicates the number of independent tissue samples that were probed for each group.

For the purposes of this study, we seek a set of differentially expressed genes for each of various comparisons of groups and group combinations. For example, we would like to identify the significant genes when comparing Group 1 with its control (Group 2), Group 3 with its control (Group 4), and also the combination of Groups 1 and 3 with the combination of their respective controls.

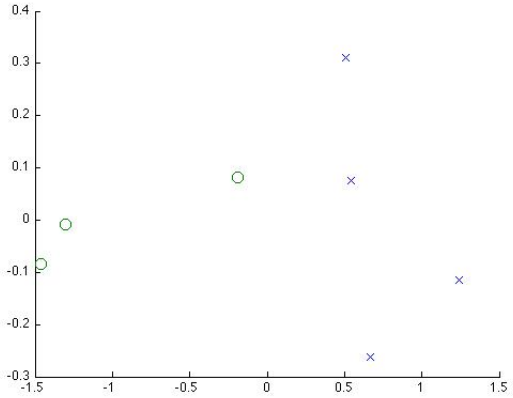


Figure 3.3: PCA plot of top ten genes returned by Graph Cut. Two-dimensional view of the ten most differentially expressed genes as determined by the Graph Cut method when comparing the effects of treatment in pregnant mice after two days (Groups 3 and 4). The control repetitions are represented by the circles and the test repetitions by the x's. As evident from the plot, the Graph Cut algorithm finds genes with a clearly defined separation between test and control.

3.6 Results

We compared our graph cut (GC) method with two others: the CLICK algorithm as modified for gene selection and the general hypothesis testing method as prescribed by the LIMMA software package (133).

We ran nine group comparison studies on the entire data set, as detailed in the first column of Table 3.2. For each group comparison study, we generated the top 20 differentially expressed genes according to the Graph Cut method, CLICK, and LIMMA. Table 3.3 shows the results when comparing Groups 1 and 3 with Groups 2 and 4 (the effect of DTx treatment in pregnant mice). Similarly, Table 3.4 describes the results when comparing Groups 1 and 2 (the effect of DTx treatment in pregnant mice one day after embryo implantation, on E4.5), and Table 3.5 shows the results when comparing Groups 3 and 4 (the effect of DTx treatment in pregnant mice two days after embryo implantation, on E5.5). For each method, we also provide the log fold change values and the p-values associate with each gene; although these values were not necessarily used to sort or generate the results, it is helpful to consider them when evaluating the quality of the outcomes.

We evaluated our results in two manners. First, we performed an analytical analysis that is independent of the biological context of the experiments; this is further described in the following subsection. We also consulted with Dr. Vicki Plaks, an expert biologist, to obtain an understanding of the results from a biological perspective. The results of this analysis are summarized in Section 3.6.2.

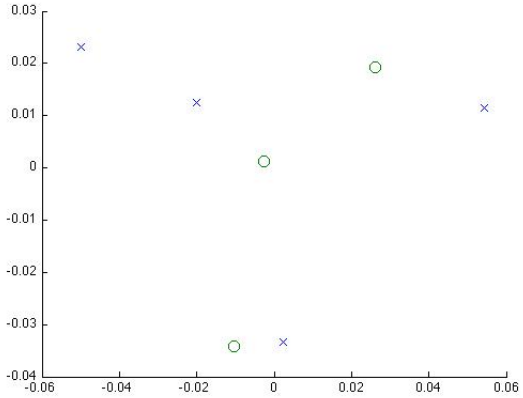


Figure 3.4: PCA plot of top ten genes returned by CLICK. Two-dimensional view of the ten most differentially expressed genes as determined by CLICK when comparing the effects of treatment in pregnant mice after two days (Groups 3 and 4). The control repetitions are represented by the circles and the test repetitions by the x's. As evident from the plot, the genes returned by the CLICK algorithm do not have a clearly defined separation between test and control.

3.6.1 Analytical Measures of Comparison

To understand analytically the performance of GC, CLICK, and LIMMA, we project the results onto a two dimensional plane using principal component analysis (PCA), a linear dimensionality reduction technique that optimally minimizes loss of information. This allows us to visualize the spread or separation of the test and control groups as determined by the genes returned by each method. Intuitively, we assume that the separation between the test and control groups is stronger if they are linearly separable in the PCA projection. For the E5.5 comparison study, we plot the results of each method individually and all combined in Figures 3.3-3.5.

Figure 3.3 illustrates the results from the Graph Cut method. The "x" markers correspond to the test repetitions, and the "o" markers represent the control repetitions. As can be seen, there is a clear separation between the test and control repetitions, which indicates that GC returned significant results. On the other hand, there is no clear, linear separation between the test and control repetitions for the CLICK algorithm, as shown in Figure 3.4. Figure 3.5 shows the results returned by LIMMA, which also demonstrate a good level of separation.

We also evaluated the sensitivity of GC and CLICK to perturbations in the data by artificially introducing varying degrees of Gaussian noise and re-running the analysis. Formally, noise is added for each gene g and experiment i in the following manner,

$$\mathcal{E}'_{g,i} \leftarrow \mathcal{E}_{g,i} + cN(0,1) \tag{3.8}$$

where $N(0,1)$ is a Standard Normal random variable and c is a constant multiplicative

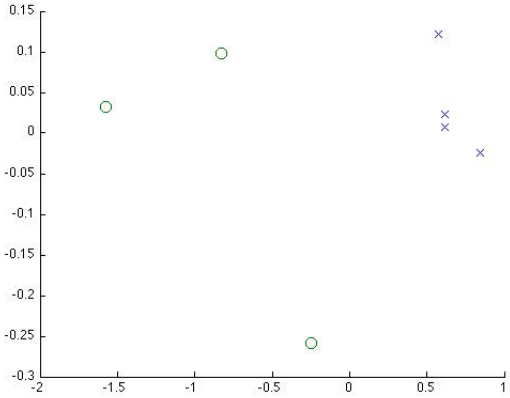


Figure 3.5: PCA plot of top ten genes returned by LIMMA. Two-dimensional view of the ten most differentially expressed genes as determined by CLICK when comparing the effects of treatment in pregnant mice after two days (Groups 3 and 4). The control repetitions are represented by the circles and the test repetitions by the x's.

factor. We performed sensitivity analysis comparing Group 1 with Group 2 with several values for c. Since noise is generated randomly, we repeated the analysis 40 times for each pre-determined noise factor. In each iteration we randomly generate a noisy data set given the fixed noise factor; we then run both GC and CLICK and compute the top 20 genes identified by each method.

For each method, we compare the resulting gene lists from the noisy data sets with our original results by using the Jaccard Index, which measures the size of the intersection of two sets divided by the size of their union. Formally, if we wish to compare gene set A with gene set B, then the Jaccard Index is defined as:

$$\text{Jaccard Index}\,(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{3.9}$$

Hence, larger values of the index correspond to greater degrees of agreement between the two sets. If $A = B$, then the index is at its maximum value of 1; conversely, if $A \cap B = \emptyset$, then the index is at its minimum of 0.

Figure 3.6 illustrates the Jaccard Index for both GC and CLICK with varying noise levels. As evident from the plot, the two methods exhibit similar behavior at lower degrees of noise, but then CLICK quickly drops to a negligible Jaccard Index while GC drops at a significantly more gradual pace. This suggests that the GC method is less sensitive to noise in the data and hence more reliable than CLICK.

3.6.2 Summary of Biological Findings

Overall, analysis of this expression array data set using Graph Cut allowed us to discover factors that were specifically down and upregulated with uDC depletion during embryo

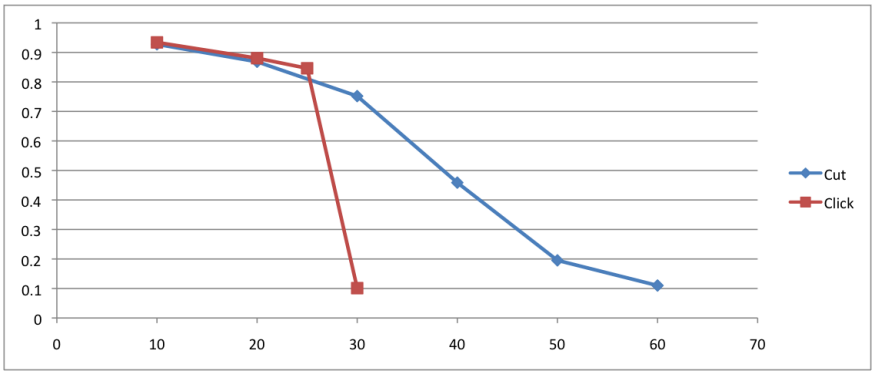


Figure 3.6: Jaccard Index computed results on the original data with varying degrees of noisy data for both GC and CLICK. The horizontal axis corresponds to the noise level and the vertical axis is the corresponding Jaccard Index.

implantation more reliably than with the other two models examined, CLICK and LIMMA. In each embryonic day tested post uDC- depletion, GC was able to show similar results to one on of the other models (but not the other) and overall, exhibited more continuation between the two days of the embryo implantation process (some of the genes in E4.5 are still expressed in E5.5 but other are absent- show examples). Specifically, examining uDC-depleted implantation sites on E4.5 (one day after uDC depletion), we were able to detect more upregulated factors which belong to twopro-inflammatory chemokine families with Graph Cut versus CLICK while this group was completely absent with LIMMA. As the process progresses, looking at uDC- depleted implantation sites on E5.5 (two days after uDC depletion), these pro- inflammatory chemokines were no longer differentially expressed. However, while CLICK exhibited a completely different set of genes from the E4.5 data and from whatever exhibited by GC and LIMMA on E5.5, Granzyme D that was downregulated on the E4.5 data still persisted on E5.5 with GC (as with CLICK). Other geneS specifically involved in uterine receptivity and embryo implantation were all downregulated, as exhibited by the results shown with GC.

3.7 Discussion and Future Work

To summarize, both the mathematical and the biological results from this study indicate that graphical methods, and specifically the Graph Cut method, can yield efficient and promising results for the gene selection problem. Furthermore, the Graph Cut algorithm was shown to be significantly less sensitive to noise in the data as compared with CLICK.

Any one algorithm for identifying the most differentially expressed genes is unlikely to be the best choice in every single scenario. As evidenced by our analysis, although Graph Cut seemed to be superior in this study, each of the three algorithms we evalu-

ated returned important and unique results. Hence, a natural and promising extension to this work is to form a more intelligent algorithm based on an ensemble of various methods. This could employ models from machine learning, ensemble learning theory, and group decision theory to help eliminate false positives and identify significant genes based on a consensus of various techniques. On the biological perspective, future work should be focused on sorting uDC and examining their expression profile versus other tissue DC or other DC in lymphoid organs. Also, upon uDC depletion, it will be interesting to examine other cells residing within the decidua (as macrophages, neutrophils, NK cells and eosinophils) and compare their expression profile (and numbers) to those residing within normal implantation sites.

3.A Running CLICK

The first step in running the CLICK algorithm is to calculate the parameters that characterize the relationships between mates and non-mates. Since the ground truth is known in advance, we are able to calculate the true values as opposed to estimating with a learning algorithm such as Expectation-Maximization (EM). Therefore, for each group comparison study we calculate a separate set of values for $p_{mates}, \mu_T, \sigma_T, \mu_F, \sigma_F$; the exact values can be found in Table 3.2 below. We used the same similarity metric defined in Equation 3.1, with α and β set to 1, and γ set to 2. The edge weight model is defined in Equation 3.5, and is taken directly from the CLICK algorithm.

Since we already know the ground truth bi-partitioning solution (that is, we know which repetitions belong to the control groups and which belong to the test groups), we do not need to run the entire CLICK algorithm. Rather, for each gene we compute the capacity of the cut under the true bi-partition, and then we sort the genes accordingly.

3.B Supplemental Data Tables

Rank	Cut			Click			LIMMA		
	Gene Name	Log FC	P-Value	Gene Name	Log FC	P-Value	Gene Name	Log FC	P-Value
1	Gzmd	-2.2551	0.0000	Cxcl1	2.78513	0.0195	Gzmd	-2.2551	2.8684E-06
2	Cxcl1	2.7851	0.0195	Gzmd	-2.25507	0.0000	Gzme	-1.2087	1.9684E-05
3	Klk1b21	1.63825	0.0233	Klk1b21	1.63825	0.0233	Gzmd	-1.4228	6.1541E-05
4	Ccl7	1.34826	0.0392	Ccl7	1.34826	0.0392	Gzme	-0.9575	3.9976E-04
5	Ear2	-1.24012	0.0001	Ceacam10	-1.46554	0.1112	Ear1	-1.1883	6.5313E-05
6	Smad6	-0.92852	0.0017	Prl8a2	-1.38913	0.1928	Ear2	-1.2401	8.8029E-05
7	Ear2	-1.18827	0.0001	Sct	-0.63460	0.3035	Igfbp5	0.8308	5.5199e-04
8	Dio3	-0.91664	0.0119	Dlk2	-0.76542	0.1864	Pyy	-0.5883	0.0012
9	Prkg2	-0.94926	0.0367	Dio3	-0.91664	0.0119	Serpina1b	-0.7519	2.6805e-04
10	Cxcl2	2.27732	0.0102	Drd4	-0.80194	0.2981	Entpd6	-0.4933	0.0020
11	Notum	-0.91818	0.0207	Cxcl2	2.27732	0.0102	Ear10	-0.9451	1.8288E-04
12	Ccl4	1.36042	0.0173	Smad6	-0.92852	0.0017	Neu2	-0.8587	5.6349E-05
13	Fst	-0.92155	0.0071	Ccl4	1.36042	0.0173	S100a9	4.0451	0.0893
14	Smad6	-0.92425	0.0053	Ear2	-1.24012	0.0001	Igfbp5	0.8818	0.0010
15	Cxcl10	1.68026	0.0255	Cxcl10	1.68026	0.0255	Saa3	2.6203	0.0612
16	LOC100044206	1.29849	0.0094	Dmkn	-1.16287	0.1940	Plce1	-0.1417	8.1760E-04
17	Gzmd	-1.42275	0.0001	Prkg2	-0.94926	0.0367	Galnt9	0.0258	7.1630E-04
18	Tnfaip3	1.15307	0.0203	Kazald1	-1.02367	0.1048	Ear2	-0.9391	4.3440E-04
19	Ebpl	-0.89934	0.0031	Notum	-0.91818	0.0207	S100a3	-1.0662	0.0167
20	Ear10	-0.94505	0.0002	Prl3c1	-1.35361	0.1490	Ear4	-0.8123	4.4637E-04

Table 3.3: Results of analysis comparing treatment versus no treatment in pregnant mice (Groups 1 and 3 compared with Groups 2 and 4.

Rank	Cut			Click			LIMMA		
	Gene Name	Log FC	P-Value	Gene Name	Log FC	P-Value	Gene Name	Log FC	P-Value
1	Cxcl1	1.37534	0.0001	Cxcl1	9.38336	0.0001	Baalc	-0.938	0.0025
2	Saa3	1.23015	0.00055	Saa3	4.08657	0.00055	Gzmd	-2.0191	0.0015
3	Gzmd	0.79776	0.00247	Gzmd	0.24127	0.00247	Gzme	-0.9734	0.0036
4	Prkg2	0.83727	0.04166	Prkg2	0.3782	0.04166	Ear2	-1.3592	0.0029
5	Cxcl2	1.38778	0.00005	Smad6	0.45546	0.03462	Neu2	-1.1474	0.0013
6	Ear2	0.84711	0.00212	Cxcl2	6.60839	0.00005	Ear2	-1.3501	0.0049
7	Rnf170	0.8795	0.00145	Ccl2	3.29606	0.00558	Gzmd	-1.1485	0.0037
8	Ccl7	1.13639	0.00434	Ccl7	2.60891	0.00434	LOC435565	0.7938	0.0080
9	Klk1b21	1.19782	0.00558	Ear2	0.36665	0.00212	1700027L20Rik	-1.2231	0.0213
10	Neu2	0.86718	0.00288	Dio3	0.4073	0.06624	Ear10	-1.1011	0.0090
11	Smad6	0.87161	0.03462	Smad6	0.45318	0.03419	4-Sep	0.2107	0.0145
12	Ear2	0.85176	0.00158	Cthrc1	0.38452	0.00145	Cd209a	-0.6601	0.0067
13	Chodl	1.15703	0.00149	Wnt4	0.50729	0.06691	Lrrc3	-0.387	0.0184
14	Smad6	0.86685	0.03419	Ear2	0.36435	0.00158	4-Sep	-0.8386	0.0213
15	Notum	0.84717	0.01036	Neu2	0.42625	0.00288	Syt15	0.5406	0.0041
16	Cxcl10	1.22851	0.00255	Chodl	2.90659	0.00149	2610109H07Rik	-0.9306	6.6017E-04
17	Adamdec1	1.17121	0.00541	Abp1	0.32073	0.04279	Entpd6	-0.568	0.0114
18	Cxcl2	1.22263	0.00041	Wnt4	0.53918	0.07968	Cthrc1	-1.3526	0.0025
19	Abp1	0.81571	0.04279	Cxcl10	3.90093	0.00255	LOC100044206	1.3934	0.0056
20	LOC100044206	1.14272	0.00142	780938	2.66674	0.2718	Ear2	-1.0646	0.0192

Table 3.4: Results of analysis comparing treatment versus no treatment in pregnant mice after 1 day (Group 1 compared with Group 2).

Rank	Cut			Click			LIMMA		
	Gene Name	Log FC	P-Value	Gene Name	Log FC	P-Value	Gene Name	Log FC	P-Value
1	Ceacam10	-2.00404	0.0069	EG627798	-0.00851	0.4189	Gzme	-1.482	0.0059
2	Gzmd	-2.53293	0.0046	Rps5	-0.00027	0.9856	Rnf182	-0.9021	1.1059E-04
3	Prl8a2	-1.63487	0.0240	Slc22a9	0.01285	0.3951	Rnf182	-1.0518	9.7485E-04
4	Tnnc1	-1.57929	0.0087	EG668406	0.01817	0.3043	Gzmd	-2.5329	0.0046
5	Angpt4	-1.47513	0.0059	ENSMUSG0..060019	0.01882	0.3487	Ceacam10	-2.004	0.0069
6	BC061237	-1.39784	0.0059	OTTMUSG0..014597	-0.01055	0.6078	Gzmd	-1.7412	0.0120
7	Il13ra2	-1.39154	0.0327	ENSMUSG0..057262	-0.01113	0.6286	Hamp	-1.8312	0.0187
8	Stmn2	-1.51103	0.0093	Rps7	-0.01424	0.5198	Angpt4	-1.4751	0.0059
9	Prl3c1	-1.64883	0.0132	Hecw1	0.01253	0.5765	Serpinb9f	-1.8738	0.0380
10	Ptger3	-1.44909	0.0227	Arl13b	-0.01049	0.6621	Tnnc1	-1.5793	0.0087
11	Arfgap2	-1.11669	0.0426	Rpl41	0.02321	0.3704	Gzme	-1.1994	0.0163
12	Gzmd	-1.74116	0.0120	NA	0.02334	0.2961	Angpt2	-0.9735	0.0181
13	Kazald1	-1.24354	0.0218	Kcng2	0.01072	0.6795	Psca	-1.0608	0.0216
14	Hsd17b1	-1.26931	0.0104	Rps12	0.0201	0.3718	Dtna	-0.8481	0.0049
15	Ptn	-1.14476	0.0280	Tcrg-V3	-0.01566	0.5481	Serpinb9e	-0.7596	0.0121
16	Krtdap	-1.43424	0.0173	V1ra3	0.00306	0.9092	Stmn2	-1.511	0.0093
17	Hopx	-1.1386	0.0111	Sox6	-0.00076	0.9785	Abcb9	-0.5649	0.0018
18	Hamp	-1.83119	0.0187	Rps9	-0.00152	0.9543	Serpinb9g	-0.8459	0.0228
19	Gzme	-1.48204	0.0059	EG625054	-0.01444	0.5935	BC061237	-1.3978	0.0059
20	Ebpl	-1.11807	0.0156	4930408O21Rik	-0.00702	0.7941	Nrcam	-0.9516	0.0169

Table 3.5: Results of analysis comparing treatment versus no treatment in pregnant mice after 2 days (Group 3 compared with Group 4).

4

Constant-time, Adaptive Collaborative Filtering Systems

4.1 Introduction

In this chapter we describe a new geometric approach to collaborative filtering and two applications that implement variations of the method. The problem of personalized recommendation relates to search and filtering in that we must search through a large set of data and analyze the patterns of that data to model and understand user preference. With this understanding, we seek to make personalized recommendations from a set of items that meet the user's search requirements or goal. The models we present here, which are based on the Eigentaste 2.0 algorithm (60), use the geometric properties of user preference data in order to make personalized recommendations in constant online time. We study extensions of Eigentaste 2.0 on the Jester joke recommender system (`http://eigentaste.berkeley.edu`) and Donation Dashboard (`http://dd.berkeley.edu`), a system that recommends a weighted portfolio of donations to nonprofit organizations.

We begin the chapter by explaining the relationship between collaborative filtering (CF) and more general recommender systems, which we follow with a description of the most commonly used techniques in CF. We also discuss how different applications can require slightly different user tasks and objectives and how that ultimately motivates the need for different types of algorithms. Since the algorithms and models presented in this chapter are based on Eigentaste 2.0, for completeness we provide a description of Eigentaste 2.0 in Section 4.1.3. In Section 4.2 we describe the Jester joke recommender system and Eigentaste 5.0, a collaborative filtering algorithm designed to adapt in real-time to changes in user preferences. Section 4.4 discusses our design and implementation of Donation Dashboard and our analysis of its performance.

4.1.1 Collaborative filtering as a special class of recommender systems

Recommender systems are designed to make personalized item recommendations to users based on some prior knowledge gathered about their preferences. These systems are becoming more and more ubiquitous in our world today. We use them to find books to read, gifts to buy, tv shows to watch, restaurants to dine at, and the news and commentary we see. Pandora is a highly popular online service that makes and streams personalized music playlists. Users are asked to provide the name of at least one song or artist they like, and then Pandora creates a radio station that plays music with similar qualities. Users have the option of giving a thumbs up or down to the songs they hear to better tailor their experience to their personal tastes. While this is a very popular service, it's important to note that it is not collaborative; songs are assessed solely on their musical properties and not on how people rate them.

Collaborative filtering is a special case of recommender systems that looks for patterns in user behavior and data to infer how users will respond to different items. The context in which collaborative filtering systems are defined traditionally involve a class of items, I, and a set of users U. Item classes may include consumer goods such as books, movies, music, or even academic papers, and they can be more eclectic or intricately defined such as running routes or medical treatments.

Given a set of users and items, the system must also have a way to collect data on how users respond to items in I, or rather, a way to characterize each user's taste profile. This can be done explicitly by collecting some form of user ratings or evaluations of the items they consume, or it can be done implicitly by observing user behavior and making inferences about their preferences. While systems that collect user data implicitly require significantly less effort on behalf of the user, the tradeoff is in the added noise to the data and the amount of control over the data available. For example, Google News (32) collects implicit user data by observing the news articles that users click on and assuming that these are positive votes for those articles. However, this is an incredibly strong assumption, since there is no guarantee that a user will actually like the article, and there is no way for the user to provide negative feedback to the system.

Most collaborative filtering algorithms fall into one of three categories: memory-based, model-based, or a hybrid systems. Memory-based CF models are the most traditional type, using rating and similarity data to make recommendations; these are often referred to as neighborhood-based algorithms. The general structure of such algorithms is to find the k nearest neighbors of the active user and then make predictions based on the aggregation of the item ratings of those k users.

Determining the appropriate measure of similarity between two elements (either items or users) is a crucial element of memory-based collaborative filtering, because the final set of recommendations can only be as good as the similarity metric used. Euclidean distance is one of the more intuitive measures (though it actually measures *dissimilarity*), but it can only be used on fully dense matrices. Since many applications contain thousands of items, this is generally not a practical model. Correlation is a family of metrics that is used most often to measure the similarity between two elements.

If we treat each element as a random variable, then correlation tells us how much the variance in one variable is explained by the variance of the other. Pearson correlation is one of the most well-known methods that measures the linear relationship between the variables. (70) Spearman and Kendall's tau correlation metrics, on the other hand, measure ordinal agreement between two rankings rather than cardinal. Xiaoyuan et. al (154) provide an excellent in-depth discussion of other similarity metrics and strategies used, including ones based on conditional probability theory.

As outlined in (154), there are several advantages to memory-based CF: it is easy to implement, item content is not considered, and the resulting recommendations can be explained relatively easily. On the other hand, these models struggle to perform well when data are sparse, and they suffer from the *cold start* problem, which is to make meaningful recommendations to new users.

Model-based collaborative filtering takes a slightly more sophisticated approach, using statistical techniques to model user taste with latent variables. Classes of algorithms falling under this category include Bayesian belief nets, clustering, Markov decision processes (MDP), latent semantic indexing (LSI), factor analysis (FA), and dimensionality reduction techniques including singular-value decomposition (SVD) and principal component analysis (PCA). According to Xiaoyuan and Khoshgoftaar (154), model-based CF algorithms are able to handle data sparsity and scalability problems better than their memory-based counterparts. They also tend to make more accurate predictions, although there is often a tradeoff here with scalability. One of the disadvantages of these methods is that they can be more difficult to explain to lay users, and the models can be expensive to build.

Hybrid CF algorithms boost either memory or model-based collaborative filtering by incorporating content data. This serves to address the cold start problem experienced by new users and makes systems more robust to sparsity. As a result, these algorithms see an improvement in predictive performance. The disadvantages here are that these models are more complex to build and explain, and they require a featurization of the items that is not always possible.

Some of the major problems in collaborative filtering research include designing mechanisms that protect privacy and security and are robust to manipulation. Another major problem is to find a method for comparing and evaluating different algorithms. Currently, many researchers resort to evaluating recommender systems based on their predictive performance, but this is only half the battle; the second half is to do something *useful* with the predictions. For example, suppose user u gives the highest possible rating to the movie The Godfather. Then a system that makes good predictions might determine that the user is highly likely to like The Godfather II. While this may be true, it defeats the purpose of the recommender system, since the user is very likely already aware of this movie. This leads to another major problem in CF research: serendipity, where the goal is to help the user find items in I that she might not have otherwise explored or known about and that she likes.

4.1.2 Applications and differences in their respective user tasks

In 1992, Goldberg and his team at Xerox PARC (57) came out with one of the earliest collaborative filtering systems, Tapestry. The system was designed to help users better manage the overwhelming amount of email and newsgroup messages they were receiving by harnessing the power of human reasoning in the automated filtering process. However, it required that users explicitly identify those that they trust and from whom they would like to receive recommendations. The 1994 GroupLens (116) project for filtering netnews relaxed this requirement by automatically identifying highly correlated users and making recommendations based on their preferences. Automating this step in the filtering process eliminated the need for rating transparency and hence enabled users to provide their opinions under the privacy of a pseudonym.

As the study of collaborative filtering took off, many different applications began to spring up. The GroupLens team at the University of Minnesota developed MovieLens (`http://movielens.org`) in 1997, a collaborative system for making movie recommendations. Work on this system, which is still active today, became the basis of the original book recommendation engine at Amazon.com (11). Other applications that have surfaced include various music recommenders (iTunes Genius and Last.fm), news (Google News), clothing (Boutiques.com) and even online dating (Match and OK Cupid).

While at first glance many of these applications seem similar and hence generalizable to a (user class, item class, recommendation set) system, they exhibit subtle differences in user tasks that require more careful attention. Herlocker et al. (70) outline a series of *user tasks* that characterize a variety of different collaborative filtering applications, both from a system viewpoint and the user's viewpoint. For example, a user may have one of the following goals:

- **High Precision. (Find some good items.)** This task is especially relevant in entertainment-related applications. Consider, for example, when the user wants to find a good movie to watch or book to read. The user is not interested in finding *every* good movie or book out there, but instead she is looking for a satisfactory suggestion. Precision can be measured as follows.

$$\text{precision} = \frac{|\{\text{relevant items}\} \bigcap \{\text{retrieved items}\}|}{|\{\text{retrieved items}\}|} \tag{4.1}$$

- **High Recall. (Find all good items.)** This task is less common in recommender systems today and is appropriate when it is truly important that the user find all relevant, high-quality items. An example would be a researcher searching for all relevant publications to a particular subject area (when performing a literature review). In this case, it is crucial that the system returns all relevant material. We can measure recall with the following formula.

$$\text{recall} = \frac{|\{\text{relevant items}\} \bigcap \{\text{retrieved items}\}|}{|\{\text{relevant items}\}|} \tag{4.2}$$

Many recommender systems require users to contribute their opinions to the site, so that the system can learn and make better recommendations. Users may have a variety of motivations for actively participating, which include the following that are highlighted by Herlocker et al. (70)

- Improve profile (and recommendation accuracy) by contributing ratings
- Express oneself
- Help others
- Influence others (possibly maliciously)

From a system perspective, the recommendation task may be to:

- Recommend a single item that the user will like / dislike / find interesting
- Recommend a set or portfolio of items (where order is not relevant)
- Recommend a sequence of items (e.g. a music playlist, or a series of jokes, where order is relevant)

Another key factor in the design of recommender systems is the time required to consume the items being recommended. In some cases, such as with music, news articles, television shows, and jokes, the items can be consumed relatively quickly, and so feedback can be obtained right away. In other cases, such as with books or hotels, the item is consumed at a later point in time and it may be difficult to learn about the user as quickly. In this case, the *cold start* problem is especially relevant.

4.1.3 Eigentaste 2.0

We now describe Eigentaste 2.0 (60), a patented constant-time collaborative filtering algorithm that was originally used with the Jester joke recommendation system. Eigentaste handles the cold-start problem by collecting real-valued user ratings of a common set of items, referred to as the *gauge set*. This set contains the items with the highest variance ratings, which gives the most information about the preferences of a user in the fewest number of ratings.

To save computation time, the algorithm is divided into an offline phase and an online phase. In the offline phase, principal component analysis (PCA) is applied to the covariance matrix of the gauge set ratings matrix, and each user is projected onto the resulting two-dimensional eigenplane. Due to a high concentration of users around the origin, a median-based algorithm referred to as recursive rectangular clustering is used on the lower-dimensional space to cluster the users into groups with similar tastes.

In the online phase, the appropriate cluster for a new user requires only a dot product with the stored principal eigenvectors, which can be done in constant time. Items are then recommended to the user in descending order of the average item ratings for users in the same cluster.

4.2 The Jester Joke Recommender System

The purpose of a recommender system is to eliminate the need for browsing the entire item space by presenting the user with items of interest early on. As such, our objective in developing an effective recommender system is to sustain higher ratings over earlier recommendations.

Although Eigentaste 2.0 provides a fast cold-start mechanism for new users, each user is permanently assigned to a user cluster and thus a fixed item presentation order that is determined by predicted ratings for that cluster. A disadvantage is that the system does not respond to the user's subsequent ratings. Another disadvantage is the potential for presenting similar items sequentially, referred to as the *portfolio effect* (87; 161). For example, in Jester, our joke recommender system, a user who rates a Chuck Norris joke highly might then be recommended several more Chuck Norris jokes. In humor, as in many other contexts like movies or books, the marginal utility of very similar items decreases rapidly.

To address these problems we propose Eigentaste 5.0, an adaptive algorithm that, in constant online time, dynamically reorders its recommendations for a user based on the user's most recent ratings. We observe that the problem of finding item clusters based on users' ratings is the dual to the problem of finding user clusters based on item ratings. We develop a hybrid approach that allows us to take advantage of the computational efficiencies provided by solutions to both problems: PCA facilitates fast placement of new users in appropriate user clusters, and maintaining clusters or *portfolios* of similar items allows us to monitor portfolio effects.

Item clustering can also be used to address the cold-start problem for integrating new items into the system. A new item can be quickly matched with similar items, and the ratings of those items can be used to predict ratings for the new item.

Jester (`http://eigentaste.berkeley.edu`) uses jokes as an item class to demonstrate and evaluate Eigentaste 2.0. Jokes are an example of an item class that can be rated quickly and without prior knowledge. Between March 1999 and May 2003, 73,421 users registered and rated a set of 100 jokes using the Jester system, producing a total of 4.1 million continuous ratings in the range [-10.00, +10.00].

In November 2006 we released Jester 4.0, which introduced a number of improvements, including 50 new jokes, a redesigned user interface, and a visible slider that hovers above the continuous rating meter. To deal with the cold-start problem for new jokes, Jester 4.0 collects ratings for the 5 most sparsely rated jokes at the time of a new user's registration. It continues to interleave sparsely rated jokes during the recommendation phase. Jester 4.0 has collected approximately 110,000 joke ratings of 150 jokes from 3,000 users (as of August 10, 2007). Both data sets, including anonymous ratings, are available upon request.

The features of Eigentaste 5.0 (dynamic recommendations and bootstrapping new items) will be implemented in Jester 5.0.

4.2.1 Related Work

Quan et al. (113) propose a collaborative filtering algorithm that adds item clusters to a *user-based* approach: when predicting an item's rating for a user, it looks at similar users with respect to ratings within that item's cluster. George and Merugu (56) use Bregman co-clustering of users and items in order to produce a scalable algorithm. Vozalis and Margaritis (149) compare other algorithms that combine *user-based* and *item-based* collaborative filtering.

Eigentaste 2.0 deals with rating sparseness by ensuring that all users rate the common set of items, but there are many alternative solutions to this problem. Wilson et al. (152) approach sparseness by using data mining techniques to reveal implicit knowledge about item similarities. Xue et al. (158), on the other hand, fill in missing values by using user clusters as smoothing mechanisms. Wang et al. (150) fuse the ratings of a specific item by many users, the ratings of many items by a certain user, and data from similar users (to that user) rating similar items (to that item) in order to predict the rating of that item by the given user. This approach both deals with sparsity and combines *user-based* and *item-based* collaborative filtering. Herlocker et al. (70) evaluate several different approaches to dealing with sparseness.

Eigentaste 2.0 scales well because in constant online time, it matches new users with user clusters that are generated offline. Linden et al. (93) at Amazon.com use an *item-based* collaborative filtering algorithm that scales independent of the number of users and the number of items. Rashid et al. (115) propose an algorithm, ClustKnn, that combines clustering (*model-based*) with a nearest neighbor approach (*memory-based*) to provide scalability as well as accuracy. Earlier, Pennock et al. (108) evaluated the method of *personality diagnosis*, another technique that combines *model-based* and *memory-based* approaches by aiming to determine a user's personality type. Deshpande and Karypis (35) evaluate *item-based* collaborative filtering algorithms and show that they are up to two orders of magnitude faster than *user-based* algorithms.

As described above, Eigentaste 2.0 is well-suited to the cold-start situation for new users. Park et al. (107) use *filterbots*, bots that algorithmically rate items based on item or user attributes, to handle cold-start situations. Schein et al. (126) discuss methods for dealing with the cold-start problem for new items by using existing item attributes, which is symmetric to the cold-start problem for new users when user attributes are available. Rashid et al. (114) survey six techniques for dealing with this situation. Cosley et al. (27) investigate the rating scales used with recommender interfaces.

Other improvements to Eigentaste 2.0 include Kim and Yum's (86) iterative PCA approach that eliminates the need for a common set of items. Lemire's (92) scale and translation invariant version of the Eigentaste 2.0 algorithm improves NMAE performance by 17%.

Ziegler et al. (161) propose a new metric for quantifying the diversity of items in a recommendation list, which is used to address the portfolio effect. Konstan et al. (87) extend this work to apply recommender systems to information-seeking tasks. While these works improve user satisfaction by increasing topic diversity in recommendation lists, we do so by dynamically reordering recommendations in response to new ratings.

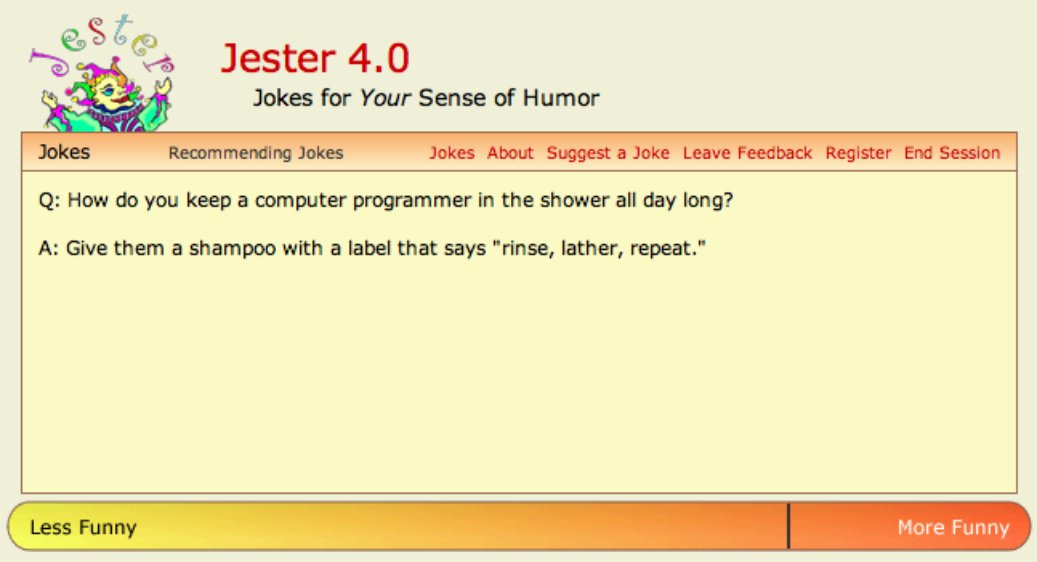


Figure 4.1: Screenshot of the Jester joke recommender system.

4.2.2 Description of the Jester system

Jokes are presented to users one at a time. Below each joke is a horizontal, real-valued sliding scale that the user can manipulate to indicate how funny she finds the joke, ranging from "Less Funny" to "More Funny." To enter a rating with this interface feature, the user clicks on the appropriate spot along the horizontal axis of the scale. (See Figure 4.1.) The coordinate at which the user clicks is then converted into a corresponding real number and stored in the Jester database. The system then automatically advances to display the next joke. It is important to note that the only way to navigate from one joke to the next is by providing a rating of the joke using this scale, which may hence result in additional bias from users who are not interested in providing an accurate rating or who simply want to skip to the next joke without taking the time to read and evaluate the present joke.

The first eight jokes presented to every new user comprise a pre-determined, fixed "gauge set." These are the jokes that historically have the greatest variance in ratings, and is thus designed to help the system learn as much as possible about the sense of humor of the user. By asking every new user to evaluate the same eight jokes, the submatrix of ratings corresponding to the gauge set is completely dense and thus allows us to project each user's ratings into two-dimensional space using PCA.

The original version of Jester followed the method prescribed by Eigentaste 2.0, placing the new user in the most appropriate cluster given her ratings of the gauge set jokes. The user was then presented with the top jokes (one at a time) in descending order of average rating by users in her cluster. Just like when rating the gauge set, to advance to the next joke the user must still provide a rating using the sliding scale

provided.

4.2.3 Analysis of user tasks

In the case of Jester, the primary *user task* is to browse jokes for entertainment purposes. Unlike many mainstream recommender applications, users are not looking to make a purchase or informed decision, but rather they are seeking a humorous experience personalized to their own individual tastes. Consequently, the challenge faced by the system is to recommend an appropriate sequence of jokes that keeps the user sufficiently entertained for as long as possible.

For a user to maximize her enjoyment of the system, she must be willing to participate in the gauge set rating process, which requires thoughtful ratings of all jokes in the gauge set. She must be made to understand that in doing so, she is improving her profile and hence the quality of her experience. By forcing the user to rate every presented joke (since it is the only way to navigate to the next joke), users are also helping the system learn about the nuances of humor. This potentially serves to improve future recommendations for other users.

Since users are not contributing content to the system, and since they therefore have no personal stake in the content, we are not overly concerned with the possibility of users desiring to manipulate the system in their favor. However, we must be aware of users who give inaccurate ratings and thus inadvertently bias the system.

4.2.4 Eigentaste 5.0: Adapting in real-time to changes in user taste

In this section we describe Eigentaste 5.0, an extension of Eigentaste 2.0 for use with Jester. Although there are no versions 3.0 or 4.0, we adopted this version numbering to maintain consistency with our recommender system; that is, the next version of Jester (5.0) will incorporate Eigentaste 5.0.

4.2.4.1 Notation

U	the set of all users
I	the set of all items
$r_{u,i}$	user u's rating of item i
$\overline{r_i}$	the mean rating of item i
$\overline{r_u}$	user u's mean item rating
$\vec{a}_u$	moving average of u's ratings per item cluster

4.2.4.2 Dynamic Recommendations

We seek to enhance the Eigentaste 2.0 algorithm so that it considers changes in a user's preferences when selecting the next item to recommend. In order to maintain the constant online running time of the algorithm, we exploit the dual nature between users and items in a recommendation system. By partitioning the item set into groups of similar items, we can make recommendations based on user preferences for certain

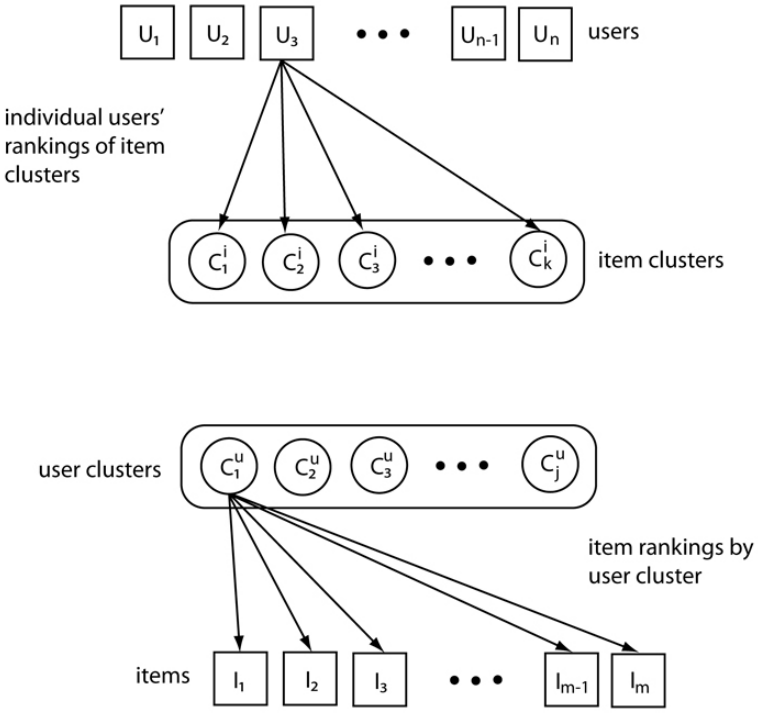


Figure 4.2: Diagram of user and item clusterings in Eigentaste 5.0.

classes of items, instead of moving users to different user clusters as their preferences change. While clustering users into groups with similar taste and aggregating their ratings is helpful in predicting how a new user would rate the items, we can tailor recommendations in real-time as we learn more about the user's preferences.

We use the k-means algorithm to cluster the item space offline and across all user ratings. The Pearson correlation function is used as a distance metric between items, where the correlation between item i and item j is defined as follows:

$$P(i,j) = \frac{\sum_{u \in U}(r_{u,i} - \overline{r_i})(r_{u,j} - \overline{r_j})}{\sqrt{\sum_{u \in U}(r_{u,i} - \overline{r_i})^2}\sqrt{\sum_{u \in U}(r_{u,j} - \overline{r_j})^2}}$$

Pearson correlation provides a real-valued measure on the scale of $[-1, +1]$, where greater values correspond to higher correlation. To use this as a distance metric, we compute 1 minus the Pearson correlation, where larger values reflect greater distances. Note that it is standard procedure to only consider users who have rated both items when computing Pearson correlation, but this is problematic for k-means clustering

Item Cluster	User u's last 5 ratings	Average
1	4.2, 5.3, 3.8, 2.1, 2.7	3.62
2	7.2, 6.5, 5.9, 0.8, -1.2	**3.84**
3	1.9, -2.4, 3.8, 2.1, 0.5	1.18

Table 4.1: Illustrative example of data storage and computations required to make the next recommendation for user u with Eigentaste 5.0.

when dealing with sparse data. Substituting average ratings for missing data dilutes the actual ratings obtained for the item, and the clustering algorithm is more likely to place the sparse items in the same item cluster. For this reason, in our experiments we only considered users that had rated all of the items in the set in order to avoid the sparsity issue.

For each user $u \in U$ we maintain a vector $\vec{a}_u$ of a moving average of u's ratings of the items in each item cluster; that is, $\vec{a}_u[c]$ corresponds to user u's average rating of the last n items in item cluster c, where n is some constant. We initialize a new user's moving average slots with his ratings of items from the common set. For new users who do not have n ratings for each item cluster, we seed the remaining slots of their moving average with the average ratings for the corresponding item cluster across users from the same user cluster. (See Figure 4.2 for an illustration.)

In Eigentaste 2.0, a user is always recommended items in decreasing order of their predicted ratings, where predictions are determined by the average rating for that item by users in the same user cluster. In essence, a user's ratings of additional items have no influence on which item is recommended next.

We give a constant online time solution as follows: user u is recommended the top-predicted item (not yet rated by u) from the item cluster corresponding to the highest value in $\vec{a}_u$. As u tires of the items from that cluster, the moving average of his ratings for the cluster will begin to decrease and eventually fall below that of another item cluster.

For clarity, we provide a numerical example that walks through the process of recommending the next item to some user u. Suppose the values in Table 4.1 represent $\vec{a}_u$, the last five ratings provided by user u for items in each item cluster.

At present, user u's moving average of items in cluster 2 is the highest, and so Eigentaste 5.0 presents u with the item from cluster 2 that has the highest predicted rating and has not yet been evaluated by u. Suppose u rates this item with -2.3. Her new moving average of item ratings for cluster 2 becomes 1.94, which is lower than the moving average for item cluster 1. Subsequently, in the next iteration, user u will be recommended the item from cluster 1 that has the highest predicted rating, and the process is repeated.

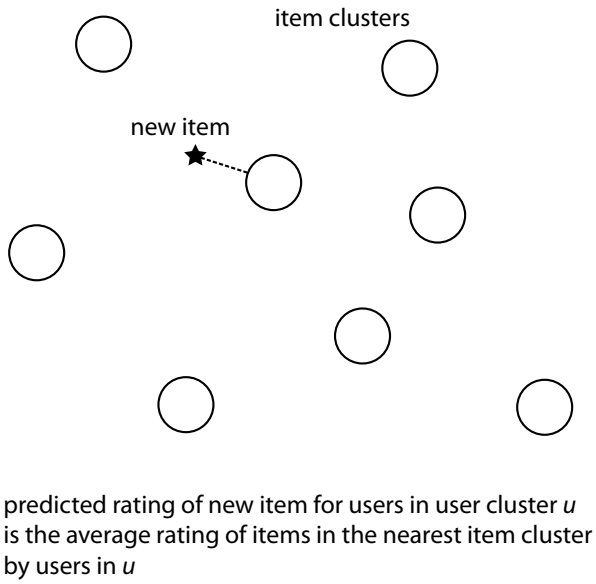


Figure 4.3: Illustration detailing how Eigentaste 5.0 addresses the cold start problem.

4.2.4.3 Cold Starting New Items

Difficulty in introducing new items to the system stems from the fact that they have so few ratings overall, and even fewer ratings within individual user clusters. Hence, the predictions generated by Eigentaste 2.0 are subject to more variability due to the small sample sizes.

Eigentaste 5.0 uses sparse ratings for a new item i to find the closest item cluster based on the Pearson distance metric described in section 4.2.4.2. User u's predicted rating of i is determined by the average rating of all items within i's nearest item cluster across users in u's user cluster (Figure 4.3). We use confidence intervals to determine the appropriate time to switch from this estimate to the estimate based only on actual ratings of item i.

4.3 Experimental Results

It is impossible to truly compare how a user would react to different item recommendation orders, as the first test would greatly bias the second. In the near future, we will release Eigentaste 5.0 for data collection and evaluate the algorithms by randomly assigning new users to either Eigentaste 2.0 or 5.0; in the meantime, we compare the algorithm with its predecessor by backtesting on data collected from Jester.

In this section we discuss recent improvements to Jester and report our findings on the performance of Eigentaste 5.0.

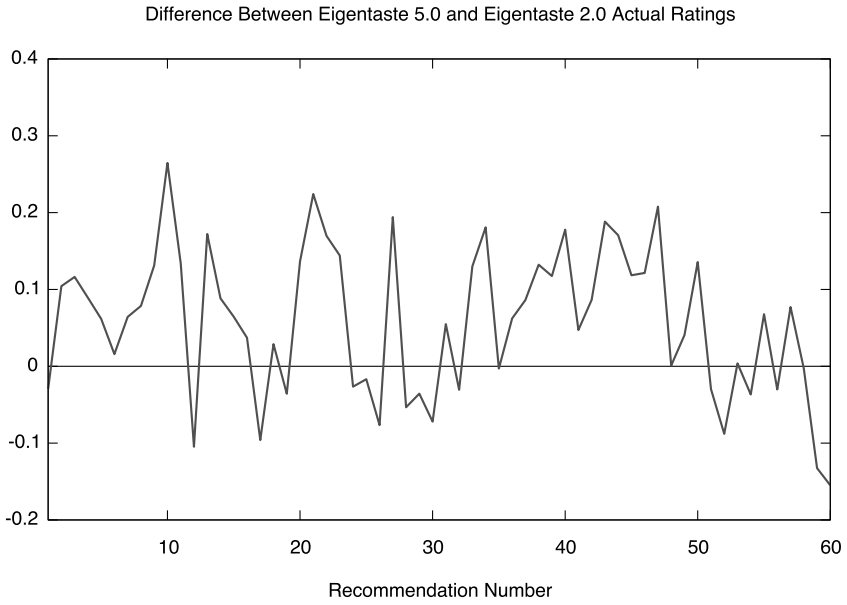


Figure 4.4: Average difference (across 7,000 users) between actual ratings for Eigentaste 5.0 and 2.0 for the i^{th} recommended item.

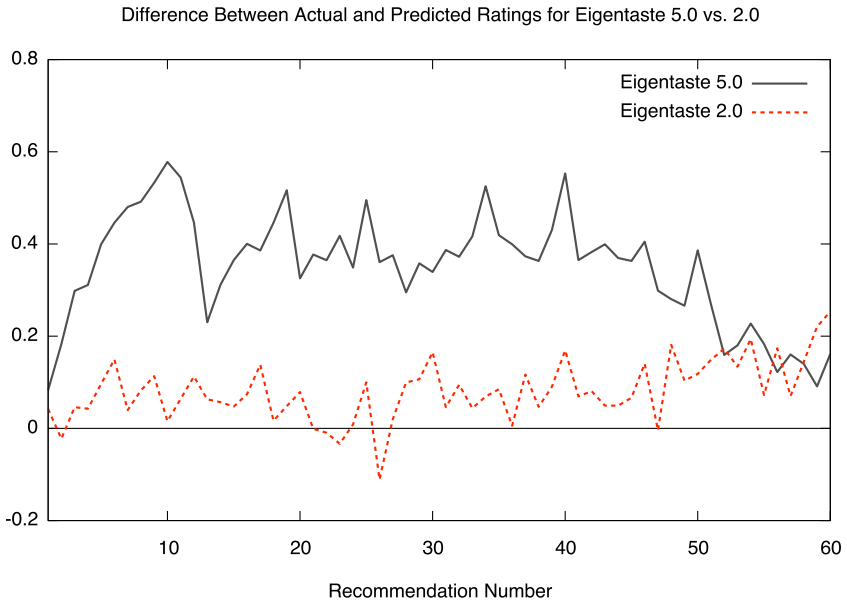


Figure 4.5: Average differences (across 7,000 users) between actual and predicted ratings for Eigentaste 5.0 and 2.0 for the i^{th} recommended item.

4.3.1 Backtested data

We simulated Eigentaste 2.0 and the dynamic recommendations of Eigentaste 5.0 with the Jester data collected between 1999 and 2003. The users are randomly partitioned into equal sized sets of "existing" and "new" users, and "existing" users are clustered using principal component analysis. The item space is clustered using a k-value of 15, and we use a 5 item moving average for each item cluster to track user preferences. We iteratively introduce the "new" users into the system and determine the order that the respective algorithms would recommend items. The two sequences of ratings (and corresponding predictions) for each user are recorded in this order. We average the ratings for the i^{th} recommended item across all "new" users for both actual and predicted rating sequences.

Figure 4.4 shows the difference between the average actual ratings for Eigentaste 5.0 and 2.0. In accordance with our objective, we find that Eigentaste 5.0 provides a distinct advantage over Eigentaste 2.0 for earlier recommendations, particularly within the range of the first 50 items.

The differences between the average actual ratings and the average predicted ratings for each algorithm are shown in Figure 4.5, which illustrates that the error in predictions is significantly greater for Eigentaste 5.0 than for Eigentaste 2.0. This is because the user clusters used to generate predictions only consider ratings for the common set of items, while with Eigentaste 5.0, we can recommend items better suited to a user's interests by using item clusters to take into account the latest user-specific information.

4.3.2 A / B testing

From August 2007 through January 2011, we conducted an A / B test where 50 percent of all new Jester users were randomly assigned to a version of the site running the original Eigentaste 2.0 algorithm, and the other 50 percent were assigned to a version of the site running Eigentaste 5.0. Overall, we collected 860,430 joke ratings from 28,158 users working with Eigentaste 2.0, and 780,204 ratings from 28,045 users working with Eigentaste 5.0.

As shown in Figure 4.6, the data collected over this four year time period roughly confirms the analysis predicted by our backtested data. On average, users tend to give higher ratings to the jokes recommended by Eigentaste 5.0 as compared to those recommended by Eigentaste 2.0. This is especially true in the earlier recommendations (first 25) as opposed to later recommendations. Since entertainment is the primary use of Jester, it is especially important that the recommendation algorithm used provide better recommendations early on, otherwise users may lose interest.

Figure 4.7 shows that the total number of jokes rated by users of the two algorithms is roughly the same, with a sharp, significant spike for Eigentaste 5.0 users in the earlier stages of system usage. This loss of interest may be attributed to the special case of the cold start problem faced by Eigentaste 5.0, as the algorithm requires a short ramp-up phase in order to determine the current "mood" or preference of the user. It is possible that during this phase, users will lose interest and leave the site. However, as evidenced

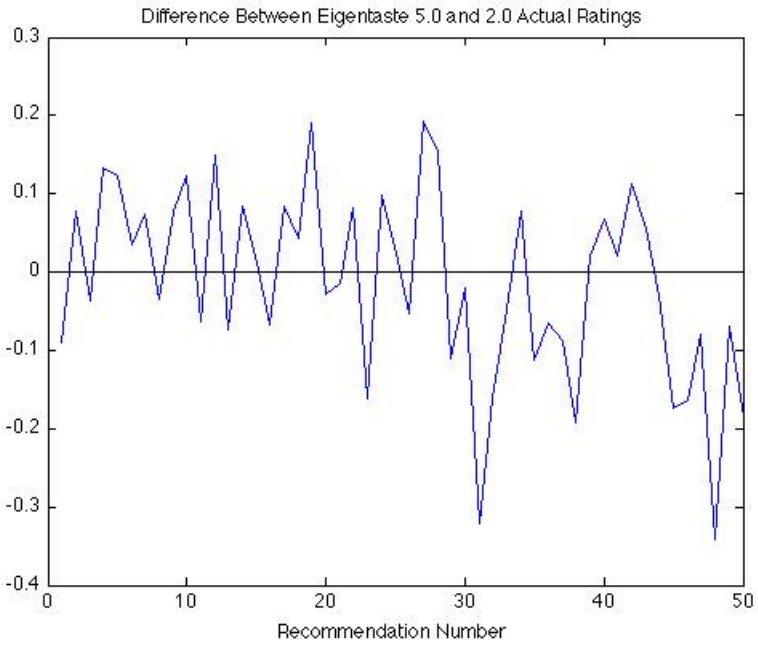


Figure 4.6: Difference of average ratings for the i^{th} joke recommended by Eigentaste 5.0 and 2.0.

by the histogram of joke ratings in Figure 4.8, users of Eigentaste 5.0 tended to use the positive side of the rating scale slightly more frequently than users of Eigentaste 2.0.

4.3.3 Discussion and future work

In this section we presented Eigentaste 5.0, a new constant-time algorithm to dynamically tailor recommendation sequences to user preferences by integrating user clustering with item clustering and monitoring item portfolio effects. We presented results from preliminary backtesting experiments, which we expect will be a lower bound on actual performance that we will evaluate with Jester 5.0.

We will also experiment with generalizations of the adaptive aspect of Eigentaste 5.0, where we recommend items by cycling through the top n item clusters for a user as opposed to just one. Doing so would introduce more diversity among recommendations and may further reduce portfolio effects. We will also experiment with giving users the ability to modify how much item similarity they desire.

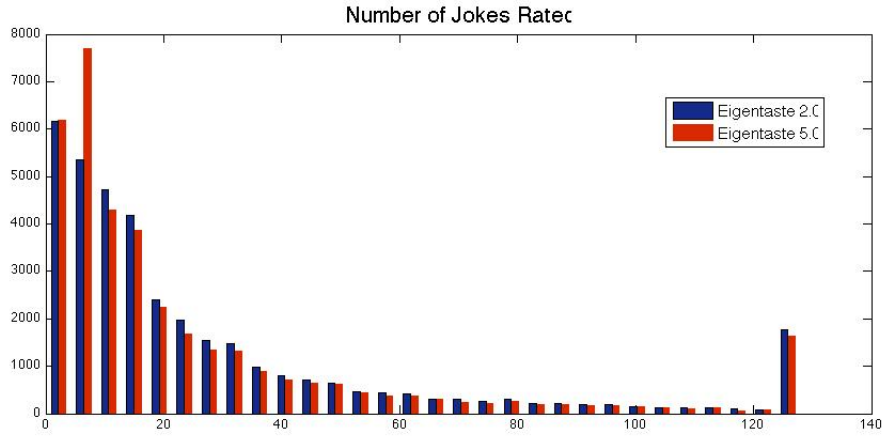


Figure 4.7: Histogram of the number of jokes rated by users of Eigentaste 5.0 and 2.0.

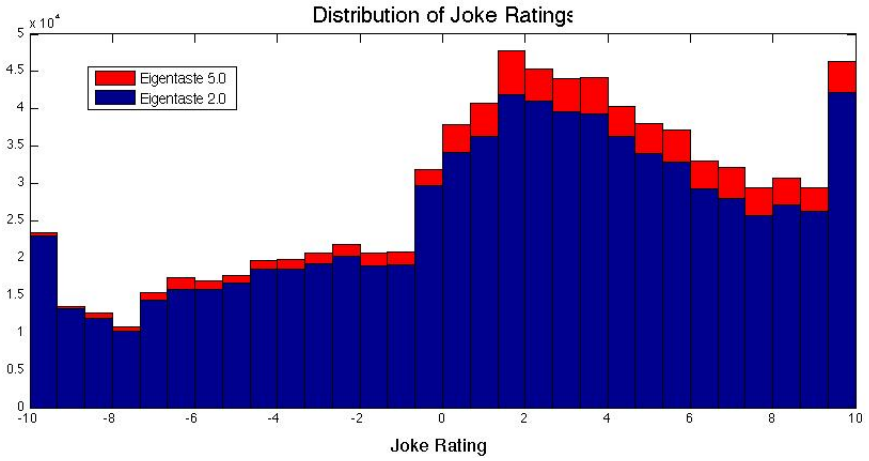


Figure 4.8: Distribution of the joke ratings collected by Jester, where ratings range on a continuous scale between -10 and 10.

4.4 Recommending Donation Portfolios

4.4.1 Motivation

There are over 1.8 million registered non-profit organizations in the United States (64), but effectively allocating one's personal funds among these causes can be a daunting task. Many non-profit organizations do not have the resources to effectively advertise their causes, and as a result most people have only heard of a select few non-profit organizations.

In this section, we describe our experience developing Donation Dashboard, an online collaborative filtering tool that recommends non-profit organizations to users in the form of a weighted portfolio of donation amounts. We explain how we populated the system with data, the design decisions motivating our user interface, and our algorithm for generating portfolios using predicted ratings. We also provide an analysis of the data collected since the launch of Donation Dashboard and measure prediction accuracy using our dataset of ratings for non-profit organizations.

4.4.2 Related work

There are currently many websites and organizations that provide information about non-profits in the United States including GuideStar, Charity Navigator, the BBB Wise Giving Alliance, and the American Institute of Philanthropy. These websites provide a wealth of information about any specific non-profit organization and also provide rankings to make it easier for people to find the non-profit organizations to which they may wish to donate. Charity Navigator, for example, reviews and rates all charities based on their performance in three areas: financially, accountability and transparency, and effectiveness and results. Under this evaluation criteria, the site features many different "top ten" lists that highlight the best and worst organizations. While Charity Navigator and the other sites listed above make available an impressive amount of information, not one of them provides users personalized recommendations tailored to their specific interests. This can make the process of researching organizations significantly more tedious, and it keeps smaller, more specialized organizations out of the spotlight.

One of the challenges in recommending a donation portfolio to a user is that the set of recommended items may contain items that the user has already rated. This is distinctly different from traditional recommender systems such as the one used by Amazon.com and Netflix.com, where users are clearly not interested in being recommended items they have already evaluated. (A person who has seen and rated The Godfather should obviously not be recommended The Godfather.) This paradigm shift requires collaborative filtering algorithms to compare items for which we already have obtained user ratings with those we have not. We can assume that the ratings provided by the user are fairly accurate or *certain*, whereas for the remaining items the best we can do is predict how they will be rated based on the ratings of users with similar rating patterns. Naturally, there will be some amount of variability in the accuracy of these

ratings, and this needs to be considered when comparing items.

Beyond the space of charitable giving, there are several existing examples of recommender systems that form an implicit portfolio of recommendations that may include items already rated by the user; the defining characteristic of these applications is that the user may wish to consume an item more than one time. In music recommender systems, for example, the user may wish to hear the same song multiple times over the course of one or more listening sessions. The collection of songs played to the user can be implicitly viewed as a portfolio of recommendations weighted by the frequency at which each song will be played. Anglade et. al. present a graph-based approach to create personalized music channels that broadcast the music shared by community members in a peer-to-peer fashion (4). Yoshii et al (159) developed a sophisticated hybrid approach to recommending music that incorporates both user ratings and acoustic features of the songs. While there are naturally many additional works in this area, much of the research today focuses on building algorithms that are able to accurately predict how a user will rate a song. This is certainly a valuable and important step in an effective recommender system, however many studies fail to address how to actually *recommend* songs (or other items) given the predictions. Many studies default to ranking items in decreasing order of predicted ratings, but a more sophisticated approach may be necessary that takes into account the variance of these predictions and the diversity of the list or portfolio of recommended items.

Ali and van Stam describe the TiVo recommender system in (2) and propose the idea of portfolio-based recommendations, arguing that the system should strive to recommend the best *set* of items as opposed to maximizing the probability that individual items will each be rated highly. Recommendation lists generated by the latter strategy tend to exhibit low levels of diversification; that is, items of the user's favorite and/or most frequently rated genre are recommended more frequently. This is commonly referred to as the *portfolio effect*. Ziegler et al. study ways of mitigating the portfolio effect by improving topic diversification in recommendation lists (161). They model and analyze properties of recommendation lists, including a metric for intra-list similarity that quantifies the list's diversity of topics. Zhang and Hurley (160) model the goals of maximizing diversity while maintaing similarity as a binary optimization problem and seek to find the best subset of items over all possible subsets. Wang and Zhu (151) take a risk-management approach in document ranking algorithm for information retrieval that considers the risk (variance) of an item and inter-item correlation in addition to the expected relevance (mean).

Most recommender systems today are what we call "single-shot" systems, where the user submits a query and is presented with a fixed set of recommendations. Depending on the application, the user can then do what she chooses with the recommendations, such as purchase or consume an item in some way. If the user is unsatisfied with the results, then she must go back and revise her query by teaching the system more about her tastes; this is often done indirectly by providing more item ratings. Conversational recommender systems are a new class that allows users to continually refine their queries. Some such systems guide the user through the query building process by

prompting the user for information. This can be done intelligently by asking the user for the information that will maximize our understanding of the user's preferences, for example by using the Information Gain model (89). Like Donation Dashboard, conversational recommenders will often present items that have been previously recommended to the user. Recent papers on conversational recommender systems include: (16; 67; 145; 153), and McGinty and Smyth consider various forms of user feedback in (96).

Just like with the Jester interface, Donation Dashboard uses a visual analog scale (VAS) to collect ratings from users. The boundaries of the scale were labeled with the goal of encouraging correct user behavior. For a discussion on the rating scales used with recommender interfaces and the effects of their designs on user actions, see (27).

4.4.3 Description of the Donation Dashboard system

In this section I provide a detailed description of the Donation Dashboard system, including design decisions motivating the user interface and the algorithms used to generate personalized recommendations for users.

4.4.3.1 User interface

New users of the system are presented with non-profits one at a time in the form of a name, logo, motto, website URL, and short description. Figure 4.9 shows a screenshot of a sample display.

The short descriptions of the charities were assembled by manually selecting statements from non-profit information sources and official charity websites; we chose statements that we felt best described the mission and activities of each organization. We aimed to be as unbiased as possible while allowing users to quickly and easily digest the nature of each organization so that they can provide us with an informed rating.

Up until January 23, 2009 we also included an "efficiency" percentage for each non-profit, which is defined as the percent of funds spent on programs (as opposed to overhead and other administrative costs). We included this metric to inform people with what percent of their donations are spent on programs, but we received a significant amount of feedback from users who felt that it is not fair to judge an organization's financial practices based solely on this metric.

Below the description of the nonprofit we display a real-valued slider with which the user is asked to indicate to what extent she is *interested* in donating to the corresponding organization, ranging from "Not Interested" to "Very Interested." Because nonprofits generally provide a beneficial public service, we chose this vocabulary as opposed to asking the user how much she "likes" or "dislikes" the organization under the assumption that users would be uncomfortable indicating dislike for such an organization.

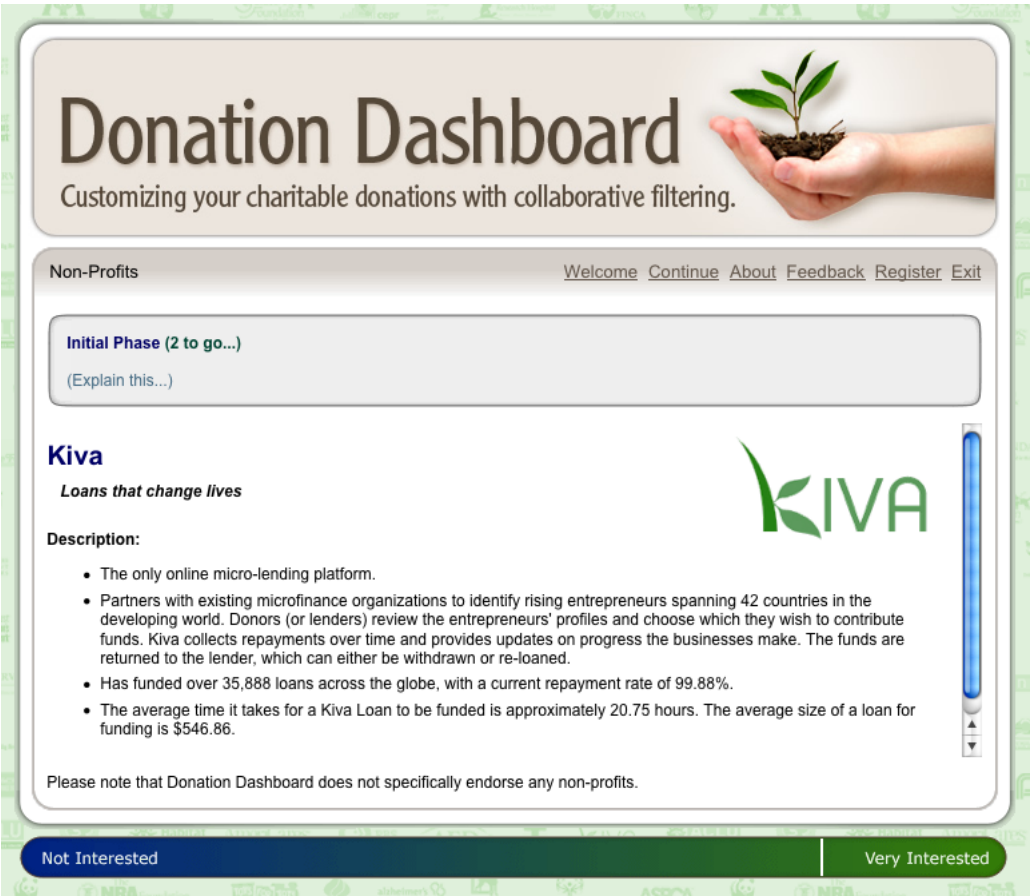


Figure 4.9: Screenshot of a single non-profit in Donation Dashboard 1.0, with the continuous rating bar at the bottom.

4.4.3.2 System Usage

From the perspective of a new user, Donation Dashboard 1.0 works as follows: first, the user is presented with an initial set of 15 non-profit organizations that are presented one at a time. After reviewing the information provided about the currently displayed organization, the user is asked to indicate the extent to which she is interested in donating to that organization by using the VAS slider bar at the bottom of the page. The user records her rating by clicking the point along the continuum that she feels best represents her interest level, ranging from "Not Interested" to "Very Interested." When the mouse button is released, the rating is stored in our database and the system presents the user with the next nonprofit. Users cannot use the "back" button to change their ratings, as the system can only propel the user forward in the rating collection process. This process continues until the user as rated all 15 non-profits, at which point the system presents the user with a weighted portfolio of recommended

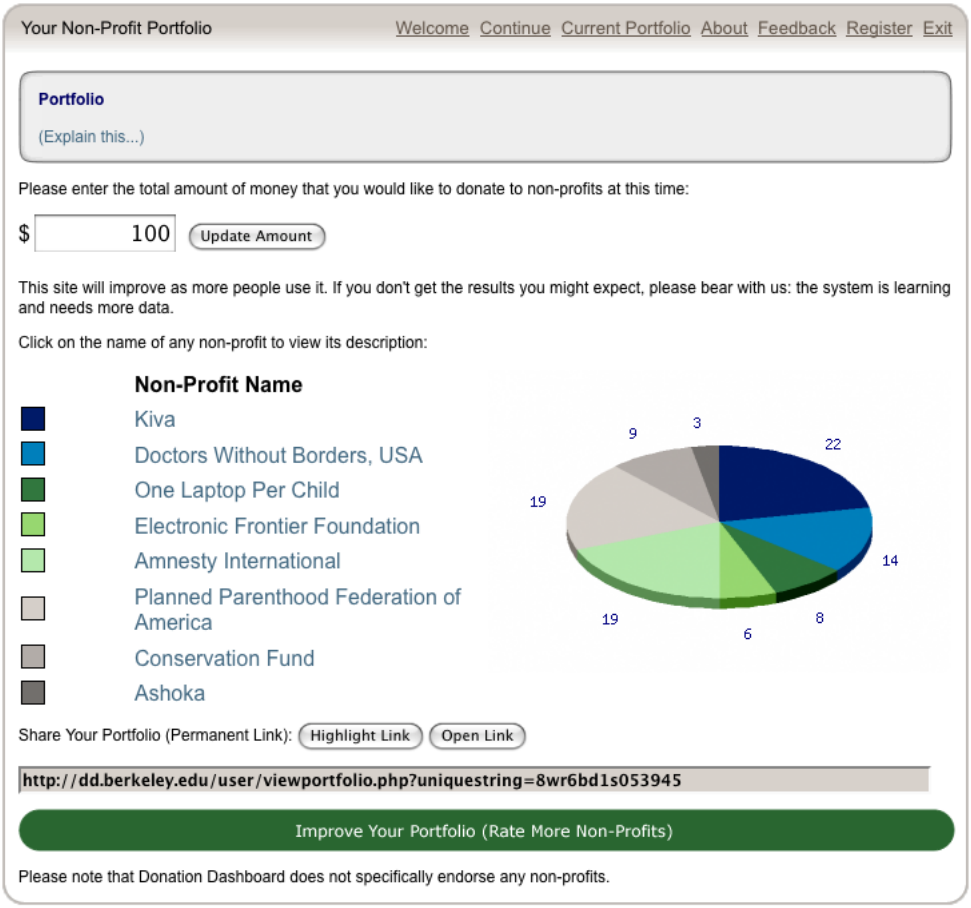


Figure 4.10: Screenshot of a donation portfolio in Donation Dashboard 1.0.

non-profits and donation amounts. The algorithm by which the portfolio is computed is described below in Section 4.4.4.1. If the user is not satisfied with the recommended portfolio, she as the option to refine or improve it by rating five additional charities; this improvement process can be repeated as many times as the user wishes.

The first five non-profit organizations presented comprise a fixed "gauge set" of items that every new user is asked to rate, as described further in Section 4.4.4. The next five are a "seed set," or organizations with the least amount of ratings in the system. Using a seed set ensures that as long as the total number of organizations in the system is relatively small they will each end up with a significant number of user ratings. The last five of organizations are those with the highest predicted ratings via the collaborative filtering algorithm used (see Section 4.4.4). If the user opts to continue rating items to improve his portfolio, he is presented with organizations in descending order of their predicted ratings.

4.4.3.3 Populating the System

As Donation Dashboard 1.0 was meant to be a prototype and we had limited human resources to vet and assemble information about the organizations, we initially populated it with 70 different non-profits. Since we were starting with a relatively small number of organizations, we avoided localized ones so as to appeal to the interests of a larger population. In future iterations of the system, we may wish to build localized instances that recommend local non-profits in addition to the organizations whose activities cover larger parts of the United States. Because local non-profits are often less known than larger, national organizations, providing a localized service would be of great benefit to citizens; the main obstacle here is to build a big enough user base so that reliably good recommendations can be made.

When selecting the set of non-profit organizations to include in Donation Dashboard, we sought to cover a diversity of interests, from environmental policy to disaster relief to micro-lending. We chose to include both big name organizations such as the Red Cross in addition to lesser known ones such as Kiva and Ashoka. The full list of organizations can be accessed by downloading the Donation Dashboard data set at `http://dd.berkeley.edu/dataset`.

4.4.4 Recommending portfolios

Donation Dashboard 1.0 uses the Eigentaste 2.0 algorithm (60) to generate a list of recommended non-profits. Eigentaste 2.0 is a constant online-time collaborative filtering algorithm that addresses the *cold start* problem by collecting real-valued ratings of a "gauge set" of items from all users. This gauge set consists of the organizations with the highest variance in user ratings, which thus maximizes the amount of information gained about new users with as little effort (on behalf of the users) as possible. The result is a completely dense ratings matrix of the gauge set items, which allows for the quick identification of users with similar interests.

Eigentaste 2.0 is divided into an offline phase and an online phase. Offline, principal component analysis (PCA) is applied to the ratings matrix and the users are projected onto the two-dimensional eigenplane. Due to a high concentration of users around the origin, a median-based clustering algorithm referred to as recursive rectangular clustering is used on the lower-dimensional space to divide users into clusters. This method ensures that cluster cell size decreases near the origin, resulting in evenly populated clusters.

Online, the position of a new user in the eigenplane can be determined in constant-time by taking the dot product of the user's ratings vector of the gauge set items and the first two principal components of the ratings matrix. Given the position of a new user u, we can determine to which cluster u belongs in time proportional to the number of clusters in the system. The predicted rating for user u of item i is the average rating of i by all users in the same cluster.

4.4.4.1 Generating Portfolios

Several factors motivate the choice of a weighted item portfolio over competing alternatives such as an un-weighted list of recommendations or a single organization. First, we believe that the natural visualization of a portfolio as a pie chart provides a more engaging user experience. Second, it emphasizes the point that we are not trying to determine which nonprofits are "bad" or "good," but rather that we seek a customized recommendation of donation *amounts*.

One of the greater challenges in recommending a portfolio of donation amounts to nonprofit organizations arises with the idea that we do not want to exclude organizations that the user has already rated; this is because the item class is not meant for one-time consumption, unlike movies or books. The concept of a portfolio is also applicable to other item classes such as stock investments. A less obvious but equally relevant application is the recommendation of music playlists; in this case, we might have a set of songs that the user enjoys and the problem would be to determine with what frequency each song or artist should be played.

For Donation Dashboard, we split the portfolio into two sections, A and B. Section A consists of the items with the highest predicted ratings that were not yet rated by the user, and it makes up ρ_A percent of the portfolio. Section B contains items rated highest by the user and comprises ρ_B percent of the portfolio, where $\rho_B = 1 - \rho_A$. At this time, sections A and B each contain a fixed number of items, and we weight items in the portfolio recommended to user u as follows.

Let r_i^u be the rating assigned to item i by user u, and let r_i^C be the mean rating of item i by users in cluster C, where $u \in C$. Item i is the item with the highest average rating in C such that $i \notin A$. We normalize the ratings for each item $a \in A$ with respect to i as follows

$$r'_a = r_a^C - r_i^C \tag{4.3}$$

Similarly, if item j is the highest rated item by user u such that $j \notin B$, then we normalize the ratings for each item $b \in B$ with

$$r'_b = r_b^u - r_j^u \tag{4.4}$$

Let $S_{A+B} = \sum_{a \in A} r'_a + \sum_{b \in B} r'_b$ be the sum of the normalized ratings for all items in the portfolio. Then the weight w_a of item $a \in A$ such that sum of the weights in A takes up ρ_A percent of the total portfolio is

$$w_a = \left(\frac{\rho_A S_{A+B}}{\sum_{a \in A} r'_a} \right) r'_a \tag{4.5}$$

A similar computation can be done to determine the weight of item $b \in B$, and all of the weights can be further normalized so that they sum to a specific donation amount, as determined by the user.

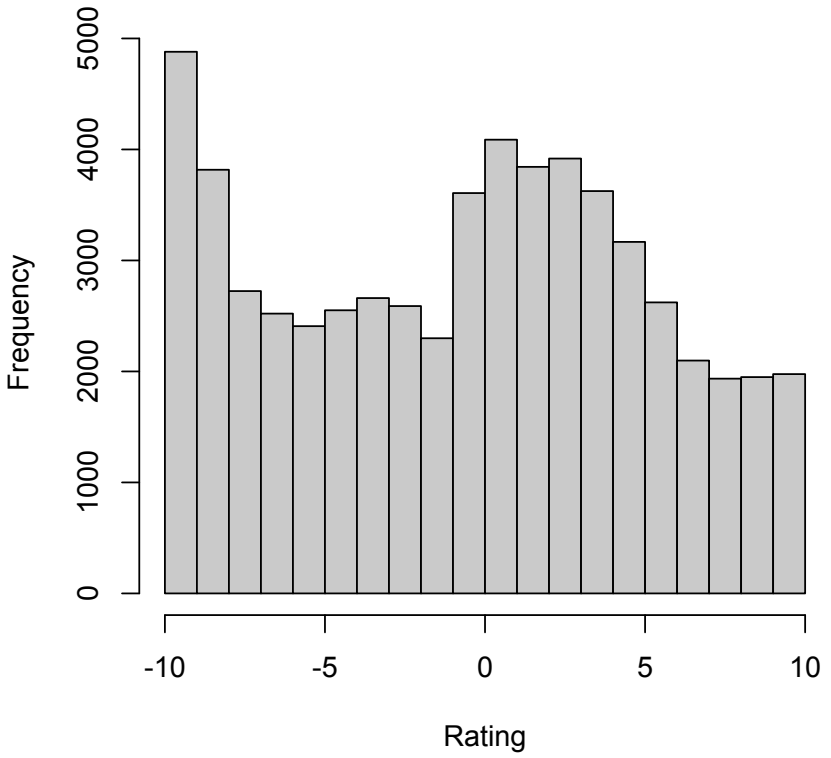


Figure 4.11: Histogram of Donation Dashboard ratings (in the interval [-10.00, +10.00]).

4.4.5 Empirical results

Donation Dashboard 1.0 launched on April 21, 2008. As of May 8, 2009, it has collected over 59,000 ratings of 70 non-profit organizations from over 3,800 users. Organizations in our database have received an average of 846.84 ratings, where the the most rated organization has received 3,061 ratings and the least rated organization has received 491 ratings. The average rating for a nonprofit is -0.65, where ratings can range between -10.00 and 10.00. A histogram of all ratings is shown in Figure 4.11.

Non-profit	**Avg Rating**
Doctors Without Borders	2.69
Public Broadcasting Service	2.34
Kiva	2.31
Planned Parenthood	1.60
Engineers Without Borders	1.56

Figure 4.12: Top non-profit organizations as of 15 October 2010, ranked by mean user rating.

The most highly rated organizations and their mean ratings are listed in Figure 4.12, and the organizations with the lowest ratings are listed in Figure 4.13. Those with the largest variance in ratings can be found in Figure 4.14.

Non-profit	Avg Rating
NRA Foundation	-6.37
Heritage Foundation	-5.10
PETA	-4.90
Boy Scouts of America	-3.74
Prison Fellowship	-3.65

Figure 4.13: Least popular non-profit organizations, ranked by mean user rating.

Non-profit	Variance
National Public Radio	36.89
Humane Society	34.04
Wikimedia Foundation	33.68
St. Jude Children's Research Hospital	32.99
ASPCA	32.40

Figure 4.14: Nonprofit organizations with the largest variance in user ratings as of 15 October 2010.

We use Mean Absolute Error (MAE) as described in (15) to evaluate Eigentaste 2.0 on the Donation Dashboard dataset, and we also list the Normalized Mean Absolute Error (NMAE). We tested the performance of the algorithm under various numbers of user clusters, ranging from 0 (the Global Mean algorithm) to 64. The results of our analysis are described in Figures 4.15 and 4.16.

# Clusters	MAE	NMAE
0	4.295	0.215
2	4.196	0.210
4	4.120	0.206
8	4.043	0.202
16	4.059	0.203
32	4.083	0.204
64	4.244	0.212

Figure 4.15: Performance of Eigentaste 2.0 with different numbers of clusters.

The error initially decreases as the cluster count increases; however, the error increases once the cluster count reaches 16. As we collect more data and more users, the

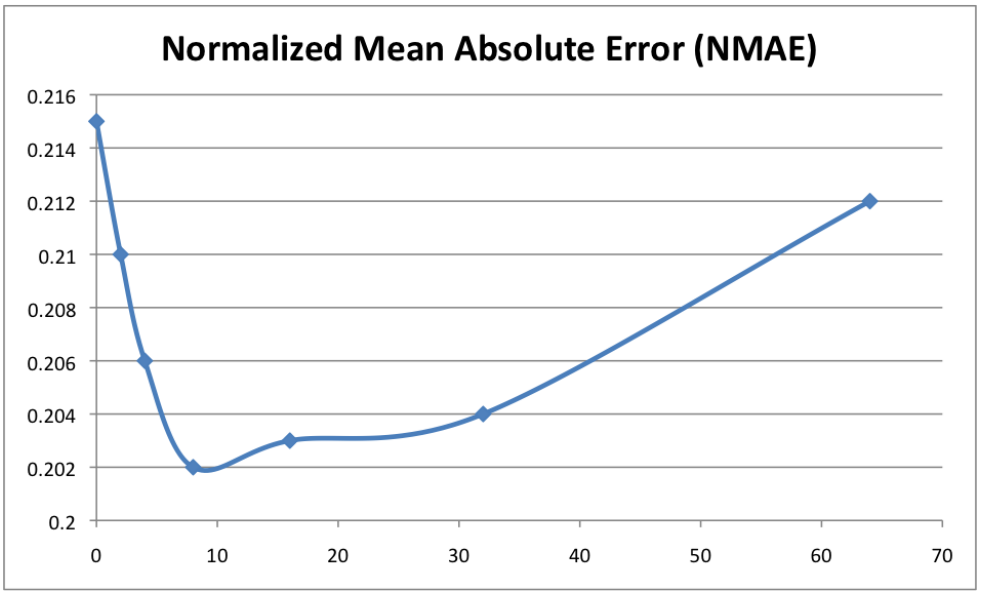


Figure 4.16: Performance of Eigentaste 2.0 with different numbers of clusters.

MAE for higher cluster counts should decrease. Note that users are currently divided into 8 clusters on the live system, which is optimal for now.

We also found that the order in which items are presented can significantly bias user ratings. Let $\bar{\delta}_{ij}$ be the average difference of user ratings for item i and item j when i is shown before j. We measured $\Delta_{ij} = |\bar{\delta}_{ij} - \bar{\delta}_{ji}|$ for every item pair (i, j), and plotted the number of item pairs that fall into different ranges of Δ in Figure 4.17. Ratings are normalized to a [0,1] scale. We observe that the Δ values of more than half of all item pairs exceed 5 percent on the rating scale, and more than 20 percent of the values exceed a difference of 10 percent.

4.4.6 Discussion and future work

One of the greater challenges with this type of application is determining mathematically what makes one portfolio necessarily better than another, and subsequently to design a good portfolio generation algorithm. A possible measure of comparison is the diversity of items recommended in the portfolio, in which case we may seek to recommend a set of organizations that cover a range of issues with minimal overlap. Our next step in this project involves a more in-depth study of portfolio generation methods, particularly when the portfolio may include items already rated by the user.

Data collected indicate that rating individual items on an absolute scale has significant dependencies on the order in which the items are presented. To mitigate this bias, we are developing a graphical model and algorithm that use *relative* ratings to generate portfolios.

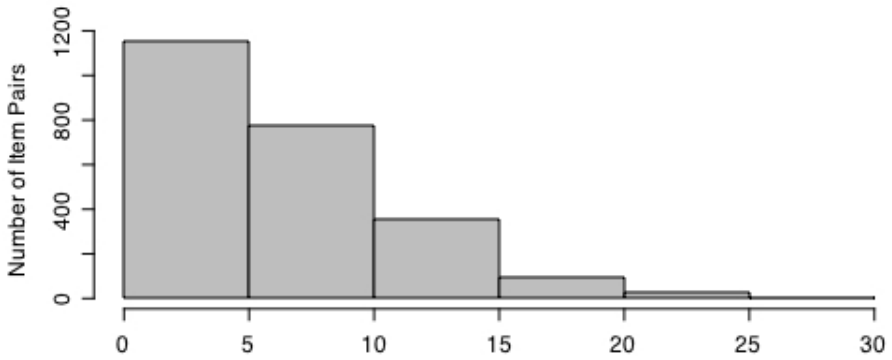


Figure 4.17: Difference between the average rating of an item pair when the presentation order is reversed, as a percent of the rating scale.

We plan to compare Donation Dashboard to other algorithms, particularly Probabilistic Latent Semantic Analysis (PLSA) (76), and to implement attack prevention protocols in the future so that non-profits are not able to promote themselves via false accounts and ratings. Some recent papers on security of recommender systems are (18; 99; 121).

5

A Geometric Model for Visualizing the Diversity of Online Textual Responses

"Opinion is the medium between ignorance and knowledge." - Plato

5.1 Introduction

Most social media websites today are unable to create a space that encourages participants to explore, empathize with, and respond to a diversity of opinions. Currently, sites typically consist of a linear list of responses that may or may not be threaded, and a single topic or thread can receive upwards of thousands of responses within a matter of hours. To help participants cope with information overload, many sites such as Digg and Slashdot ask participants to rate the responses they read and then highlight the responses with the highest average rating. Many sites use a binary thumbs up/down rating system, and some use a five-point Likert scale. The problem with this approach is that only the responses reflecting the most popular points of view are emphasized, discouraging dissenting views regardless of quality; this can silence minority opinions and produce conformist behavior.

In this chapter we present Opinion Space. Accessible at `http://opinion.berkeley.edu`, Opinion Space is a new online system that explores how data visualization models and statistical analysis can be combined to enable crowdsourced insights. It consists of an interactive, dynamic interface that allows participants to visualize and navigate through a diversity of textual responses to a discussion question.

The first version of Opinion Space (v1.0) was released to the general public on March 28, 2009. The project was led by Professor Ken Goldberg and was developed by the following core team of students: Ephrat Bitton, Tavi Nathanson, David Wong, Alex Sydell, and Siamak Faridani. Opinion Space is written in Adobe Flex and powered by the Django web development framework.

In the first few months, Opinion Space 1.0 attracted 21,563 unique visitors of which 4,721 registered with their email address with the purpose of saving their settings. In this in the wild experiment, each registered participant rated on average 14.2 responses. The positive response to Opinion Space motivated us to conduct a controlled user study to quantify and compare Opinion Space with other interfaces in terms of user engagement and the ability to find insightful responses. In this chapter, we present the user interaction design and implementation of Opinion Space and the results of the user study.

5.2 Opinion Space: Motivations and System Overview

In this section section we further motivate the system, which we follow with a description the generalized framework of Opinion Space and its use from the perspective of a new participant.

5.2.1 Motivation and Goals

A central aspect of participatory culture is that users of online sites for news, blogs, videos, and commerce increasingly provide feedback in the form of textual responses. While participatory culture thrives on the sharing of diverse opinions among large populations over the network, there are several problems with existing systems. First, thoughtful moderates are often shouted down by extremists. Online discussions, conducted through threaded lists of responses, often end in "flame wars" predicated on binary characterizations. Second, the amount of data available can be overwhelming. News stories and blog posts often generate hundreds or thousands of responses. As the number of responses grows, presenting them in a chronological list is simply not a scalable interface for browsing and skimming. Third, many websites tend to attract people with like-minded viewpoints, which can reinforce biases and result in "cyber-polarization" (140). With Opinion Space, we aim to address the above problems by incorporating ideas from deliberative polling, dimensionality reduction, and collaborative filtering.

Opinion Space solicits opinions on a set of statements as scalar values on a visual analogue scale (80), a continuous scale from strongly disagree to strongly agree. The statements are designed to elicit a range of responses from participants so as to obtain a better and more differentiating understanding of their unique opinion 'profiles.' Given this data, Opinion Space applies dimensionality reduction to project the participants onto a two-dimensional plane for visualization and navigation, effectively placing all participants onto one level playing field. Points far apart correspond to participants with very different opinions, and participants with similar opinions are proximal. One of our goals is to move beyond one-dimensional characterizations of opinion: the arrangement of points is statistically optimized to convey the underlying distribution of opinions and does not correspond to conventional left/liberal and right/conservative polarities. Participants are also asked to contribute a textual response to a discussion

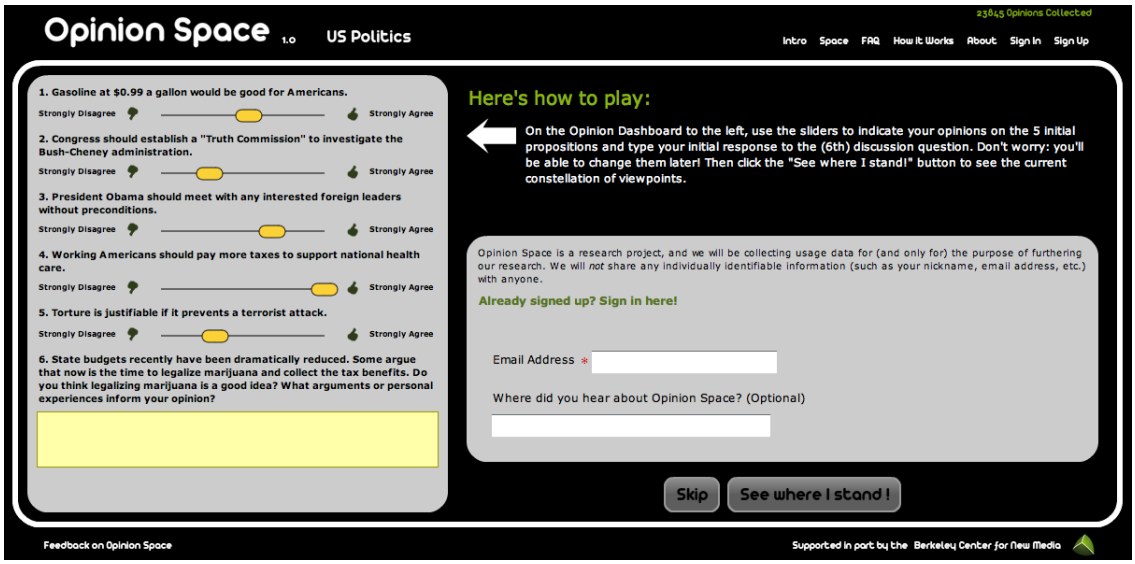


Figure 5.1: Screenshot of sign-up interface for Opinion Space 1.0.

topic; each response is associated with the position of the contributing participant in the visualization of the space. We designed Opinion Space to be a self-organizing system that builds on the wisdom of crowds to highlight the most insightful responses and to reward participants who consider the opinions of those with whom they might normally disagree. Opinion Space is a general tool that could potentially be used to collect and visualize participant opinions on topics ranging from politics to parenting, from art to zoology.

5.2.2 User Experience and Interface Design

Figure 5.1 illustrates the process for signing up with Opinion Space. New participants to the system are presented with five propositions and asked to rate them on a sliding continuous scale between strongly disagree and strongly agree. The propositions can reflect any desired topic that elicits a diversity of opinions from the target population. Example topics include US politics, parenting, philosophy, or almost anything that requires subjective analysis or creative thought.

The numerical ratings collected form an *opinion profile* of the participant corresponding to a point in five-dimensional space. We can compare any two participants in the space via a similarity metric, such as Euclidean distance or Pearson correlation.

Participants are also asked to enter a textual response on the most current discussion topic. Use of this feature is optional, and the participant may elect to enter a response at a later stage.

After the participant fills out the sliders she is presented with the Opinion Space map, which is a projection of the participants onto a two-dimensional plane. See Figure

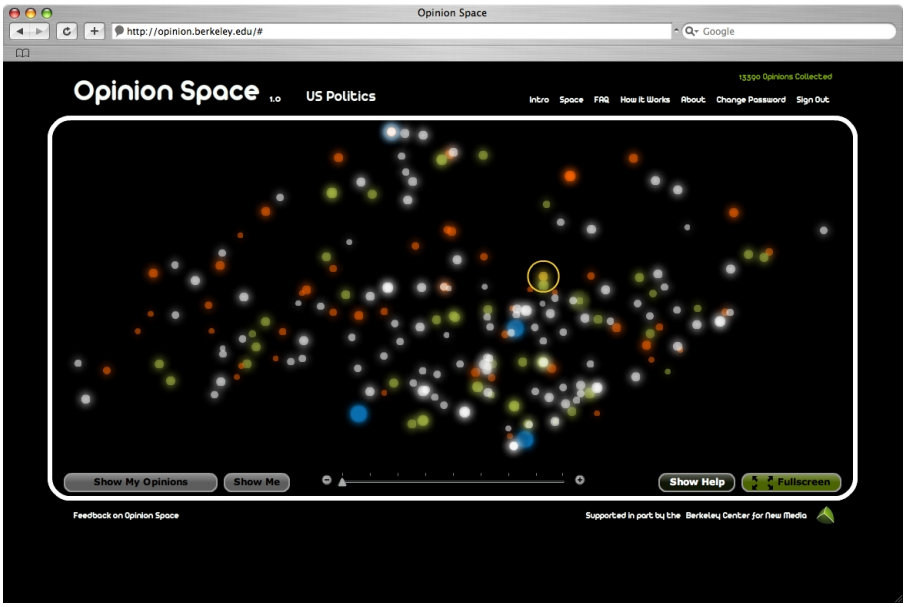


Figure 5.2: A screenshot of the Opinion Space map. The point with the halo corresponds to the position of the active participant. participants can visually measure their distance from famous politicians or political commentators (blue dots). Larger and brighter dots are associated with the responses that are rated more positively by a diversity of participants.

5.2 for a screenshot. The location of the active participant in the map is a function of her opinion profile and is indicated by a glowing point surrounded by a pulsing halo. The positions of other participants are initially displayed as white points. For scalability purposes, a random sample of 200 participants are displayed at a time.

Mathematically, the map is a projection of the five-dimensional opinion profiles onto the two-dimensional plane. We use a statistical technique known as principal component analysis (PCA) to determine the locations of the participants. (See Figure 5.3.) This method allows us to reduce the dimension of the opinion profiles while maximizing variation in distance relationships between participants and operates under the following assumptions:

1. The ratings data can be modeled as a *linear* combination of a set of basis vectors.
2. The mean and variance are sufficient statistics to describe the ratings data.
3. The ratings have a large signal-to-noise ratio.
4. The basis vectors are orthonormal.

Shlens (132) provides an extremely thorough tutorial on PCA and motivates the computational advantages offered by each of these assumptions. It is important to note that since the location of each point in the PCA-generated map is decided purely by

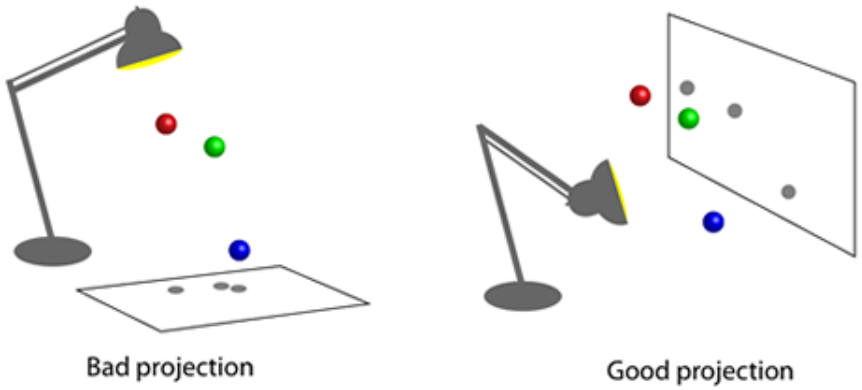


Figure 5.3: In three dimensions, the problem of dimensionality reduction can be thought of as shining a light onto a set of points and observing the resulting shadows. If you looked at just the shadows in the "bad projection," if you would think that the green point is closer to the blue point than it is to the red point, when in reality, the opposite is true. In the "good projection," the distance relationships are better preserved because the angle of the light was adjusted. Similarly, PCA can determine the best projection of the data points into two dimensions.

the mathematics, there is no imposed interpretation of the structure of the points. For example, points on the left do not necessarily correspond to political leftism.

Participants can explore the map by clicking on the points. When a point is selected, a window displays the response entered by the corresponding participant with two prompts, each accompanied by two sliders: 1) "How much do you agree with this response?" and 2) "How insightful is this response?" These two parameters are often, but not always correlated. For example, one may agree with a response but not find it insightful or more importantly, one may disagree with a response but find it highly insightful. Participants earn points based on how others evaluate their response and how they evaluate the responses of others, with special incentives for participants who consider the responses of those far from them (with different opinions on the initial five statements). First proposed by Fishkin (44; 45; 46), deliberative polling is an alternative to traditional polling techniques where participants are first polled on a set of issues, allowed to deliberate for a period of time, and then polled once more. The outcome is often a better understanding of how public opinion would change if people were more informed on the issues. Opinion Space can be thought of as an online, asynchronous version of deliberative polling, where participants can inform each other and adjust their opinions over time.

5.2.3 Releases of Opinion Space 1.0 and 2.0

Here we give the statements and discussion questions published with Opinion Space versions 1.0 and 2.0, which were released to the general public.

5.2.3.1 Opinion Space v1.0

The first version of Opinion Space focused on United States domestic politics. An archive of the site is accessible at `http://opinion.berkeley.edu/1.0`. The statements defining the Opinion Space map were chosen to address a variety of different timely issues that we expected would elicit responses with a high degree of variability:

1. Gasoline at $0.99 a gallon would be good for Americans.
2. Congress should establish a "Truth Commission" to investigate the Bush-Cheney administration.
3. President Obama should met with any interested foreign leaders without preconditions.
4. Working Americans should pay more taxes to support national health care.
5. Torture is justifiable if it prevents a terrorist attack.

In conjunction with these statements, we released the following two discussion questions.

1. The U.S. economy is in turmoil. Nobel Prize winning economist Paul Krugman warned of a "crisis in confidence" over a year ago. Do you have a personal experience that illustrates this crisis in confidence? And what strategies might be effective to restore the confidence of American citizens?
2. State budgets recently have been dramatically reduced. Some argue that now is the time to legalize marijuana and collect the tax benefits. Do you think legalizing marijuana is a good idea? What arguments or personal experiences inform your opinion?

For the first question, we collected 13,111 agreement ratings of 1,601 responses. For the second discussion question, we collected 3,921 agreement ratings of 550 responses. Note that in this initial release of the system, participants were asked to rate responses based only on agreement and not on insightfulness.

5.2.3.2 Opinion Space v2.0, with the US Department of State

Opinion Space version 2.0 (`http://state.gov/opinionspace`) was released in conjunction with the US Department of State with the objective of soliciting insights on foreign policy issues. The statements used in this release were the following:

1. The most urgent security threat to the United States is a terrorist armed with a nuclear weapon.
2. Continuous diplomatic efforts are required to produce lasting, sustainable peace in Afghanistan and Pakistan.
3. Climate change poses a threat to political stability around the world.
4. Investing to increase food production in other countries will ultimately benefit me and my family in the future.
5. The best way to advance a country's economic development is to empower its women.

We released the following three discussion questions in conjunction with the above statements. The first question was only tested internally and was eventually scrapped in favor of the second question.

1. From your perspective: 1. What U.S. foreign policy innovations in approach and method were most successful in this administration's first year? and 2. What new ideas do you feel would be most effective in making the world safer and more just going forward?
2. If you met U. S. Secretary of State Hillary Clinton, what issue would you tell her about, why is it important to you, and what specific suggestions do you have for addressing it?
3. How can the international community strengthen global efforts to prevent nuclear proliferation?

For question 2, we collected a total of 20,795 insightfulness and 21,191 agreement ratings of 2,149 responses. For the third discussion question, we collected 5,558 insightfulness and 5,566 agreement ratings of 1,139 responses.

5.2.4 Opinion Space in Theory: Generalizing to Other Applications

Although Opinion Space 1.0 focuses on issues pertaining to U.S. politics and 2.0 focuses on U.S. foreign policy, the system is designed to generalize to any number of topics. To create a new instance of Opinion Space, the following components are required.

1. **An overarching theme**. The theme can be on just about any topic requiring thoughtful human analysis and discussion.
2. A fixed set of **at least two statements** regarding issues corresponding to the theme or demographic information such as age or income. These cannot change without losing the position information of previous participants. When building a new Opinion Space, extra care must be taken to select a set of statements that

will elicit a range of responses from the participant base. To understand why, consider the case where every participant rates the statements the same; there would be no spread of opinions in the space and nothing new could be learned by the visualization.

3. A **discussion question** to which participants can provide a textual response. Ideally, the discussion topic should relate to each of the propositions. This way, the spread of responses will correlate to a certain degree with the spread of points in the map. To illustrate, if the statements are on classical musicians and the discussion topic is on fast food in America, the positions of the points in the space would not provide the participant with any intuition regarding the diversity of opinions on fast food. Essentially, the statements are meant as a way to give *context* to the opinion of each participant in the absence of the social cues on which we have traditionally relied.

We envision Opinion Space being useful in a variety of online settings. Aside from topic-based discussion forums on subjects such as parenting and diet, or product review sites such as on Amazon.com, the model can be generalized for any application where visualizing the spread of opinions is valuable. So long as there is a numerical way to mathematically compute some measure of "similarity" between any two participants or entities, a visualization can be created.

For example, we could project a subset of the blogosphere (e.g. food blogs) onto the plane, where the distance between any two blogs is a function of the number of shared links or a text-based analysis. Clicking on a blog's point in the space could then pull up the latest entry or photo in the RSS feed. Similar visualizations could also be made for articles on Wikipedia and tweets on Twitter.

5.3 Related Work

5.3.1 Crowdsourcing Insights

It is important to distinguish Opinion Space as a system for crowdsourcing insights from more traditional polling technology. Obtaining insightful outcomes from a poll requires careful design of the poll, both in terms of the questions asked and the population sampled. Internet polls are often viewed as unscientific, as there is a natural selection bias that is difficult to avoid; for example, participants in poor communities with limited access to the Internet are likely to be underrepresented in such polls. With Opinion Space, on the other hand, we do not seek to make generalized claims about a population, but rather we seek to collect insightful and innovative ideas from those who choose to participate. Opinion Space is more about community-building and idea-generation than about making scientific claims about a population based on data collected with the system. Further, in a scientific poll, those who analyze the contributions of participants are independent of the participants, whereas in Opinion Space those who are providing the textual responses or insights also have the power to rate the responses.

Despite some of the aforementioned problems that have emerged among collaborative online systems, several successful technologies have been developed that rely on human insight to solve problems that are difficult, if not impossible, to quantify or compute; these technologies fall under the umbrella term Games with a Purpose (GWAP), and they provide a rich, game-like incentive structure to attract participants and sustain participation. The first of such games was the ESP Game, created by Luis von Ahn (147; 148) at Carnegie Mellon University and designed to label a vast set of images on the web as a byproduct of humans playing a game. With the success of this game came the design of several others, such as the Foldit video game for solving protein folding problems. These technologies make citizen science significantly more engaging by breaking down human tasks into small, verifiable pieces that can be rewarded with points or other prizes. Amazon.com's Mechanical Turk provides a generalized framework for human computation and a large workforce of online participants willing to complete batches of such tasks for small amounts of money.

The brilliant insight of GWAP is that, with careful design of incentive structures, many people are willing to participate without compensation: the ESP game collected 10 million image labels within a matter of months (148). While GWAP have demonstrated an exciting level of success both in terms of sustained participation and generating meaningful outcomes, with Opinion Space we seek to explore models for collaboration that can be applied towards scenarios requiring greater degrees of interaction and idea-sharing between participants. We want to determine how we can use GWAP-like incentive methods to create the conditions for crowdsourcing insights. Can we solicit participant contributions, not just in the genre of word association games and other small tasks, but in the genre of well-considered, thoughtful insights about economic, social, political, and cultural issues?

5.3.2 Dimensionality Reduction

Opinion Space uses dimensionality reduction to map the opinion data collected from participants (on the fixed set of statements) from five dimensions to two. The idea is to allow participants to easily visualize the spread of opinions in the higher dimension with minimal loss of information. When it comes to dimensionality reduction, there are two families of techniques to consider:

1. **Feature Selection**: A representative subset of the feature space is selected for use. Hence, if we wish to reduce the data to two dimensions, we would select the two features that provide the most information about the participants. For example, in the context of Opinion Space we could use information gain to identify the two statements that yield the most information about the user and use the rating values for these two statements as coordinates in the two-dimensional plane.

2. **Feature Extraction**: All of the observations are combined in some fashion and mapped into a lower-dimensional space. There are a variety of techniques for doing this, both linear and non-linear, the most prominent of which we describe below.

Algorithm 1 Principal Component Analysis (PCA)

Require: Given a set of m centered observations $\mathbf{x}_1, \ldots, \mathbf{x}_m$ ($\mathbf{x}_k \in \mathbb{R}^N$, $\sum_{k=1}^{m} \mathbf{x}_k = 0$)

1: **procedure** PCA($\mathbf{x}_1, \ldots, \mathbf{x}_m$)
2: Compute the covariance matrix: $C = \frac{1}{m} \sum_{j=1}^{m} \mathbf{x}_j \mathbf{x}_j^T$
3: Solve the eigenvalue equation: $\lambda \mathbf{v} = C\mathbf{v}$
4: To project an observation vector $\mathbf{x}$ onto an $n < N$ dimensional space, take the dot product between $\mathbf{x}$ and the n eigenvectors corresponding to the largest eigenvalues.
5: **end procedure**

In Opinion Space, we have chosen to use feature extraction rather than selection. This way, we consider all of the data provided by the participants rather than a small subset.

Principal Component Analysis (81) is a linear technique for mapping observation data into lower dimensions. It is done in such a way that the variance of the mapped or transformed observations is maximized, so as to minimize the amount of information lost. The PCA model makes four assumptions regarding the properties of the data: 1) the observation data can be expressed as **linear** combinations of certain bases or factors. 2) The mean and variance are sufficient statistics to describe the observation data, and hence the data follows a distribution in the exponential family (e.g. Gaussian, Exponential, etc.). 3) The data has a large signal-to-noise ratio. That is, the principal components with larger associated variances correspond to interesting dynamics and those with lower variances reflect noise in the data. And lastly, 4) that the principal components are orthogonal; that is to say, the principal components are uncorrelated with each other. The generalized method for dimensionality reduction using PCA is outlined in Algorithm 1.

In addition to being a well-formed mathematical model for linear dimensionality reduction, one of the strengths of Principal Component Analysis is that it is a non-parametric method, meaning it does not require tweaking of parameters or adjusting coefficients. Furthermore, the solution it returns is always unique, and once the principal components are found, dimensionality reduction becomes a computationally fast procedure. In fact, the coordinate of a point in the projected space can be computed in time linear in the number of features (i.e. statements in the case of Opinion Space). On the other hand, because it is non-parametric it does not allow for the incorporation of a-priori knowledge of the structure of the data (if available).

While PCA is a fast and powerful method, in general it is impossible to linearly separate n data points in $d < n$ dimensions. Kernel PCA (127) is a nonlinear extension to PCA that takes advantage of the idea that the points can often be separated in $d \geq n$ dimensions. Kernel PCA works by computing the covariance matrix of the data *after* being transformed into a higher dimensional space:

$$C = \frac{1}{m} \sum_{j=1}^{m} \Phi\left(\mathbf{x}_j\right) \Phi\left(\mathbf{x}_j^T\right) \tag{5.1}$$

Just like PCA, it then projects the data onto the first k eigenvectors of that matrix. The *kernel trick* is used to avoid much of the added complexity. Essentially, a kernel is used so that we don't have to explicitly map the data into the higher dimensional space. If Φ is the identity, then Kernel PCA is equivalent to PCA. The downside to using this method is that an appropriate kernel must be chosen, and so the solution is not unique as with PCA. The technique is generally more valuable for finding groups or clusters of objects with arbitrary shape, but it does not aim to preserve distance relationships in lower dimensions.

Factor Analysis (FA) is another popular technique used for dimensionality reduction (68). It is a statistical method that attempts to find a smaller set of "factors" or variables that describes the observed data, assuming that the data can be modeled as a linear combination of the factors plus a certain amount of error. It assumes that there is a certain amount of correlation between the observations, and so the number of attributes can be reduced. Another important assumption is that the observations are comparable. Within the context of ratings data collected with Opinion Space, this means that participants would be required to interpret the ratings scale in the same manner. As Brady argues in (13), this is a strong and unrealistic assumption. It implies, for example, that when participant i rates item j with a 4, it means the same thing when participant k rates item j with a 4. Further, we note that a participant's of the rating scale might not even be linear. For example, for a given participant, the difference between a 9 and and 8 on a 10-point scale might not be the same as the difference between a 4 and a 3. It is possible to correct for this to some degree by ipsatizing the ratings, which involves normalizing the ratings for each participant with respect to his/her mean rating; however, this only allows us to compare, for example, participant A's relative preference for item i over j with participant B's relative preference for i over j. It does not allow us to compare participant A's preference for i with participant B's preference for i.

While PCA and Factor Analysis find a set of signals / factors / components that are mutually decorrelated, Independent Component Analysis (ICA) is a model that seeks a set of source signals or factors that are mutually independent. As explained by Stone in (138), the model operates under the observation that "independence implies a lack of correlation, but a lack of correlation does not imply independence." Formally, the goal of ICA is to find a basis such that the joint probability distribution can be factorized for all $i \neq j$:

$$\Pr(\mathbf{x}_i, \mathbf{x}_j) = \Pr(\mathbf{x}_i) \Pr(\mathbf{x}_j) \tag{5.2}$$

The advantage here is that ICA only makes an assumption of linearity of the components and not on the sufficient statistics of those components. Hence, it finds the axes "the most formal form of redundancy" - statistical independence. (138)

Multidimensional Scaling (MDS) is an alternative to Factor Analysis that is used to visualize distance relationships in a low dimensional space (28). Unlike in Factor Analysis where the similarity between two observations is strictly determined by the correlation matrix, with MDS any kind of similarity function can be used. The goal is to rearrange the objects in a low dimensional space (e.g. 2D) so as to reproduce the

observed distances in the higher dimensional space. As with PCA and FA, the orientation of the axis is arbitrary and does not hold any semantic meaning. The strategy employed by the MDS method is to rearrange and compare different configurations with the goal of maximizing goodness-of-fit. The most common measure of goodness-of-fit is *stress* (Φ):

$$\Phi = \sum_{i,j} \left[d_{ij} - f\left(\delta_{ij}\right) \right]^2 \tag{5.3}$$

Where d_{ij} is the distance between i and j in the lower dimensional space, and δ_{ij} is the original input / observation distances. The function f is a nonmetric, monotonic transformation of the observed data.

One of the major strengths of MDS is that, unlike FA, it does not require the underlying data to be distributed as a multivariate normal or the relationship between observations to be linear. It also tends to yield fewer factors than FA, which makes the results in lower dimensions stronger or more meaningful. However, the main downfall of this technique is that it is extremely slow and therefore not scalable. (28) A plethora of techniques have been proposed to solve MDS problems, including singular value decomposition (SVD) techniques and an iterative algorithm known as alternating least squares scaling (ALSCAL) (141), depending on the type of data and application.

5.3.3 Visualizing Social Networks

Opinion Space defines a metric relationship between participants based on similarity of opinion, which lends itself well towards forming a geometrically meaningful visualization of the participants in a two-dimensional plane. In doing so, an underlying network structure emerges in the space as participants interact by rating each others responses.

Understanding the structure of social networks is an active area of research (21). Freeman (49) provides background on visualization in social network analysis, from hand-drawn to computer-generated. Viegas and Donath (146) explore two visualizations based on email patterns: a standard graph-based visualization and a visualization that depicts temporal rhythms of email interactions. They found that the latter complements and enhances the former, suggesting that going beyond visualization of relationships in the graph, which is what we aim to do with Opinion Space, is a more effective way to explore and analyze interactions in social networks.

There are several systems available that were designed to aid in the analysis of social networks by providing effective visualization and navigation capabilities. Morningside Analytics (`http://morningside-analytics.com/`) is a company that develops powerful tools for mapping and visualizing emerging trends in online communities using textual analysis. Sack presents the Conversation Map interface that analyzes messages using a set of computational linguistics and sociology techniques to generate a graphical display of links between messages based on textual content (125). Other visualization interfaces include SocialAction, which, like Opinion Space, allows for the visualization of several social network analysis measures (109). Vizster is a system for visual search and structure analysis (69). Like Opinion Space, Vizster uses proximity

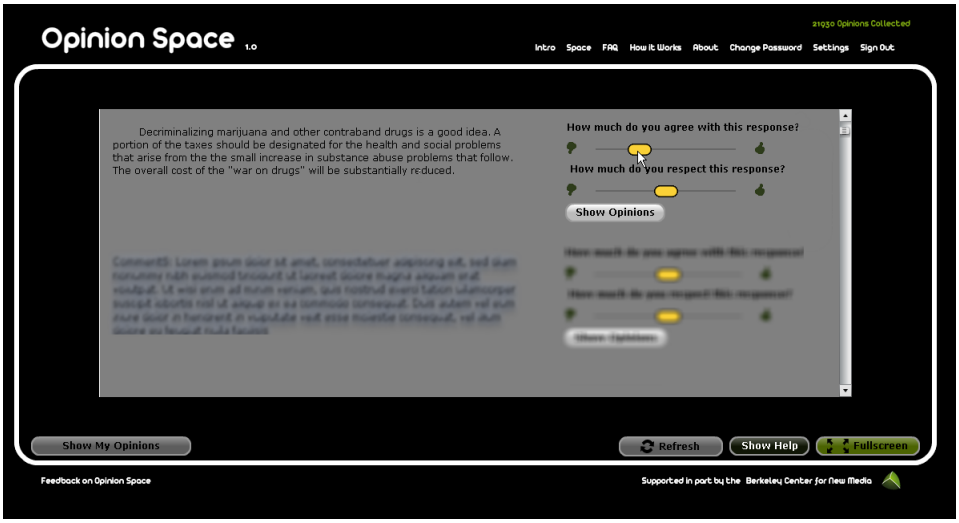


Figure 5.4: Screenshot of interface for linear list of responses.

to highlight similarity. However, Vizster is based on binary connectivity models and does not represent gradations of opinion.

5.4 User Study

In this section we describe the design and results of a user study we conducted on Opinion Space 1.0. Our goal was to compare the Opinion Space interface for browsing textual responses with the more traditional linear list interface in terms of participant engagement with the site. The results of this study were originally published in (39).

5.4.1 User Study Design and Protocol

We created three interfaces, List, Grid, and Space (the last most similar to Opinion Space 1.0), and populated each with a set of 200 randomly selected participant responses from the "in the wild" experiment. We presented each of the interfaces in random order to 12 study participants in a within-subject study using the Space interface as the experimental condition and the List and Grid interfaces as two control conditions, and we recorded data as the participants read and rated the responses of others.

In the following subsections, we describe each of the three interfaces in greater detail, the hypotheses we formed regarding Opinion Space 1.0, and the protocol we followed for conducting the user study.

5.4.1.1 Three Interfaces Compared in Study

1. List Interface

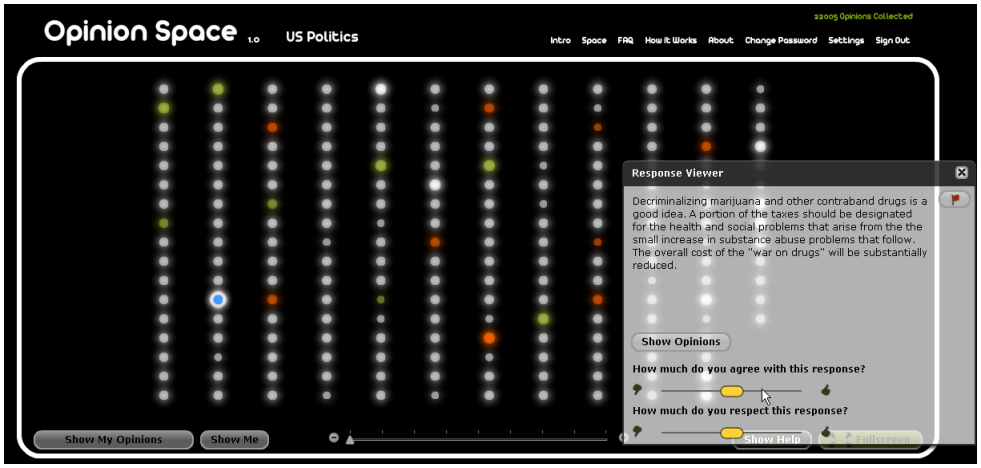


Figure 5.5: Screenshot of interface for responses organized chronologically on a grid.

The List interface (shown in Figure 5.4) is based on standard comment lists found on blogs and other websites. In the List interface, 200 responses are presented in a chronological linear list. We record the amount of time participants spend on every response they view (dwell time) as well as the agree and respect ratings they give to each response. To more accurately measure the time participants spend reading a response, neighboring responses are blurred and then instantly de-blurred as the participant scrolls up or down the list.

2. Grid Interface

The Grid interface (shown in Figure 5.5) is designed to be a control for studying the effect of visualizing the points based on the spread of opinion profile data. The Grid interface is a graphical display similar to Opinion Space 1.0, the primary difference being the positioning of the points. Here, points are ordered on a uniform rectangular grid according to time of entry; the location of a point is only a function of the time it was entered and is independent of the corresponding participants opinion profile. The size and brightness of the points varies with participant ratings, as in the Space Interface. Study participants were asked to click on points in any order they wished and to rate the responses.

3. Space Interface

The Space interface is the experimental condition and is nearly identical to Opinion Space 1.0. We turned off cosmetic features such as the twinkling of points to avoid any bias they might introduce by unintentionally influencing which points participants choose to click.

5.4.2 Hypotheses

We considered the following five hypotheses for our study. One of our primary goals in designing Opinion Space 1.0 was to create a system for browsing online comments that is more engaging than traditional methods (i.e. linear lists). We believed that providing participants with a visual means to interpret the scope of opinions and navigate the textual responses would serve to this effect. That is to say, our assumption was that participants are more likely to engage with others when more contextual information is available and when they are not overloaded with information. By mapping participants onto a meaningful "space" of opinions, our goal was to give participants control over the diversity of the responses they choose to read, hence creating an environment that promotes further engagement. Formally, we hypothesized that:

> *Hypothesis 1 (H1): Opinion Space will be significantly more engaging than List or Grid in terms of average dwell time (H1a) and in terms of participant ranking of overall preference (H1b).*

Since Opinion Space ranks responses by combining participant ratings with metric information about relative opinion positions (See (10) for more details.), it was our hope that participants would have an easier time finding responses they value. Response quality information is communicated visually via the size of a point on the space; larger points correspond to responses that were given higher ratings by other participants, and smaller points were given lower ratings. Since it is also easier to click on a larger point than a smaller point (larger points have more surface area), we expect that just clicking randomly on points will result in viewing a higher proportion of quality responses as a result of this feature. This intuition motivates our second hypothesis:

> *Hypothesis 2 (H2): Participants will report Opinion Space as more conducive to finding useful responses than List or Grid interfaces.*

An important goal for Opinion Space was to expose participants to a wider range of insightful opinions rather than the majority view or the most recently posted responses. We measure the diversity of a response encountered by participant i as the Euclidean distance between participant is opinion profile and that of the participant who wrote the response. We define the diversity of a set of responses encountered by participant i as the average pairwise Euclidean distance between i and each author in the set. By giving participants the option to easily navigate the scope of opinions held by other participants, our third hypothesis is that

> *Hypothesis 3 (H3): Participants of Opinion Space will read a significantly more diverse set of responses than with the List or Grid interfaces.*

Since Opinion Space is designed to highlight the most insightful responses by increasing the size and brightness of the corresponding points in the map, we expect that participants will find and read more responses with which they agree when using the

Space interface. This is also because it is much easier to identify the top responses given the visual clues than to sort through a long linear list of responses that are sorted in chronological order.

> *Hypothesis 4 (H4): Opinion Space participants will report significantly great agreement with the responses of others than they do when using the List or Grid interfaces.*

Finally, motivated by the notion that it is easier to respect the opinion of an individual given more contextual information (such as the political views of that person), it was our hope that participants would report more respect for the responses they read using the Space interface. Although it was still possible for participants to view how others rated the five initial statements, doing so requires an extra step; not only does the Space interface make it fast and easy to interpret this information, but the information is easy to interpret relative to oneself. That is, participants can quickly determine who tended to agree or disagree with themselves, which provides greater context or background information to consider when reviewing a response. Hence, we formed the following fifth hypothesis:

> *Hypothesis 5 (H5): Opinion Space participants will report significantly greater respect for the responses of others than they do when using the List or Grid interfaces.*

5.4.3 Method

To test our hypotheses, we designed a within-subject study using the Space interface as the experimental condition and the List and Grid interfaces as two control conditions. Each participant interacted with all three interfaces, and the interfaces were presented in random order so as to reduce the potential for bias.

User Study Participants 12 participants were selected from a pool of 36 volunteers who responded to our ads posted across the UC Berkeley campus and Facebook. All of the volunteers in that pool completed an online pre-screening survey to ensure that they were not already familiar with Opinion Space 1.0 and that they had a relatively good understanding of current political issues in the US. Each participant was offered a 10 dollar gift certificate to Amazon.com for successfully completing the experiment. We had two female and ten male volunteers participate in the study. Three participants identified themselves as Republican (25%), five as Democrats (42%) and 4 as Independents (33%). Additional information about the participants is provided in Table 5.1.

Protocol Each individual experiment took approximately one hour to complete. Sessions began by having participants use the proposition sliders to enter their own opinion profiles and by having them enter a textual response on the current discussion question

Question	Mean	Variance
Age	19.9	0.9
How tech savvy are you?	5.9/10	4.9
How familiar are you with the current political issues?	6.0/10	3.0

Table 5.1: Characteristics of the 12 user study participants.

regarding the legalization of marijuana. They were then asked to explore each of the three interfaces, which were presented to them in random order. Participants were free to switch to the next interface whenever they wanted so long as they had rated at least 10 responses; we wanted to ensure that participants had at least a minimal amount of experience interacting with each interface. If a participant did not ask to switch to the next interface after 15 minutes, the system did so automatically.

After using each interface and before moving on to the next, participants were given a short questionnaire that asked them to indicate on an integer scale of 1 (not at all) to 5 (very) how enjoyable, interesting, and useful they found the interface. Participants were encouraged to explore each interface freely by reading and rating responses in any order they wished. We automatically recorded participant dwell time for each response. Participants were asked to read responses carefully and rate them individually based on how much they agree with the response and how much they respect it (Figure 3). Upon completion of the experiment, participants were given an exit survey that asked them to rank the three interfaces on a series of 7 qualities.

5.4.4 Results

In this section we describe the results of our study as determined both objectively with numerical, observational data and subjectively through questionnaires completed by the participants.

5.4.4.1 Usage and Survey Data

Table 5.2 shows the mean and standard deviation of the number of responses rated by the participants in each of the three interfaces. The third and fourth rows show the average participant rating of each response on a continuous scale between 0 and 1, in terms of the agree and respect measures, respectively.

Table 5.3 summarizes the mean and standard deviation of participant responses to the short questionnaire asking participants how enjoyable, interesting, and useful they found each interface by providing an integer value from 1 (not at all) to 5 (very much). Table 5.4 summarizes data from the exit survey that asked participants to rank the interfaces after trying all three.

	List	Grid	Space
Average number of responses rated	23.5 (11.2)	20.9 (9.9)	21.1 (9.0)
Average dwell time per response (sec)	516.4 (242.5)	458.4 (180.4)	582.9 (187.1)
Average "agreement"	0.443 (0.266)	0.515 (0.278)	0.567 (0.269)
Average "respect for"	0.396 (0.294)	0.479 (0.300)	0.510 (0.284)

Table 5.2: Average usage data from the 12 study participants. Standard deviations are given in parentheses.

	List	Grid	Space
I found this version of the system enjoyable to use.	2.2 (1.3)	3.3 (1.2)	4.8 (0.4)
I learned something interesting while using this version.	2.9 (0.9)	3.6 (0.9)	4.2 (0.7)
This version is conducive towards finding useful comments.	2.0 (1.2)	3.3 (0.8)	4.2 (0.7)

Table 5.3: Average response data from short questionnaires asking participants to indicate how enjoyable, interesting, and useful they found each interface by providing an integer-valued rating from 1 (not at all) to 5 (very much). Standard deviations are given in parentheses.

5.4.4.2 Carry-over Effect of Participant Fatigue

Interfaces were presented in random order for each participant. To check for the presence of carry-over effects between interfaces due to participant fatigue, we recorded the total time participants spent with each interface as a measure of engagement. We conducted a two-way ANOVA analysis on the distributions of the time participants spent with the first, second, and third interfaces presented to them. Our analysis yielded a p-value of $0.534 >> 0.05$, which suggests that participant fatigue did not cause significant carry-over effects.

5.4.4.3 Response Browsing Strategies

Participants were also asked to report how they selected responses to read in each interface. For the List interface, 6 participants replied that they read the responses in the order they were displayed, and the other half said that they randomly selected the responses.

Question	List	Grid	Space
1. Which version enabled you to read more insightful comments?	16%	8%	75%
2. In which version are you more likely to leave your own comment or response?	16%	16%	67%
3. Which version would you prefer to use if you wanted to participate in a discussion about US politics?	8%	8%	83%
4. In which version do you expect to spend more time reading comments and browsing?	8%	16%	75%
5. Which version highlights the most insightful comments?	8%	33%	58%
6. In which version did you see more diversity among comments?	16%	33%	50%
7. Which version do you prefer overall?	8%	0%	92%

Table 5.4: Summary of responses to the exit survey, which asked participants to rank the interfaces according to various criteria after trying all three.

For the Grid interface, 7 out of 12 people replied that they tried to diversify the responses they read by selecting a balanced combination of large and small point sizes. Four people said that they picked the points in random order and did not pay attention to the point size. Only one replied that she started with the biggest point size and continued in descending order of point sizes. Survey responses for the Space interface are presented in Table 5.5. 11 out of 12 participants reported that their strategy for reading responses with the Space interface was to diversify by clicking on points positioned far from their own.

5.4.5 Evaluation of Hypotheses

To analyze the data collected from our study, we used Analysis of Variance (ANOVA), ANOVA on Ranks, Student t-tests, Friedmans test, Welchs test, and the Wilcoxon signed-rank test for significance, as well as Bartletts test for homogeneity of variance. ANOVA generalizes the Student t-test for measuring the statistical significance of the differences between data sets by analyzing their relative means and variances. Given n data sets, these tests produce a p-value that estimates the probability that the outcome is by chance, ie, that the sets were sampled from the same distribution; known as the null hypothesis. Lower p-values correspond to greater significance of the data.

Performing ANOVA reduces the chances of encoutering type I errors that may occur in executing multiple t-test hypothesis testing (97). Similar to the Student t-test, ANOVA assumes that the observations are normally distributed and that the variances are equal. Before performing ANOVA, we use Bartletts test to make sure

that the homogeneity of variances (homoscedasticity) property holds. If the p-value for this test is high, we can perform an ANOVA analysis on the dataset. For analyzing ranked data (as with hypotheses H1a, H2) we use Friedmans test, which is an extension of ANOVA for nonparametric data (97).

5.4.5.1 Hypothesis 1: Participant Engagement

Hypothesis 1 (H1): Opinion Space will be more significantly engaging than List or Grid in terms of average dwell time [H1a] and based on participant ranking of all three interfaces in terms of overall preference [H1b].

We recorded dwell times for the 959 responses viewed by the participants while working with the three interfaces (participants did not rate all responses they read). There are 329, 285, and 345 dwell times for the List, Grid and Space interfaces respectively. Average dwell times for these interfaces are reported in Table 2.

Bartletts test rejected the assumption of homogeneity of variances for the dwell times, and so we performed a two-way, within-subject ANOVA on Ranks as suggested by (26). For our analysis, the within-subject factor is the type of interface. The resulting p-value ($1.098 \times 10^{-}14$), is significantly less than 0.05 suggesting that the type of interfaces impacted participant dwell times. We used Welchs t-test to measure the extent of this impact, which is a generalization of the Students t-test for cases where the variances are not equal [26]. Pairwise analysis using Welchs test shows that the dwell times in Grid and Space interfaces are significantly longer than the List interface (p-values for Grid-List is $2.2 \times 10^{-}16$ and is $5.387 \times 10^{-}10$ for List-Space, both $<< 0.05$). However, we did not find a significant difference in the dwell times between the Grid and Space interfaces (p-value$= 0.1126 > 0.05$).

We also performed Freidmans test on participant responses to the question: "In which version do you expect to spend more time reading comments?" (Table 4). Freidmans test on this data yields a p-value of $0.0000984 << 0.05$. We used Wilcoxons signed-rank test as a pairwise post-test for nonparametric distributions. The test showed statistical significance between the participant reported ranks for each pair of interfaces (p-values are: 0.02332 for Grid-List, 0.02351 for Grid-Space and 0.002608 for Space-List), which supports H1a.

The self-reported, subjective data suggests that participants are significantly likely to spend more time reading responses on the Space interface, but the observed (objective) data does not show a significant difference between the Space and Grid interfaces.

To assess hypothesis H1b, we consider the data collected from the exit survey question that asked participants to rank the three interfaces by preference. Almost all (92%) of participants reported that they prefer Opinion Space to the List and Grid interfaces (H1b), as shown in Table 4. Friedmans ANOVA analysis on this data produces a p-value $= 0.000486512 << 0.05$, and Wilcoxons signed-rank post-test shows statistical significance between each pair of participant interfaces with p-values < 0.05. The results of this analysis mildly support hypothesis H1b. (for List-Space p-value$=$ 0.01188, for Grid-Space p-value$=$ 0.03884 and for Grid-List, p-value $= 0.0209$).

5.4.5.2 Hypothesis 2: Finding Useful Responses

Hypothesis 2 (H2): Participants will report Opinion Space is more conducive to finding useful responses than List or Grid interfaces.

In the questionnaires following the use of each interface, participants subjectively reported Opinion Space to be more conducive to finding useful responses than the List and Grid interfaces (Table 3). Conducting Freidmans test on this ranked data yields a p-value = 0.00361 << 0.05. Wilcoxons post-test suggests that statistical significance holds for all pairs of interfaces (p-values for the follow up tests are: 0.003583 for Grid-PCA, 0.01868 for List-PCA and 0.03667 for Grid-List), in support of H2.

5.4.5.3 Hypothesis 3: Response Diversity

Hypothesis 3 (H3): participants of Opinion Space will read a significantly more diverse set of responses than with the List or Grid interfaces.

As noted earlier, we define the average diversity of a set of responses rated by participant i as the average Euclidean distance between participant i and the authors of those responses. In the 5D opinion profile vector space, the maximum distance between any two participants is 2.23 units. The average diversity for the 959 responses read by the 12 participants was 0.960, 0.924, and 0.992 for the Space, List, and Grid interfaces respectively. The data passes Bartlett's test for homogeneity of variances with a p-value of 0.1628 > 0.05, and ANOVA yields a p-value of 0.7848 >> 0.05. This suggests that there is no statistically significant difference between the diversity of responses read in each interface; hence, the data does not support H3.

Interestingly, participants (subjectively) perceived greater response diversity in Opinion Space. In the exit Survey, 50% of participants reported Opinion Space allowed them to see more diverse responses; while only 16% chose List and 33% chose Grid, as indicated by Question 6 in Table 4.

5.4.5.4 Hypothesis 4: Agreement with Responses

Hypothesis 4 (H4): Opinion Space participants will report significantly greater agreement with the responses of others than they do when using the List or Grid interfaces.

Participants indicated the degree of their agreement with a total of 782 responses (281 responses in the List interface, 249 in Grid, and 252 responses in Space) on a continuous scale from 0.0 (strongly disagree) to 1.0 (strongly agree). Average values are reported in Table 2. Bartletts test on this data gives a p-value of 0.850 >> 0.05, suggesting that the homogeneity of variances assumption is valid. ANOVA yields a p-value of 0.00002073 << 0.05, and a follow up analysis with a two-tailed t-test shows statistical significance between all pairs of interfaces. P-values for each pair are: 0.03335 for Grid-Space, 0.000000149 for Space-List and 0.002115 for List-Grid, which supports H4.

5.4.5.5 Hypothesis 5: Respect for Responses

Hypothesis 5 (H5): Opinion Space participants will report significantly greater respect for the responses of others than they do when using the List or Grid interfaces.

Participants rated their degree of respect for a total of 782 responses by using a continuous scale from 0.0 (do not respect) to 1.0 (respect greatly). See Table 2 for the average values. ANOVA analysis yields a p-value of $0.001105 << 0.05$, and a follow up analysis with a two-tailed t-test showed that participants exhibited significantly greater respect for responses in both the Grid and Space interfaces as compared to the List interface (p-values are: 0.0007299 for List-Grid and 0.00003479 for List-Space). However, we did not find a statistically significant difference in respect values between the Grid and Space interfaces (p-value of 0.1191). We believe this is because both Grid and Space use the same visual method for highlighting the most insightful responses by adjusting the size and brightness of the points.

5.4.6 Discussion

Conventional list-based comment interfaces do not scale well: as the number of responses grows, participants quickly become overwhelmed and read only a few responses, often the most recent or most extreme as voted by binary "thumbs up / down" ratings. We designed Opinion Space as a scalable way to visualize the "opinion landscape" and to operate as a self-organizing system that encourages participants to find and consider responses written by those who hold opinions different from their own.

We found that participants were significantly more engaged with the Space and Grid interfaces as compared to List in terms of dwell time per response, and participants perceived the Space interface to be significantly more engaging than Grid and List and indicated by subjective rankings of the three interfaces (H1). We also found that participants reported significantly greater agreement (H4) with the responses they read using the Space interface, and they had significantly more respect for responses they read using Grid and Space as compared to List (H5). Our hypothesis that participants would find the Space interface significantly more conducive to finding useful responses (H2) was marginally supported. These results are consistent with the results reported by Ludford et. al (94), where online participants in movie discussion groups were more engaged when the diversity of viewpoints and the uniqueness of each participants opinion were conveyed.

Our hypothesis that participants using the Space interface would read significantly more diverse responses, based on Euclidean distance between responses to the profile statements (H3), was not supported by the data. However, as illustrated in Table 5, study participants describing their response browsing strategies for the Space interface reported that they made use of the specific graphical layout and the position of their own opinion point to seek out responses written by those with a diversity of opinions.

Response diversity was also high with the List and Grid interfaces. The chronological ordering of responses in the List and Grid interfaces induced a random ordering of diversity (relative distances) between responses, so these interfaces were also effective

on average for exposing participants to a diversity of responses. The outcome may have been different if the List interface had been sorted based on binary thumbs up/down ratings, which would highlight more extreme viewpoints. On the other hand, it is interesting and encouraging to note that the graphical display of Opinion Space did not significantly bias participants toward only reading responses written by those with similar opinions.

5.5 Empirical Data Collected Online

In this section we present and discuss the data collected with the second discussion question from Opinion Space 2.0, as it is our richest data set.

5.5.1 Eigenvectors of the Space

Figure 5.6 illustrates the directional and magnitude of influence that each statement has on the position of a point in the Opinion Space map. As the participant adjusts her rating of one of the statements, her point will move parallel to the corresponding line in the figure. Lines with larger magnitude indicate that the point will move farther per unit change in rating. The wider the angle between two lines the less correlated the rating values are for the corresponding statements.

For Opinion Space 2.0, we found that the statement claiming that nuclear weapons is the most urgent security threat to the US is nearly orthogonal to the statements on climate change and proactive democracy (with Pearson correlation values of -0.15 and -0.16, respectively). This implies that there is little to no correlation between the ratings collected along these two dimensions. The remaining three statements, however, have much smaller inner angles and hence appear to be more closely correlated. Furthermore, the two statements that are most highly correlated are the ones on diplomacy in Afghanistan and climate change, which have a Pearson correlation of 0.39.

5.5.2 Insight versus Agreement Ratings

Figure 5.7 is a scatter plot of the insightfulness versus agreement response ratings collected with question 2 of Opinion Space 2.0. As expected, there is a fair amount of correlation ($\rho = 0.7469$) between the two scales, however, there are many responses with which participants agreed but did not find insightful. This indicates that by introducing the two rating scales is indeed helping us separate the agreement signal from the insightfulness signal, which would result in less noise for our ranking algorithms to deal with. Also encouraging is the significant number of participants who found a response to be insightful but with which they did not agree. This indicates that participants were able to put their opinions aside and recognize the insights of others even when they don't agree; this is precisely the behavior we are looking to encourage with Opinion Space.

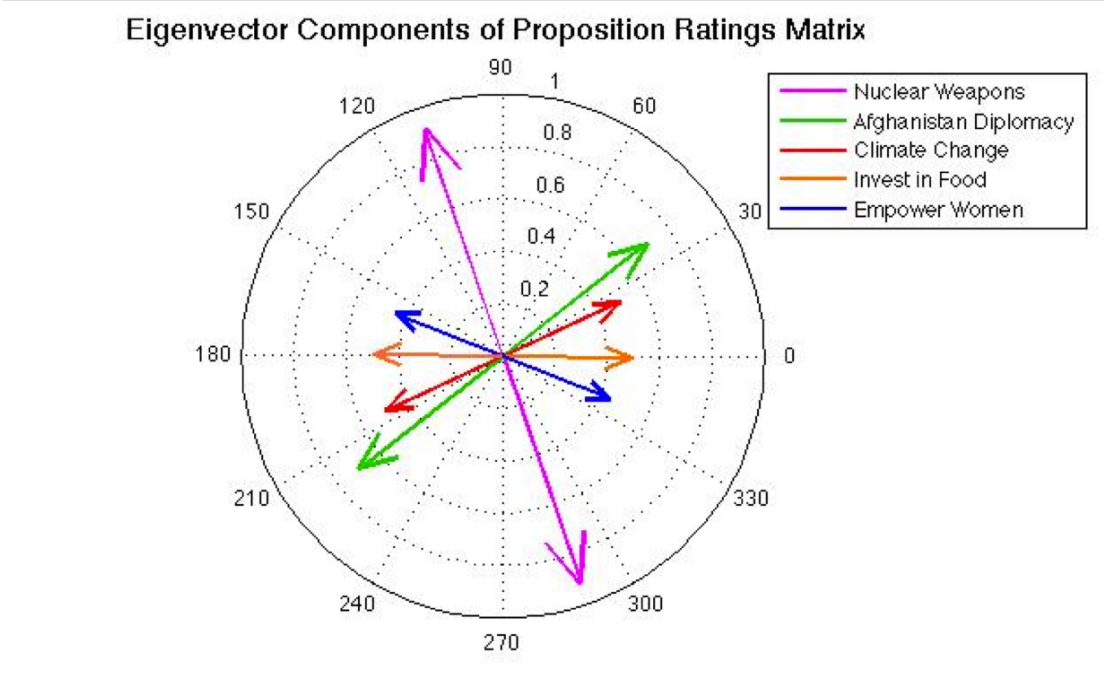


Figure 5.6: Eigenvectors defining the Opinion Space map.

5.5.3 Measuring Changes in Opinion

One of the crucial components of a social computing system is that the outcomes be deemed meaningful. Ideally, a consequence of meaningful output (especially when it comes to idea and opinion sharing) is that participants learn something by participating with the system. Although it is difficult to directly measure whether participants are learning, we assume that a by-product of learning is an evolution or change of opinion. Fortunately, this can be measured directly in Opinion Space, and back-testing has on "wild" data shown evidence of this behavior. In fact, in response to the discussion question hosted by the US Department of State on the prevention of nuclear proliferation, participants changed their opinions on the proposition about nuclear threats by an average of 30 percent. A summary of the changes in opinion for all statements is illustrated in Figure 5.8.

5.6 Future Work

Although comment lists have many faults, they are a familiar and straight-forward interface. This user study suggests that Opinion Space can be an effective alternative, but our primary challenge is reducing the barrier to entry by making the interface easier to use and more intuitive. Opinion Space is a new model; its spatial arrangement of points

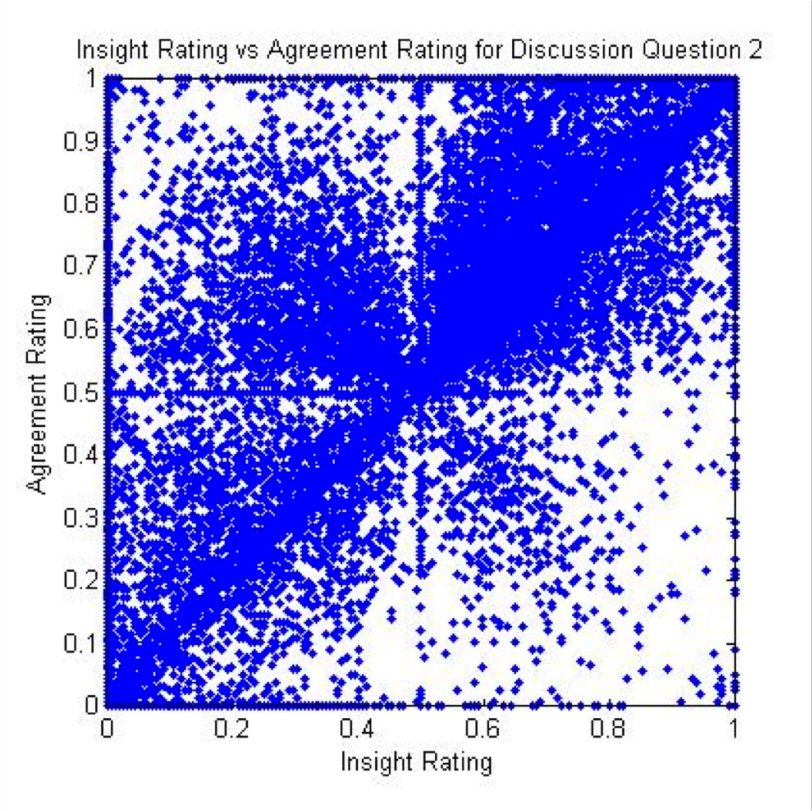


Figure 5.7: Scatter plot of insight versus agreement ratings.

may not yet be intuitive to participants who expect to see the space labeled with axes such as "liberal" and "conservative." We view this as potentially a strong advantage it conveys that the range of opinions do not fall along a single axis and that they are far more diverse. However, feedback we have received from participants suggests that they want to better understand the arrangement. One idea we are exploring is to insert "landmarks," well-known people such as Jon Stewart or Oprah Winfrey into the space, and to automatically label regions of the space by clustering the points and performing textual analysis on the responses in each cluster to extract significant keywords that can be overlaid on the space.

We are also curious whether a scoring model can introduce incentives to increase participant engagement. We posit that there are three types of participants: 1) casual participants who want to quickly find and read the most insightful responses, 2) authors who want to contribute insightful responses that gain the respect of other participants, and 3) gamers who want recognition for their role in shaping the space by rating the responses of many others. We are developing new scoring metrics that cater to these three participant personalities, with close attention to avoiding malicious participant behavior. These models are discussed at length in Chapter 6.

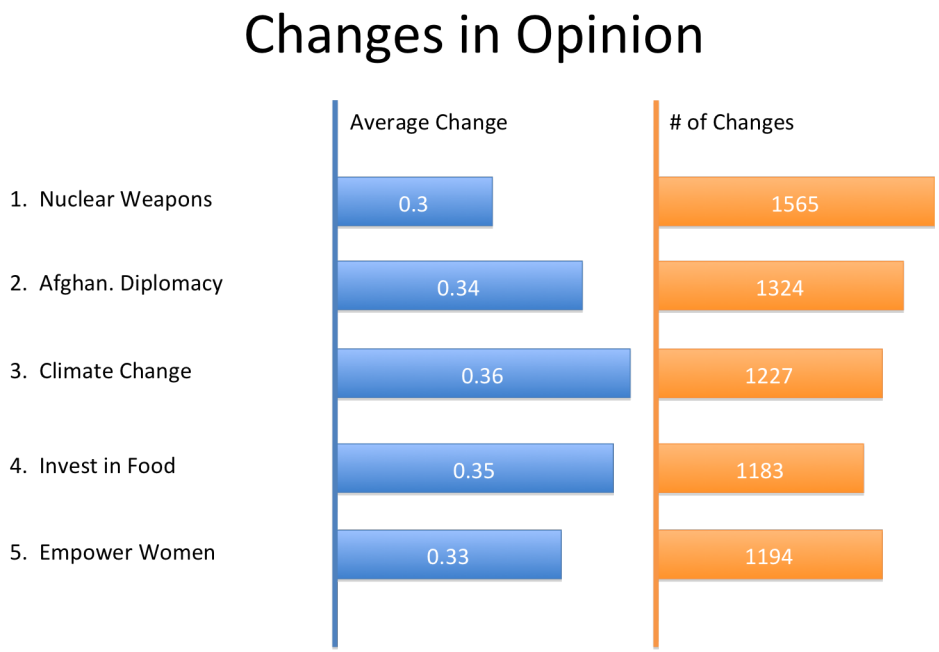


Figure 5.8: Changes in opinion over time.

The controlled user study reported here was limited to one hour per participant. To further investigate behavior over time, we would like to conduct a longitudinal user study. We also intent to experiment with various forms of incentive structures to determine which yields the greatest level and quality of participation.

We are now exploring how Opinion Space might be extended and applied to commercial websites such as Netflix, Amazon, Slashdot, and Digg. A scalable tool for managing massive online discussions requires a method for filtering participant-generated content. In future versions of Opinion Space, we will extend our work on Eigentaste (60), a PCA-based collaborative filtering algorithm that runs in constant online time, and combine it with our model for identifying insightful responses (10) to make personalized response recommendations. This is discussed in greater detail in Chapter 7.

Participant	What strategy did you use to explore the comments in the space?
1	Explored all the extreme opinions and ones very close to mine as well.
2	I chose a circle on the left side, then chose a corresponding circle on the right side. Also, I started from the periphery and came in towards the middle.
3	I only picked a few near me ... then I picked the ones farthest from me. And then I looked at the landmarks. Picked a few near big clusters
4	I first chose the ones by me. Then I chose the particularly brighter and darker points. I chose the brighter points because I assumed they would be in conflict with mine. After that, I chose the darker points for the same reason; I assumed that they would be more aligned with my views.
5	Random
6	I tried to get a good cross-section of differing opinions and views so that I would be able to view and try to understand all sides of the argument.
7	Random at first, to see what was there. Then I began looking at opinions in different areas of the space to see how those of different viewpoints thought about this particular issue.
8	I looked at the points that were nearest me and furthest from me just to see if the system was accurate.
9	I picked a few comments near where mine were so I could just see what likeminded people thought. Then I picked comments far away from mine to see what other people on the social/political/moral spectrum thought.
10	I tried to pick a variety of points on the left and right and points that were bright and dim.
11	I checked the politicians'/commentators' opinions first, then took a look at one of the points near mine, then at the farthest one I could find, and sort of hopped back and forth from there, looked at some around the large blue points, looked at some at random ...
12	I clicked on points that ranged from being very close to my position and very far.

Table 5.5: Strategies participants reported using when browsing responses in the Space interface.

5. A GEOMETRIC MODEL FOR VISUALIZING THE DIVERSITY OF ONLINE TEXTUAL RESPONSES

6

Reputation Metrics for Textual Responses

6.1 Introduction

One of the most powerful features of the Internet is its ability to collect, organize, publish, and disseminate massive amounts of user-generated content, all in a fraction of a second. For many, the Internet has become a vital part of every day life. We turn to websites such as Amazon to compare a wide range of household and entertainment products. Aggregator sites such as Digg, Slashdot, and Metafilter provide social and technology news, and all of the major news publications publish their content on the web. People stay connected on social networking sites such as Facebook, LinkedIn, and MySpace. And finally, there are several popular marketplaces for buying and selling used items and artwork.

One of the core elements that makes these websites so successful is user participation. With the free and fast flow of information between people comes a mountain of data that can be used to organize content in a meaningful and even personalized way. Many merchant websites collect and publish product reviews so that consumers can make more informed purchases, and online marketplaces aggregate and publish consumer feedback for each seller. Almost every news article, feature, or blog post has an associated comment list that allows people to anonymously comment on the content and participate in debates.

The anonymity of the Internet provides significant advantages and disadvantages. On one hand, anonymous identities allow participants to express their opinions without fear of real-life retributions. On the other hand, because people are not held accountable for their actions, this sort of environment tends to elicit extreme behavior. Many unmoderated discussion threads are subject to "trolling," where a user purposely posts an inflammatory response in order to sidetrack the conversation and spark an angry, emotional debate. Furthermore, because it is so easy to create false identities and provide false feedback, systems are vulnerable to manipulation by individuals looking to promote themselves in one way or another. (53; 82)

When it comes to choosing a seller or believing the claims of another's comment, one of the greatest challenges for participants is to determine who is reliable and who is not. It is possible to enhance this process in the form of a *reputation system.* (117) These systems take as input the history of actions for each participant and output a ranking of the participants (and their content) based on their trustworthiness; rankings are either global or personalized with respect to a particular individual. There are several challenges in creating such systems:

1. Providing participants with an incentive to cooperate and strive to improve their reputation or rank.

2. Aggregating the data to form a "fair" and nontrivial ranking of the participants and content.

3. Introducing mechanisms that make the system resistant to manipulation.

We note that participants would only want to build a good reputation for themselves if there is a promise of future gains as a result. (33; 117) For example, with one-time transactions such as on Craigslist, it is not economically advantageous to take an initial loss in order to build a good reputation. On the other hand, merchants on eBay desire better reputations because it will give consumers more confidence in making a purchase and hence more likely to do so. (119)

In the following section we give a formal definition of the problem. In Section 6.3 we give a history of related work in the area, ranging from theoretical to practical models for reputation. Section 6.4 describes the empirical data sets collected with Opinion Space and presents. We then present an analyze several original models for participant reputation and ranking the content participants generate.

6.2 Problem Setup

Our goal is to design a reputation model for participants of Opinion Space that is both "fair" and resistant to manipulation, where we define fair in an axiomatic way in Section 6.3.2. We assume that discussion forum participants are interested in promoting both themselves and those with similar viewpoints. Formally, a reputation model is defined as follows.

Definition. A reputation model is a function that takes as input the ratings provided by each participant and outputs either a cardinal or ordinal rank of every participant in the system. A reputation function is considered to be trivial if it assigns the same value to every participant. (25)

Let n be the current number of participants in the system, where each participant has a profile with some distinguishing quality; for example, a participant may be associated with a textual response, a piece of artwork, or a set of opinions. In all prior

releases of Opinion Space, a participant's profile is defined by a) her ratings of five initial statements which are used to generate the Opinion Space map, and b) her textual response to a given discussion question.

Participants are prompted to rate aspects of other participant profiles on a continuous scale; traditionally, the participant is asked to rate textual responses, but future versions may ask participants to evaluate other forms of media user-generated content such as video, photos, or even music. In the prior releases of Opinion Space, which are centered around innovation, participants evaluate a response on two scales: a) how insightful the response is, and b) to what extent they agree with the response. However, other measures may be more appropriate depending on the application, including quality and trustworthiness.

6.3 Related Work

In this section we survey the literature on existing theoretical and real-world models for social choice and reputation, including game- and graph-theoretical approaches.

6.3.1 Surveys on Formal Reputation Models

Jøsang et al. survey current work on modeling trust and reputation for online transactions in (82).

We seek to design a user reputation model that is both resistant to manipulation and satisfies certain axioms of fairness. Friedman et al. (51) survey recent results in the manipulability of reputation systems, and Altman and Tennenholtz (3) lay the foundations for studying ranking systems in an axiomatic way. In (74), Hochbaum models group ranking as a convex optimization problem that minimizes the difference between individual rankings and the final ranking; she shows that the model can be solved in polynomial time.

Our reputation model builds on ensemble learning theory by treating users as "human classifiers." Polikar details an extensive survey of the literature in (111), and Kuncheva and Whitaker studied measures of classifier diversity in (90).

6.3.2 Social Choice and Reputation

Reputation is similar to the social choice problem (5) or collective decision making in that we seek a global ranking of a set of alternatives; the main differences are that we do not assume to receive from each participant a complete ranking of the alternatives, and the participants providing the rankings are also the individuals that are being ranked. Arrow (5) defined the following set of five fairness axioms for a *social welfare function* or voting scheme, which takes as input a ranking vector R_i of all the alternatives from each participant i and outputs a single ranking.

1. **Universal Domain:** Any ranking R_i of the alternatives is acceptable as input from participant i.

2. **Monotonicity:** All else being equal, if the rank of an alternative x rises or stays the same for every participant, then in the final social ordering x should not decrease in rank.

3. **Independence of irrelevant alternatives:** Consider a subset S of the alternatives. Changes in participant rankings of alternatives outside of S (i.e. irrelevant alternatives) should not affect the final ordering of S.

4. **Surjectivity (or Citizens' Sovereignty):** Any ranking of the alternatives should be achievable by some admissible input.

5. **Non-dictatorship:** There should not exist a participant i such that the final ranking is equal to R_i. That is to say, the final ranking should not be solely determined by the ranking provided by a single participant.

Arrow's Impossibility Theorem showed that for any instance with at least two voters and three alternatives, there does not exist a ranking function that satisfies all of these axioms (5). Hence, with any voting scheme we can only hope to satisfy a subset of the axioms.

To define a set of conditions that make a reputation system "fair," we adopt the axioms from Arrow's impossibility theorem. We also require that the reputation function be deterministic and that it should return a unique ranking of the participants.

One crucial assumption that Arrow makes is that each participant can submit as input a single ranking vector or vote. However, for many reputation systems on the Internet, this assumption is no longer valid due to the ease of creating unlimited anonymous identities. Furthermore, Arrow assumes that every participant ranks the alternatives truthfully, and he does not consider the impacts of a participant who strategically gives false rankings to manipulate the outcome. This motivates the need for a reputation function that is both fair *and* resistant to manipulation.

We say that a reputation system is vulnerable to manipulation if a participant can influence her own rank by either providing false ratings, hiding her history of transactions by creating a new account (also known as *whitewashing*), or by creating fake accounts that are used to fraudulently increase give herself positive ratings (also known as a *sybil attack*).

6.3.3 Intuitive Models

We now take a step back to discuss four intuitive models for reputation that are commonly used as a way to recommend items or participants. Let r_{ij} be participant i's rating of item (or participant) j, and let x_i be the rank of participant i. Ratings are real values, either discrete or continuous, and rankings are cardinal (i.e. numerical values, as opposed to ordinal).

6.3.3.1 Mean and Median

The averaging model is probably the most intuitive and commonly used. (82) In this model, the rank assigned to participant or item i is simply the average of the ratings given to i. Let U_i be the set of participants that have rated i. Then

$$\textbf{Average: } x_i = \frac{1}{|U_i|} \sum_{j \in U_i} r_{ji} \tag{6.1}$$

Using the median rating instead of the average can be slightly more appealing, because it is less sensitive to outliers. Unfortunately, both of these methods are highly vulnerable to manipulation, even in the presence of irrelevant alternatives. (74)

6.3.3.2 In-degree

The in-degree ranking metric was designed to rank nodes in a network in order of importance; it is also known as a type of graph *centrality* metric. For any collaborative system we can construct an underlying directed graph $G = (V, A)$ as follows. The set of vertices V is equal to the set of participants. A directed arc from participant i to participant j is assumed to be a positive vote for j from i. The in-degree of a node j is the total number of incoming arcs $(i, j) \in A$ to j. (37) This value is typically normalized to a value between 0 and 1 by dividing by the maximum possible number of incoming arcs. Formally, the rank of participant i according to the in-degree metric is

$$\textbf{In-degree: } x_i = \frac{|\{(j, i) \in A\}|}{n - 1} \tag{6.2}$$

In Opinion Space, the in-degree method is a more naive approach that ranks participants in order of the number of ratings received, so the participant with the most ratings has the highest rank. For discussion threads, the in-degree model can be thought of as ranking discussion topics or responses according to how active or even controversial they are.

6.3.3.3 Weighted In-degree

In the context of the Internet, the in-degree metric says that the more websites that link to a particular site k, the more "important" it is; in this case, ratings are determined implicitly rather than explicitly, and it is not possible to give a site a negative vote or rating. For other systems, it may be more valuable to explicitly collect higher-precision ratings, such as on a discrete or continuous scale. Doing so makes it possible to collect negative rating information as well, and the in-degree metric must be adapted appropriately. We define in-degree in the weighted sense as follows. Let participant ratings be a value in the range [-1,1]. Then the rank of participant i according to the

	Mean	Median	In-degree	Weighted In-degree
Mean	1	0.9806	0.0404	0.2488
Median		1	0.0418	0.2819
In-degree			1	0.4692
Weighted In-degree				1

Table 6.1: Pairwise correlation between rankings produced by the mean, median, in-degree, and weighted in-degree models. Data is from Opinion Space 2.0, question 2.

weighted in-degree model is

$$\textbf{Weighted In-degree: } x_i = \frac{\sum_{i:(j,i)\in A} r_{ji}}{n-1} \tag{6.3}$$

6.3.3.4 Comparing the Four Models

Table 6.1 gives the pairwise Pearson correlation between the rankings produced by the mean, median, in-degree, and weighted in-degree models described above. The ratings data were adjusted to fall within the range [-1,1]. We found that the mean and median are extremely highly correlated, and that (mean, median) and in-degree have almost no correlation. However, the (mean, median) models show some positive correlation with the weighted in-degree model, and in-degree correlated quite highly with its weighted version.

6.3.4 Collaborative Filtering

Collaborative filtering is a family of techniques for recommending items based on ratings data provided by the users of the system. The primary assumption is that users with similar rating patterns will provide similar ratings in the future. Hence, by finding clusters of similar users we can make predictions on how a user will rate an item; this information can then be used to make item recommendations. Similarity can be defined in a variety of ways, and we refer the reader to Chapter 4 for a more in-depth discussion.

Collaborative filtering is related to reputation systems in the sense that both collect and aggregate ratings from participants, and both are vulnerable to manipulation. With reputation systems, we are typically more interested in finding a single global ranking of the participants according to some definition of their trustworthiness, whereas in collaborative filtering we seek a personalized ranking of the item set tailored to the tastes of a specific user. (82) However, both types of systems must solve the problem of aggregating the ratings provided by a (sub)set of users to form a single ranking.

Eigentaste as a Reputation System. While Eigentaste was originally designed to be a collaborative recommender system for items, it can be easily adapted for use as a reputation system where interactions between participants are free (i.e. exchange of

ideas or written content). In this case, the gauge set will consist of the participants with the highest variance of ratings, and participants are recommended other participants instead of items. A global ranking of the participants can be found by placing them all in the same cluster. An Eigentaste reputation system might be particularly appropriate for recommending networking contacts on social networks, or coalition building for political activists, since interactions in these applications do not require the exchange of money.

6.3.5 Group Rankings with Network Flows

Hochbaum and Levin studied the generalized problem of group ranking. In (74) they present a formal optimization model and efficient algorithm for rank aggregation and show how it addresses the shortcomings of many other ranking systems, including dependence of irrelevant alternatives. In this section we review their results.

In the group ranking problem, the goal is to rank a set of alternatives that best matches the individual preferences of the users. It is assumed that the number of alternatives is significantly large and that most participants will be unable to rank all of the alternatives. Hence, the problem takes as input an incomplete set of rankings from each participant, and the output is in the form of a complete ranking of the alternatives.

Hochbaum and Levin define a ranking to be a pairwise comparison between two alternatives; intensity or cardinal rankings specify the numerical degree of preference for one alternative over another, and preference or ordinal rankings only specify the order of preference. For our purposes, we only discuss results for cardinal rankings where each participant submits feedback in the form of pairwise intensity rankings.

If participant u rates alternative i with r_i^u and alternative j with r_j^u, then the intensity of u's preference for i over j can be defined as either $r_{i,j}^u = r_i^u - r_j^u$ in the additive sense or $r_{i,j}^u = \frac{r_i^u}{r_j^u}$ in the multiplicative sense. Hence, we say that participant u prefers i to j if $r_{i,j}^u > 0$ or $r_{i,j}^u > 1$ respectively. A set of comparison ratings is said to be *consistent* if for each i, j, k we have that $r_{ik}^u = r_{ij}^u + r_{jk}^u$ (additive) or $r_{ik}^u = r_{ij}^u \cdot r_{jk}^u$ (multiplicative). For simplicity, we discuss the model for additive intensity rankings.

The ratings provided by a participant u can be represented by a directed graph $G(V, A_u)$, where the set of vertices V is equal to the set of items I. A directed arc (i, j) is added with weight $r_{i,j}^u$ to A_u if u prefers item i to item j.

Hochbaum and Levin prove in (74) that a necessary condition for consistency is that the graph be acyclic. If a participant's ratings are consistent, then a topological sort of the vertices will yield a consistent preference ranking of the items. They also prove that if the ratings are consistent, then all paths between any two nodes will be of equal length. Hence, given an incomplete but consistent matrix, we can construct the *consistent closure* of the user's ratings by setting r_{ij}^u to be the length of a path from i to j, should one exist. Otherwise, i and j are said to be *incomparable*.

Since the ranking graph of every user has the same set of vertices V, we can form a super graph of all user rankings, where the vertices are unchanged and the set of arcs

is defined by $A = A_1 \cup A_2 \cup \cdots \cup A_n$. In any practical scenario, this super graph is most likely to be inconsistent, which leads to a very intuitive definition of optimality for an aggregate ranking: the objective is to find a *consistent* group ranking that is "close" to the original, inconsistent ranking matrix.

Hochbaum and Levin give the following Close Rankings (CR) optimization model to formalize the problem. Let z_{ij} be a decision variable indicating the preference intensity between alternatives i and j in the final group ranking, and let x_i be the weight variable for alternative i. If $F_{ij}()$ are general convex functions, then (CR) is defined as:

$$\begin{aligned} \text{(CR)} \ \min \quad & \sum_{i<j} F_{ij}(z_{ij}) \\ \text{subject to} \quad & x_i - x_j = z_{ij} && \text{for } i < j \\ & -n \leq x_j \leq n && j = 1, \ldots, n \\ & -n \leq z_{ij} \leq n && \text{integer, } \forall i, j \end{aligned}$$

The functions $F_{ij}()$ are meant to measure the deviation of the individual rankings from the solution. Let w_i^u be the rank or weight given to alternative i by participant u. Then the function could be defined by the absolute deviation: $F_{ij}(z_{ij}) = \sum_{u \in U} |w_i^u - w_j^u - z_{ij}|$. Several other formulations have been proposed, including quadratic .

Hochbaum and Levin give an efficient algorithm for solving (CR) by showing that it is a special case of the convex dual of minimum-cost network flow, which can be solved in polynomial-time using the the method in Ahuja et al. (1)

6.3.6 The PageRank Algorithm

Google's PageRank algorithm (17) has revolutionized search on the Internet. The algorithm analyzes the link structure of the web to rank webpages according to citation importance, which can also be thought of as reputation since hyperlinks are for the most part public information. (82) Brin and Page assume that a link from page A to page B is a positive vote from A for B. Under this assumption, the reputation of a webpage B is a function of the number of pages linking to B as well as their reputations. Formally, the PageRank model is given as follows, and can be solved via an iterative process.

Let $\{T_1, T_2, \ldots, T_N\}$ be the set of pages pointing to page A, let $C(T_i)$ be the out-degree of page T_i, and let d be a damping factor. Then Brin and Page define the PageRank (17) of page A as

$$PR(A) = d + (1 - d) \sum_{i=1}^{N} \frac{PR(T_i)}{C(T_i)} \tag{6.4}$$

Although originally intended for ranking webpages, the PageRank algorithm can be adapted for use in any system that has an analogous graph structure. In the case of Opinion Space, participants are analogous to webpages, and (under the PageRank model assumption) a positive rating by participant i for participant j is analogous

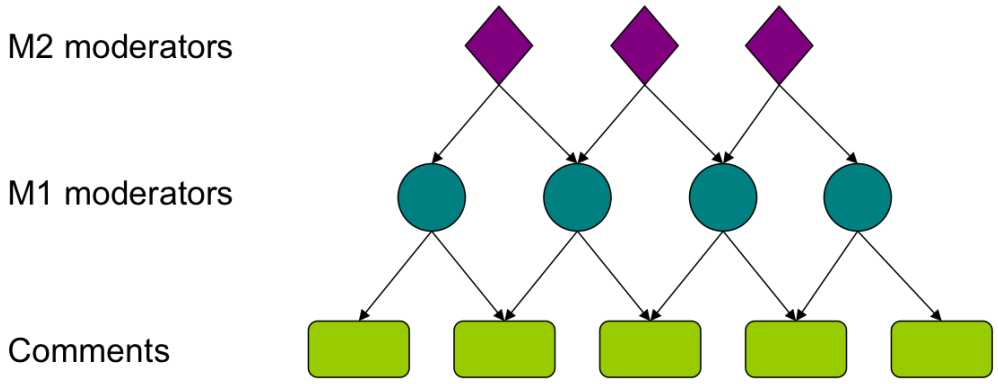


Figure 6.1: Illustration of hierarchy of moderators on Slashdot. A new set of M1 moderators is selected from all active participants approximately every 30 minutes; these moderators are responsible for assessing and classifying comments and are given a fixed number of comments they can moderate. M2 moderators are chosen from participants with the most long-standing accounts and are responsible for assessing the performance of M1 moderators.

to webpage i pointing to webpage j. Since PageRank is not designed to incorporate negative feedback from participants, it is nontrivial to adapt for use in Opinion Space. See Section 6.6 for a more in-depth discussion.

6.3.6.1 Slashdot

Slashdot (`http://slashdot.org`) is a technology news aggregator and discussion website, where participants can submit links to and comment on news articles. Out of the set of links submitted, a handful are selectively chosen by the Slashdot staff for display on the home page. Participants have the option to browse, comment on, and submit stories anonymously or logged in with a free account. Accountholders have the ability to save their preferences, indicate which participants are "Friends" or "Foes," and their posted comments have higher visibility than ones that are posted anonymously.

Initially, Slashdot was a small site and did not require any moderation. But as the site began to grow, the signal to noise ratio diminished. According to Slashdot's Frequently Asked Questions page, the site gets thousands of comments a day, and tens of thousands a month. With these kinds of numbers, information overload is a real problem. To help participants sort through the comments to find the ones of highest "quality," the site has developed a moderation and meta-moderation method which they combine with a reputation system they call "Karma."

Comment Scores Every comment has an integer score between -1 and 5. A comment submitted by an anonymous participant has an initial score of 0, and the default score of a comment submitted by a logged in participant is 1, though it can range between 0 and 2 depending on her Karma (see below).

Moderation About every 30 minutes, a group of currently logged in participants is automatically selected to become level 1 or M1 moderators. M1 moderators are given a fixed number of "points of influence," and it costs one point to moderate a comment. An M1 moderator can moderate a comment by choosing an adjective from a list that is meant to classify the comment (e.g. flamebait, funny, or informative). If the moderator selects a negative adjective, then the score of the comment is decreased by one; similarly, a positive adjective will increase the score by one. To prevent conflicts of interest, an M1 moderator cannot comment in a discussion that she has moderated.

Meta-Moderation When Slashdot first introduced the M1 mass moderation system, they found that a significant number of participants were abusing their privileges. To cope with this problem, they introduced M2 or *meta*-moderation. Only long-standing account holders that have accounts older than 7.5 percent of Slashdot participants are eligible to become M2 moderators; new participants must therefore wait several months before they are eligible. Anyone who is eligible can meta-moderate up to several times a day. Meta-moderation works by asking volunteer M2 moderators to rate 10 randomly selected M1 moderations as either fair, unfair, or neither. Depending on whether the meta-moderations of a participant are compatible with those of other participants, meta-moderating can improve or reduce the karma or reputation of a participant, as defined below.

Karma Every logged in Slashdot participant has an associated Karma, which mainly reflects how the participant has previously been moderated. Karma is measured on the following discrete scale: Terrible, Bad, Neutral, Positive, Good, and Excellent. Comments that are moderated up will improve the Karma of the poster, and comments that are moderated down will decrease the Karma of the poster. A participant can also improve her Karma by submitting a story link that is posted to the site and by doing a good job at meta-moderation.

The Slashdot model maintains scores for comments and karma levels for participants. The comment scoring function is given formally as follows. Let x_i be the current score for comment i. If M1 moderator j gives a positive vote for i, then $x_i \leftarrow \min\{x_i + 1, 5\}$; similarly, if j gives i a negative vote, then $x_i \leftarrow \max\{x_i - 1, -1\}$. To the best of our knowledge, Slashdot has not published the specifics of the algorithm they use to compute Karma, though according to their FAQ the function is monotonic in that positive moderations for the comment of a participant will only serve to increase her karma and negative moderations can only decrease the karma of the participant.

One of the drawbacks of the Slashdot reputation and moderation system is that it is difficult for new participants to understand due to its complexity. It can also take several months before users can participate in moderating comments, which is a significant barrier to entry. Friedman et al. (51) argue that meta-moderation is inefficient, since those users could otherwise spend time evaluating actual ratings instead of the evaluations of other participants.

On the other hand, the systems is both scalable and functional. Lampe and Resnick

OS Version	Question #	# Responses	# Ratings	Rating Type
1.0	3	1,601	13,111	Agree
1.0	4	549	3,866	Agree
2.0	2	2,147	20,789	Insight
2.0	3	1,123	5,472	Insight
Auto Industry	1	1,148	94,535	Insight

Table 6.2: Description of the different data sets compiled with Opinion Space as of 20 January 2011. The number of textual responses contributed, the number of response ratings, and the format (agree or insight) of those ratings are provided.

OS Version	Question	Avg # Rated	Std Dev	% Rated
2.0	2	13	33	0.0116
Auto Industry	1	95	178	0.0828

Table 6.3: The average and standard deviation of the number of responses rated by contributors to the two largest data sets compiled with Opinion Space: question 2 of version 2.0 and the Automotive Industry study. The last (right) column gives the percent (on average) of responses rated by participants; this is a measure of the density of each data set.

present an empirical study of participant behavior on Slashdot in (91). They found that more than three times as many participants used comment ratings to navigate discussion threads as those who did not, implying that participants find the reputation and moderation system to be significantly helpful. In a survey of 8,121 registered Slashdot participants, 84.7 percent felt that "the moderation system is important in identifying good comments," while only 8.5 percent disagreed.

6.4 Empirical Data Collected with Opinion Space

Before presenting the various response ranking models we have developed, we describe the nuances of the data with which we are working.

6.4.1 Data Sets

Over the two years since the launch of the first version of Opinion Space, we have accumulated data sets for five different discussion questions in three different implementations of the system. See Table 6.2 for a breakdown of the data, which we now describe.

Opinion Space 1.0 focused on domestic United States politics, Opinion Space 2.0 was hosted by the US Department of State and focused on foreign policy, and the Automotove Industry study asked participants to reflect on the future of US auto manufacturers. While versions 1.0 and 2.0 relied on participants to be entirely self-motivated,

participants of the Automotive Industry study were given monetary incentives: 100 dollar gift cards were promised to the participants that contributed the most insightful comments. Furthermore, the identities of participants of version 1.0 and 2.0 were anonymous, whereas the identities of the participants of the Automotive Industry study were not. As evident in Table 6.3, this resulted in significant differences in user behavior. Namely, on average participants of the Automotive Industry study rated seven times as many responses as those in our most active instance, version 2.0 question 2. Interestingly, both data sets exhibited similar mean response ratings: 0.5155 and 0.5163 for OS 2.0 and Auto Industry, respectively. However, the standard deviation of the ratings for OS 2.0 was 20 percent higher at 0.3094 versus 0.2566.

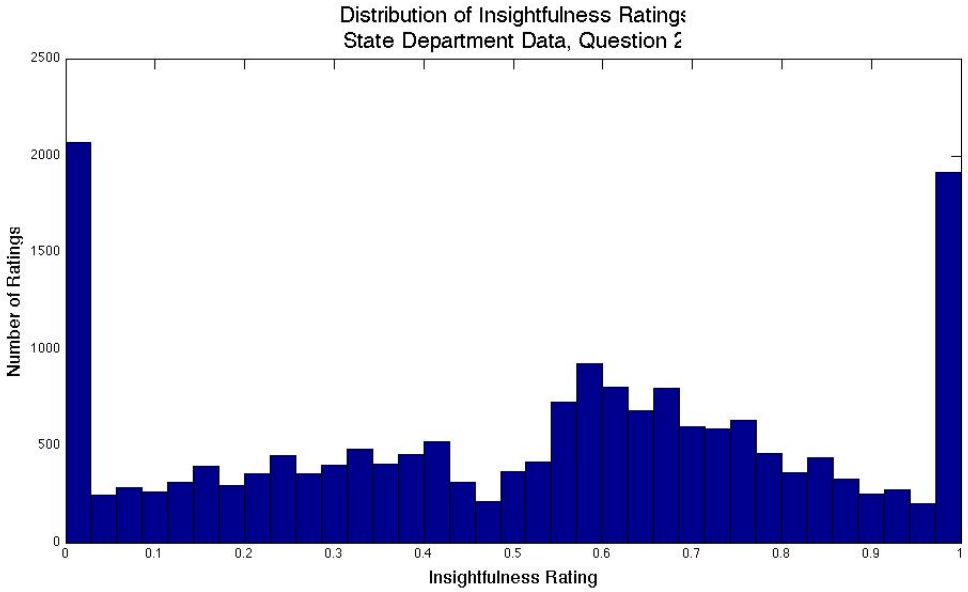
6.4.2 Response Ratings

Figure 6.3 illustrates the distribution of response ratings collected with Opinion Space on both the Department of State study (discussion question 2) and the Automotive Industry study. It is important to note that the data in Figure 6.3a is not well-described with a Normal distribution. Rather, it appears to be a mixture of two Normals, one with a mean around 0.3 and the other with a mean around 0.65 (assuming a rating scale ranging between 0 and 1). There are also two large spikes, one for insightfulness ratings valued at 0 and the other for ratings valued at 1. The Automotive Industry data (Figure 6.3b), on the other hand, appears to be more Normally distributed plus the two large spikes on the extremities of the rating scale.

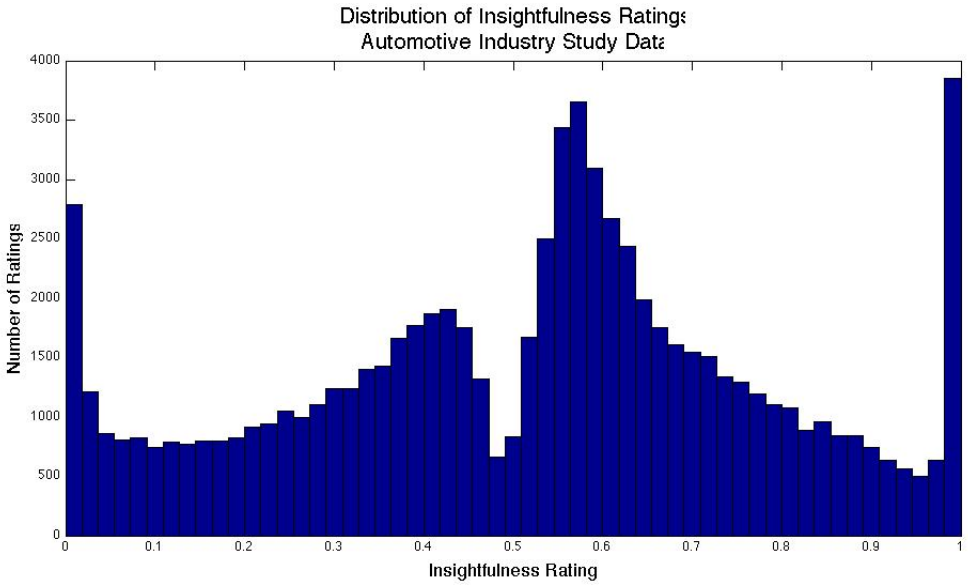
When rating responses in the earliest versions of Opinion Space (v1.0), participants were only asked to indicate their level of agreement with the response. As we developed version 2.0 of the system for the US Department of State, we came to the realization that while agreement is a good measure of opinion, it is difficult to distinguish highly insightful responses from agreeable ones given only agreement data. Hence, we decided to objectively separate the insightful signal from the agreement signal by asking each participant to rate responses according to a) how much they agree with a response, and b) how insightful it is. In our subsequent analysis and models for ranking responses, we treat the agreement ratings as noise and choose to only consider insightfulness ratings.

Figure 6.4 is a scatter plot of insightful versus agreement ratings collected with the Automotive Industry study. The Pearson correlation between these two signals is 0.7469 for the OS 2.0 question 2 data set and 0.729 for the Automotive Industry data set. While this level of correlation is significantly high (as can be expected), there is also a clear amount of separation between the two signals (as illustrated in Figure 6.4). This indicates that participants are in fact using the two rating scales differently and as a way to separate (un)insightful comments from (dis)agreeable responses.

By asking participants to rate responses along these two axes, we are able to filter out the responses in the upper left quadrant, which are responses with which others tend to agree but are not insightful. For example, let us consider the responses in the Automotive Industry data set that received at least 50 (insight, agreement) rating pairs. The response with the most negative average difference between insight and agreement ratings is “Make cars much more fuel effenent” [sic]. As most people would

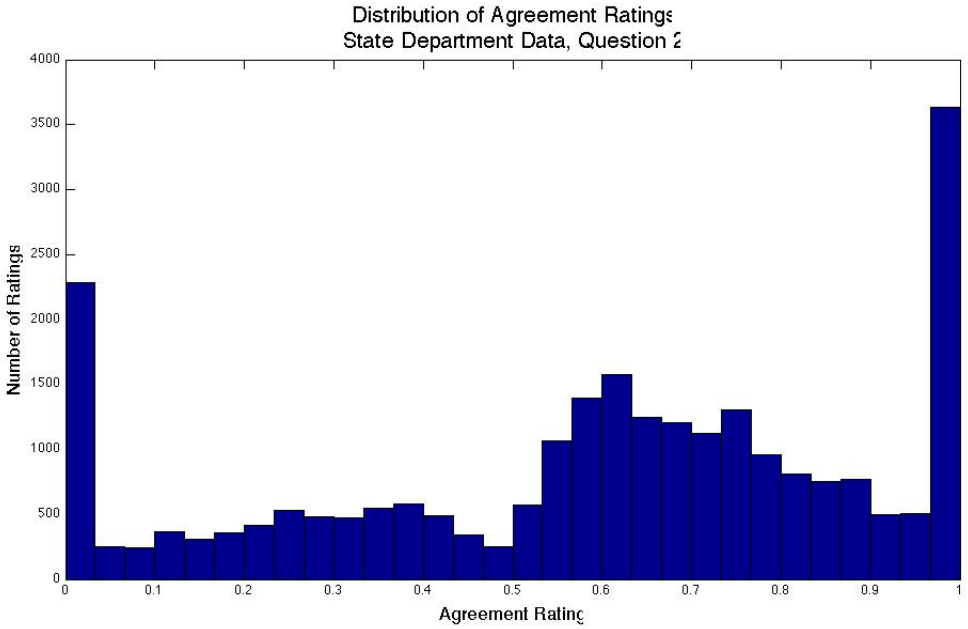


(a) State Department Data

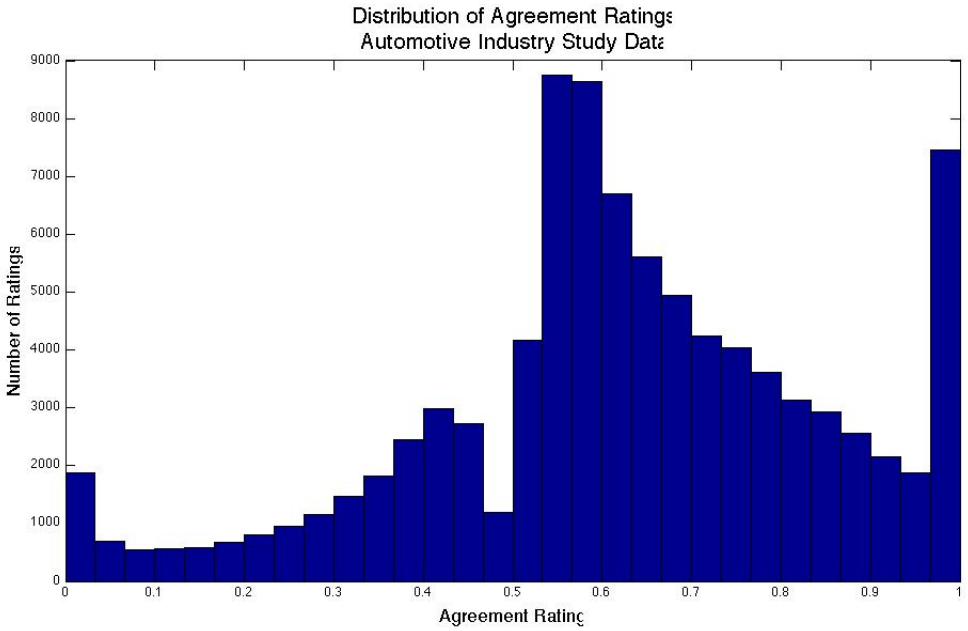


(b) Automotive Industry Data

Figure 6.2: Distribution of insightfulness ratings collected for the (a) Department of State and (b) Automotive Industry Opinion Space studies.



(a) State Department Data



(b) Automotive Industry Data

Figure 6.3: Distribution of agreement ratings collected for the (a) Department of State and (b) Automotive Industry Opinion Space studies.

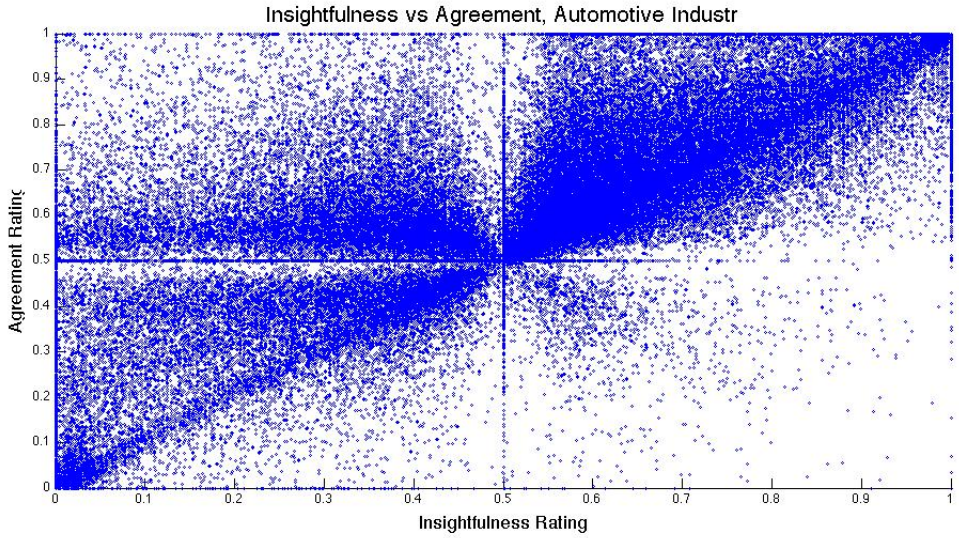


Figure 6.4: Insight versus Agreement response ratings for the Automotive Industry study. The Pearson correlation between these two signals is 0.729. We are most interested in filtering out the points in the upper-left quadrant (i.e. responses with high agreement and low insightfulness) and investigating the points in the lower-right quadrant (i.e. responses with high insightfulness and low agreement).

prefer vehicles that are more fuel efficient, this is clearly a response that is difficult for most people to disagree with; however, it fails to contribute anything new or insightful to the conversation.

On the other hand, the response with the most positive average difference between insight and agreement ratings (lower right quadrant of Figure 6.4) is the following:

> *The materials the auto manufacturers use should be stonger and lighter. Materials such as plastics, carbon fiber and alluminium should be used. Engines should be smaller yet more powerful and generate higher fuel efficiency. The electrical system of the automobile should drive the air conditioner and all power assists such as steering and brakes. The twelve volt battery system must be changed to a 43 or 44 volt system. The engine should be just for performance. All automobile engines should shut off when stopped and restart when the accerator pedal is pressed. The energy of all braking should be recycled into the electrical system. Manufacturers should offer engine displacement on demand for all their engines. A six clinder engine could run on three cylinders on the highway. Traction control, stability control, rear camera and avoidance control should be standard on all automobiles. Zenon lighting and LEDs should be standard and headlights must follow from side to side.* [sic]

This response proposes several different ideas that push the boundaries of car technol-

ogy, and so it is not surprising that participants found it insightful but were reluctant to agree with it. This is a great example of the type of response we are seeking: one with which participants are not completely comfortable yet find compelling or insightful in some way.

Interestingly, the difference in average insight and average agreement ratings is skewed significantly to the left. Specifically, for the most agreeable but least insightful response above, the difference is -0.246, and the difference for the most insightful but least agreeable response is 0.0235. Figure 6.5a illustrates the distribution of the difference between average and insightfulness ratings for individual rating pairs of the Automotive Industry study, and Figure 6.5b describes the average agree rating minus average insight rating across all responses. As evident from these histograms, the majority of responses were to be found less insightful than agreeable. While the peak of the distribution for individual (agree, insight) rating pairs is centered around 0, it is skewed to the left of 0 when averaged for each response. For both histograms, the rate of descent to the left of the peak is much steadier than the rate of descent to the right, indicating that it is more difficult for participants to admit that they find a response insightful when they disagree with it; this is precisely why we are interested in the responses that *do* elicit such feedback.

6.4.3 Relationship Between Position and Ratings

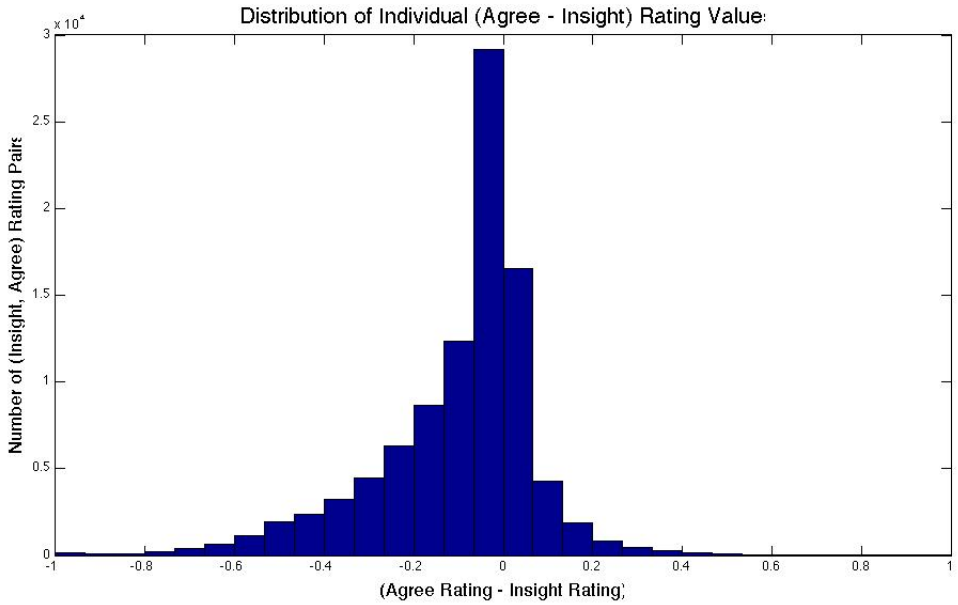
We are also interested in the spatial relationship between the position of a participant in the Opinion Space map and how she rates the responses of others. Theoretically, the extent to which such a relationship exists is dependent on three key factors:

1. The connection between the discussion question and the initial statements used to generate the map. For the distances between participants in the map to be meaningful in terms of their opinions on the topic of discussion, the statements must serve as predictive or dependent factors; that is, a participant's ratings of the statements must be somewhat correlated with the participant's opinion on the discussion topic.

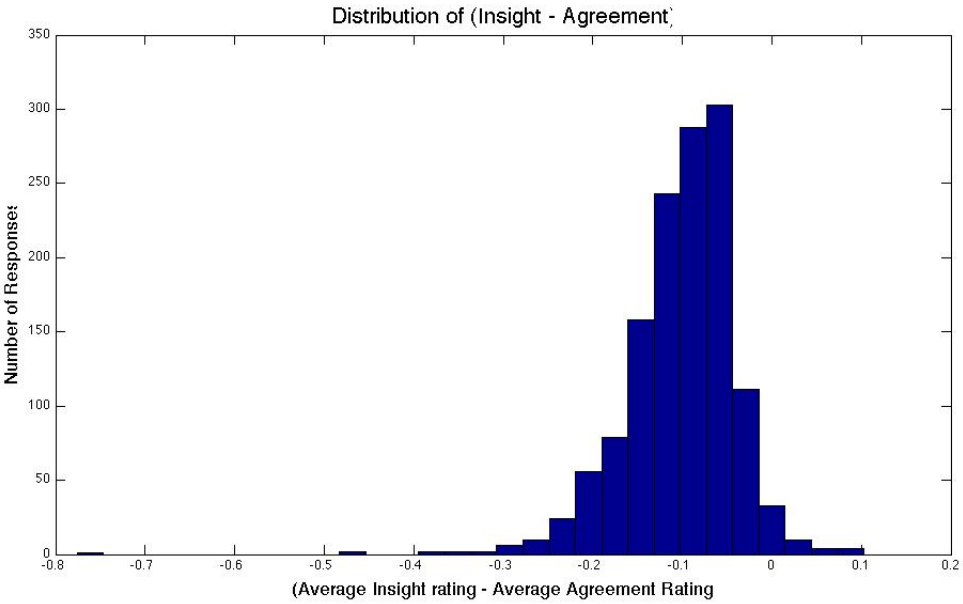
2. The style or phrasing of the discussion question. Specifically, we consider two types of discussion questions that can be asked: a) soliciting an opinion on a controversial topic, or b) soliciting innovative ideas to solve a problem.

3. The statements used build the Opinion Space map. If the statements are all opinion-based in nature, then we suspect that with an open-ended question of type (b), we are less likely to see a dependence between position in the space and response ratings; this is because idea-sharing and innovation has less to do with opinion and more to do with creativity. On the other hand, if some of the statements are demographic-based, then we may expect to see a higher correlation between position in the space and response ratings.

Table 6.4 gives the Pearson correlation between response ratings and the distance between the author and reviewer for our two primary data sets. The distance between



(a) Individual Rating Pairs



(b) Average for Each Response

Figure 6.5: Histograms depicting a) the distribution of (agree - insight) rating values for individual (agree, insight) rating pairs, and b) the distribution of the average agree rating minus the average insight rating for each response.

OS Version	Question #	Insight (2D, 5D)	# Agree (2D, 5D)
2.0	2	-0.1568, -0.1588	-0.2048, -0.2176
Auto Industry	1	-0.0310, -0.0114	-0.0611, 0.0405

Table 6.4: Pearson correlation between response ratings and the distance between the author and reviewer, where distance is computed in both 2- and 5-dimensional space.

two participants is computed in both 2- and 5-dimensional space so as to get a full view of the situation. All correlations reported are statistically significant with p-values less than 0.01. With Opinion Space 2.0 question 2 (State Department), we find that response ratings (both insight and agreement) show significant negative correlation with the distance between the author and reviewer. This implies that participants are more likely to rate a response positively if the author of the response is closer to the participant in the space. Furthermore, the effect is greater with agreement ratings as opposed to insightfulness ratings, which supports our above hypothesis that innovation or insight has less to do with opinion (i.e. position in the space).

This is even more evident when looking at the Automotive Industry, where the correlations are much closer to 0. The primary purpose of this study was to crowd-source innovative ideas and is less opinion-based than the foreign policy-themed State Department version.

As participant behavior on the site is so dependent on the nuances of the statements and discussion question, we may wish to employ different models for filtering and ranking responses based on the behavior we observe or the structure of the particular instance.

6.4.4 Mutual (Dis)Agreement Between Participants

Another interesting measure to consider is the extent to which participants' views align with each other. For example, if participant A agrees with participant B, how likely is participant B to agree with participant A? Looking at the Department of State data set, question 2, we found that about 70 percent of the time either: participant A agreed with participant B and B agreed with A, or A disagreed with B and B disagreed with A, where agreement is defined as a rating of 0.5 or higher on the agreement rating scale.

6.4.5 Summary of Empirical Data

In exploring the data collected with various instantiations of Opinion Space, we made several surprising and key observations. First, the response ratings collected from participants are not normally distributed, which indicates that using the sample mean as a measure of response quality may be problematic. Second, by separating the agreement and insightfulness signals for response ratings, we seem to be able to reduce noise and filter out responses that are agreeable but not insightful. And finally, under certain conditions, participants exhibit a rating bias towards those who are closer to them-

selves in the space. That is, participants are more likely to give a positive rating to those who share a similar baseline opinion on the initial statements.

These three observations motivate the design of more sophisticated metrics for modeling participant reputation in Opinion Space. In the remainder of this chapter we propose three such metrics and compare their performance on empirical data.

6.5 Reputation Metric I: A Spatial Approach for Finding Insightful Responses

To help participants better manage the large amount of information available in Opinion Space, we have developed a mathematical model for identifying the most insightful responses as a function of both participant-provided response ratings as well as their position in the map.

The model is motivated mathematically and described in (10), and we summarize it here. It operates under an assumption borrowed from recommender systems theory that like-minded participants are more likely to agree than those who differ in opinion:

> **Recommender System Hypothesis:** *A participant is more likely to rate highly (find insightful) the response of another participant who shares a similar baseline opinion and more likely to rate poorly (find uninsightful) the response of another participant who has a different baseline opinion.*

The model is designed to adjust for this bias by weighing the response ratings collected from each participant. Hence, a rating is given greater influence in the overall rank of a comment when it indicates consensus among diverse participants (or conversely, when it indicates disagreement among similar participants). The first phase of the model is to transform the comment ratings to reflect the degree of positive or negative influence each rating should have, and the second phase involves aggregating the comment ratings to form a global ranking of the comments.

Let x_i be the numerical vector of agreement ratings provided by participant i on the initial five statements used to build the Opinion Space map; this is referred to as the opinion profile of participant i. We denote r_{ij} as the numerical insightfulness rating participant i gave to the response posted by participant j; this number is limited to the continuous range between -1 and 1, where larger-valued ratings indicate higher degrees of insightfulness.

We measure the similarity between any two participants according to the Euclidean distance of their opinion profiles in five dimensions. That is to say, lower distances correspond to greater similarity. Symbolically, let $d_{ij} = ||x_i - x_j||$ be the Euclidean distance between the five-dimensional opinion profiles of user i and user j. Let $d_{max} = \sqrt{20}$ be the greatest possible Euclidean distance between any two participants.

When participant i rates the response by participant j with a value of r_{ij}, the

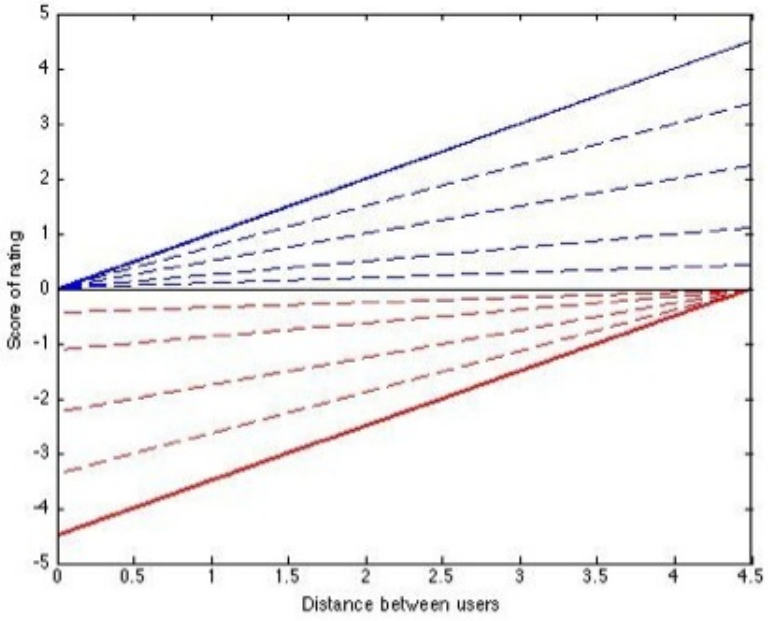


Figure 6.6: Plot of the transformed rating for a series of different ratings, as a function of the distance between participants i and j. The upper half of the plot shows the transformed rating when the original, raw rating is positive, and the lower half shows the transformed rating when the original rating is negative. Each line corresponds to a different original (raw) rating, as reflected by the slope of the line.

transformed rating is computed as follows.

$$r'_{ij} = \begin{cases} r_{ij} d_{ij} & \text{if } r_{ij} \geq 0, \\ |r_{ij}| \, (d_{ij} - d_{max}) & \text{otherwise} \end{cases} \tag{6.5}$$

Intuitively, when participant i gives the response submitted by participant j a positive rating, then the rating is weighted higher the further i is from j in the space. Conversely, when i gives j's response a negative rating, it is given greater weight the closer i is to j in the space. Figure 6.6 visually describes the behavior of this transformation, where each line reflects the transformed rating for a different original rating as a function of the distance between the participants. This linear model can easily be generalized to an exponential setting, the effect of which would be to grant greater (or less) influence to ratings that indicate consensus among diverse participants.

Our next step is to aggregate the transformed response ratings to form a global ranking of the responses in the space. Since our model transforms the raw response ratings according to how much influence they should have over the global rank of their corresponding responses, we require any rating aggregation function to be *monotonic*. Specifically, a positive rating should never decrease the rank of a response, and a

negative rating should never improve the rank of a response; at worst, a rating should leave the rank of a response unchanged.

Observe that our requirement for monotonicity does not hold if we let the rank of a response be the average of the transformed ratings for that comment. For example, suppose 10 participants gave response k an average (weighted) rating of 0.5 and participant i comes along and rates it 0.7. Now suppose that participant i is very close in the space to the author of response k, and so her weight-adjusted rating becomes 0.3. Then the new average or score of response k is $0.48 < 0.5$.

We therefore choose to rank the responses according to their weighted-indegree, defined as the normalized sum of the transformed ratings. Specifically, the score C_j of the response submitted by participant j is determined by

$$C_j = \frac{1}{c_{max}} \sum_i r'_{ij} \tag{6.6}$$

where $c_{max} = \max_j |\sum_i r'_{ij}|$ is the greatest magnitude sum of transformed ratings for a single response.

It is important to distinguish this aggregation method from simply taking the average, which tends to yield much worse results. This is because our model re-scales the response ratings according to how much influence they should have over the final ranking of the response. Intuitively, if a participant rates his neighbor really highly, then this information is less interesting or valuable (according to the Recommender System Assumption) and the rating is therefore given less overall impact. Using the normalized sum aggregation method, a positive response rating will always serve to improve the rank (or score) of a response, but to varying degrees of impact. If, on the other hand, we were to take the average of these transformed ratings to find a global ranking, then the rating from the above example could actually serve to worsen the rank of the response. That is to say, the averaging method is non-monotonic, which is an undesirable property. Subjective analysis of the two rating aggregation methods showed a clear benefit in response quality for the normalized sum method as compared to the averaging method.

6.5.1 Accounting for Confidence

When only a few people have rated a particular response, it doesn't take much to unfairly promote or demote its rank. For this reason, we have augmented the reputation model to account for our confidence in the ratings received for a response. Our approach is to compute the *standard error* of the ratings received for each response and wait until it falls below a predetermined threshold before assigning the response a rank. The standard error is defined as the standard deviation σ of the ratings for a response divided by the total number of ratings for that response, n:

$$SE = \frac{\sigma}{n} \tag{6.7}$$

Intuitively, larger values for standard deviation reflect greater variability in the ratings collected. As more participants contribute ratings, we can be more confident that the ratings represent a fair sample. In future work, it may be more valuable to design a confidence metric that not only considers the number of ratings and the variability of those ratings, but the diversity of the participants who contribute those ratings.

6.6 Reputation Metric II: Considering Reviewer Quality

We next consider an extension of the spatial ranking model that incorporates patterns of reviewer behavior. In this approach, we require a model that analyzes the response rating data across all participants to determine the ability of each participant to give reliable response ratings. Essentially, we seek a way to model the reputation of each participant as a response rater in Opinion Space. The information can then be used to weight the response ratings provided by a participant according to her reputation.

This concept is similar to that of the PageRank algorithm, where the reputation or rank of a web page is a function of the ranks of the webpages voting for or pointing to it. In our model, we assume that the quality of a participant's response is not necessarily correlated with that participant's ability to assess the responses of others. Hence, we only wish the Author score to be functionally dependent on the Reviewer score, and not the other way around.

The three key components of our generalized model are:

1. An Author scoring model, which is a function that models the quality (or insightfulness) of a response. It can take as input the raw response ratings and the distances between the author and the reviewers in the Opinion Space projection.

2. A Reviewer scoring model, which models a participant's reputation as a reviewer, and can take as input some combination of the participant's rating (or quality assessment) of the response, the mean participant rating for that response, and possibly the standard deviation.

3. A Combined scoring model, which weights each rating of a response by the corresponding Reviewer score of the participant who provided that rating.

Our goal here is to consider the Reviewer score of each participant who rated a response when computing the Author score for that response. The PageRank algorithm divides a web pages rank (or importance) evenly among all of the pages it points to. So, a webpage with rank x that points to 5 different pages will have twice the impact per vote or link as a webpage with the same rank that points to 10 different pages. This strategy makes sense when considering the corner cases: if an important web page points to every page on the net, then the value of this information is very little. On the other hand, if a very important web page points to only one or two other pages, then those pages must be pretty important and the weight of each vote is scaled accordingly. The network graph on which PageRank operates does not contain negative

or "distrust" edges; that is, a hyperlink from page A to page B is considered to be a positive vote by page A for page B. In the Opinion Space network graph, we not only have participant-weighted links, but we also have distrust information (i.e. negative ratings).

In our proposed model, each participant builds a reputation as a reviewer. The better the reputation of a participant, the more impact her ratings should have on the overall rank of a response. Unlike the PageRank model, it is not a problem when a participant with a really good reputation rates a large number of responses. Hence, we do not necessarily wish to divide a participants reputation across all of the ratings she gives. Instead, we could weigh each of the participants response ratings according to the participants reputation. Let Z_i be participant i's reputation as a reviewer. Then when participant i rates the response by participant j, we can compute the transformed rating as follows.

$$r'_{ij} = \begin{cases} e^{-Z_i}\, r_{ij}\, d_{ij} & \text{if } r_{ij} >= 0 \quad (6.8) \\ e^{-Z_i}\, |r_{ij}|\, (d_{ij} d_{max}) & \text{otherwise} \quad (6.9) \end{cases}$$

To compute this, we require a "good" Reviewer Score model. Let n be the number of responses rated by participant i, r_{ij} be participant i's rating of response j, and μ_j and σ_j be the mean rating and standard deviation for response j. We consider the following models for computing the reputation score of participant i as a reviewer:

$$Z_i = \begin{cases} \frac{1}{n}\sum_{j=1}^{n} |r_{ij} - \mu_j| & \text{Mean Absolute Error} \quad (6.10) \\ \frac{1}{n}\sum_{j=1}^{n} \log_{10} |r_{ij} - \mu_j| & \text{Log Mean Absolute Error} \quad (6.11) \\ \frac{1}{n}\sum_{j=1}^{n} \frac{|r_{ij} - \mu_j|}{\sigma_j} & \text{Absolute Standard Normal} \quad (6.12) \\ \frac{1}{n}\sum_{j=1}^{n} \log_{10}\left(\frac{|r_{ij} - \mu_j|}{\sigma_j}\right) & \text{Log Absolute Standard Normal} \quad (6.13) \\ \sum_{j=1}^{n} \sqrt{\frac{(r_{ij} - \mu_j)^2}{n}} & \text{Root Mean Squared Error} \quad (6.14) \\ U(0,1) & \text{Uniform Random Number} \quad (6.15) \end{cases}$$

Since we are working with an unlabeled data set, it is difficult to objectively compare each of these models to determine the best performing and most robust one. However, it is possible to use simulation to get an idea of how well the models perform under varying conditions. In the next subsection, we describe the design of such a simulation.

6.6.1 Simulation Design

Our goal is to determine which of the Reviewer Score models defined in Equations 6.10 - 6.15 gives the most accurate ranking of reviewers based on their ability to assess the true quality of the responses they rate. To do so, we designed a simulation of reviewer behavior as follows.

The simulation tracks a set of n participants. We assume that every participant has an associated response and rates the responses of some subset of the other participants. Each response is randomly assigned a ground truth "response quality" value, which is a real number in the range $[-1, 1]$, sampled from empirical data collected from Opinion Space. Each reviewer is randomly assigned to one of three "reviewer quality" categories: *good*, *average*, and *poor*. Reviewer i's rating of response j is normally distributed around the mean rating for j with varying degrees of Gaussian noise depending on the quality of the reviewer. Good reviewers are more likely to accurately rate a response than average and poor reviewers, as their level of noise will be lower. The parameters that must be initialized for each run of the simulation are:

- The number of participants in the system
- The percent of responses to be rated by each reviewer
- The percent of good, average, and poor reviewers
- The respective rating noise levels (variances) for good, average, and poor reviewers
- The number of iterations of the simulation to run
- Which model to use for computing reviewer scores

In every run of the simulation, we compute the reviewer score for each participant as a function of the parameters described above. Theoretically, if the model used to compute the reviewer scores works well, then for every pair of reviewers (i, j): if reviewer i is better than reviewer j, i should be ranked above j. Hence, we can evaluate each model by computing the total number of conflicts in its final ranking of the reviewers, where a conflict is the event that reviewer i is ranked below reviewer j when i is better than j. A second option for evaluating the models is to look at the the percent of total conflicts *weighted* by category. That is, if a poor reviewer is ranked above a good reviewer, then the weight on this conflict would be $|3-1| = 2$; intuitively, this scenario is especially bad and should be weighted more heavily when comparing different models.

Revs (g,a,p)	Noise (g,a,p)	MAE	LogMAE	AbsNorm	LogAbsNorm	RMSE	Rand
0.3, 0.5, 0.2	0.05, 0.2, 0.4	5.339	5.399	6.001	7.010	5.374	15.999
0.3, 0.5, 0.2	0.05, 0.3, 0.5	6.125	6.198	6.713	7.705	6.156	16.692
0.3, 0.5, 0.2	0.1, 0.2, 0.3	7.811	7.894	8.574	9.587	7.704	16.679
0.2, 0.5, 0.3	0.05, 0.2, 0.4	6.397	6.485	7.137	8.160	6.348	16.450
0.2, 0.5, 0.3	0.05, 0.3, 0.5	7.349	7.435	7.912	8.805	7.316	16.650
0.2, 0.5, 0.3	0.1, 0.2, 0.3	8.528	8.619	9.280	10.227	8.388	16.330
0.1, 0.6, 0.3	0.05, 0.2, 0.4	2.450	2.470	2.837	3.603	2.579	13.193
0.1, 0.6, 0.4	0.05, 0.3, 0.5	2.449	2.474	2.780	3.606	2.568	13.234
0.1, 0.6, 0.5	0.1, 0.2, 0.3	4.012	4.065	4.703	5.661	4.010	13.743
0.5, 0.4, 0.1	0.05, 0.2, 0.4	6.568	6.666	7.411	8.450	6.477	15.648
0.5, 0.4, 0.1	0.05, 0.3, 0.5	7.908	8.019	8.461	9.275	7.777	15.397
0.5, 0.4, 0.1	0.1, 0.2, 0.3	8.879	8.954	9.547	10.243	8.797	15.630

Table 6.5: Results of reviewer score simulation with 1,000 participants and 25 repetitions per run. The distributions of response quality and the number of responses rated by each participant are sampled from the data collected during the Department of State study. The first column gives the percentage breakdown of good, average, and poor reviewers. The second column gives the breakdown of Gaussian noise levels for good, average, and poor reviewers. Every subsequent column gives the total percent of conflicts counted for each ranking method. For each row, the cell corresponding to the best performing algorithm is highlighted.

6.6.2 Simulation Results

In this section we present the results from two different runs of the simulation, one using data from the Department of State, and the other using data from the Automotive Industry study. We ran the simulation using a variety of parameters describing the breakdown of good, average, and poor reviewers and their respective noise levels for rating accuracy. Each run of the simulation generates 1,000 different participants, and we repeat the run 25 times. For every participant, we then determine what percent of the responses she will rate using empirical data, and then we randomly assign responses for that participant to rate. Each response is assigned a "ground truth" quality score, which is used in conjunction with various degrees of Gaussian noise to generate response ratings.

Tables 6.5 - 6.6 give the simulation results for each of the models considered under varying conditions. The best-performing model is highlighted for each scenario. There are two tables corresponding to each data set: one that describes the results when looking at the number of conflicts (as a percent of the total number of possible conflicts) and a second that gives the weighted number of conflicts. Although it doesn't dominate in every scenario, the Root Mean Squared Error (RMSE) model seems to be the strongest across the various scenarios. In the cases where it does not appear to be the best model, it still behaves reasonably well.

Revs (g,a,p)	Noise (g,a,p)	MAE	LogMAE	AbsNorm	LogAbsNorm	RMSE	Rand
0.3, 0.5, 0.2	0.05, 0.2, 0.4	3.452	3.490	3.896	4.642	3.496	11.871
0.3, 0.5, 0.2	0.05, 0.3, 0.5	3.898	3.944	4.312	5.059	3.943	12.368
0.3, 0.5, 0.2	0.1, 0.2, 0.3	5.208	5.268	5.799	6.591	5.142	12.358
0.2, 0.5, 0.3	0.05, 0.2, 0.4	3.730	3.778	4.179	4.845	3.724	10.864
0.2, 0.5, 0.3	0.05, 0.3, 0.5	4.238	4.289	4.616	5.221	4.237	11.014
0.2, 0.5, 0.3	0.1, 0.2, 0.3	5.124	5.182	5.653	6.308	5.045	10.787
0.1, 0.6, 0.3	0.05, 0.2, 0.4	2.450	2.470	2.837	3.603	2.579	13.193
0.1, 0.6, 0.4	0.05, 0.3, 0.5	2.449	2.474	2.780	3.606	2.568	13.234
0.1, 0.6, 0.5	0.1, 0.2, 0.3	4.012	4.065	4.703	5.661	4.010	13.743
0.5, 0.4, 0.1	0.05, 0.2, 0.4	3.507	3.558	3.964	4.548	3.467	8.954
0.5, 0.4, 0.1	0.05, 0.3, 0.5	4.180	4.239	4.498	4.964	4.119	8.782
0.5, 0.4, 0.1	0.1, 0.2, 0.3	4.772	4.813	5.164	5.576	4.735	8.915

Table 6.7: Results of reviewer score simulation with 1,000 participants and 25 repetitions per run. The distributions of response quality and the number of responses rated by each participant are sampled from the data collected during the Department of State study. For each row, the cell corresponding to the best performing algorithm is highlighted.

Revs (g,a,p)	Noise (g,a,p)	MAE	LogMAE	AbsNorm	LogAbsNorm	RMSE	Rand
0.3, 0.5, 0.2	0.05, 0.2, 0.4	0.826	0.858	0.860	1.398	0.827	29.822
0.3, 0.5, 0.2	0.05, 0.3, 0.5	1.436	1.512	1.472	2.410	1.371	29.662
0.3, 0.5, 0.2	0.1, 0.2, 0.3	2.229	2.305	2.281	3.693	2.163	29.749
0.2, 0.5, 0.3	0.05, 0.2, 0.4	1.091	1.140	1.121	1.930	1.083	30.092
0.2, 0.5, 0.3	0.05, 0.3, 0.5	2.271	2.375	2.326	3.683	2.195	30.161
0.2, 0.5, 0.3	0.1, 0.2, 0.3	2.816	2.929	2.904	4.872	2.688	30.200
0.1, 0.6, 0.3	0.05, 0.2, 0.4	0.181	0.179	0.189	0.193	0.203	24.285
0.1, 0.6, 0.4	0.05, 0.3, 0.5	0.087	0.086	0.085	0.087	0.113	24.280
0.1, 0.6, 0.5	0.1, 0.2, 0.3	0.513	0.517	0.537	0.924	0.582	23.933
0.5, 0.4, 0.1	0.05, 0.2, 0.4	1.303	1.367	1.340	2.414	1.261	27.631
0.5, 0.4, 0.1	0.05, 0.3, 0.5	2.976	3.121	3.028	4.879	2.840	28.009
0.5, 0.4, 0.1	0.1, 0.2, 0.3	3.582	3.718	3.684	5.971	3.395	27.786

Table 6.8: Results of reviewer score simulation with 1,000 participants and 25 repetitions per run. The distributions of response quality and the number of responses rated by each participant are sampled from the data collected during the Automotive Industry study. For each row, the cell corresponding to the best performing algorithm is highlighted.

Revs (g,a,p)	Noise (g,a,p)	MAE	LogMAE	AbsNorm	LogAbsNorm	RMSE	Rand
0.3, 0.5, 0.2	0.05, 0.2, 0.4	0.455	0.471	0.473	0.749	0.466	22.131
0.3, 0.5, 0.2	0.05, 0.3, 0.5	0.754	0.791	0.771	1.238	0.728	21.989
0.3, 0.5, 0.2	0.1, 0.2, 0.3	1.289	1.329	1.319	2.122	1.275	22.060
0.2, 0.5, 0.3	0.05, 0.2, 0.4	0.574	0.598	0.590	0.993	0.578	19.946
0.2, 0.5, 0.3	0.05, 0.3, 0.5	1.160	1.212	1.188	1.870	1.126	19.970
0.2, 0.5, 0.3	0.1, 0.2, 0.3	1.507	1.566	1.553	2.613	1.449	20.035
0.1, 0.6, 0.3	0.05, 0.2, 0.4	0.181	0.179	0.189	0.193	0.203	24.285
0.1, 0.6, 0.4	0.05, 0.3, 0.5	0.087	0.086	0.085	0.087	0.113	24.280
0.1, 0.6, 0.5	0.1, 0.2, 0.3	0.513	0.517	0.537	0.924	0.582	23.933
0.5, 0.4, 0.1	0.05, 0.2, 0.4	0.666	0.698	0.685	1.222	0.648	15.711
0.5, 0.4, 0.1	0.05, 0.3, 0.5	1.501	1.574	1.527	2.453	1.435	15.912
0.5, 0.4, 0.1	0.1, 0.2, 0.3	1.836	1.904	1.888	3.061	1.745	15.792

Table 6.6: Results of reviewer score simulation with 1,000 participants and 25 repetitions per run. The distributions of response quality and the number of responses rated by each participant are sampled from the data collected during the Automotive Industry study. For each row, the cell corresponding to the best performing algorithm is highlighted.

6.7 Reputation Metric III: Accounting for Uncertainty with Confidence Intervals

This method ranks responses according to the lower bound of a 95 percent confidence interval around its mean rating. Intuitively, the lower bound of the confidence interval corresponds to the value at which there is a 95 percent chance that the *true* mean rating is at least that value. Note that this is a conservative approach, as a response with a high sample mean but high variance may be ranked lower than a response with a lower sample mean and lower variance. In essence, this method considers both the sample (estimated) mean rating of a response *and* our confidence in that estimate given the distribution of ratings.

6.7.1 Generalized Confidence Interval

Our goal is to determine the quality of a response X given a collection of n insightfulness ratings $r = (r_1, \ldots, r_n)$ for that response. If we assume that the quality of X is defined by the mean rating for X, then by the Central Limit Theorem (22) we know that the quality of X is normally distributed with mean μ and variance σ^2. Hence, the maximum likelihood estimate of μ is given as follows.

$$\hat{\mu} = \bar{X} = \frac{1}{n} \sum_{i=1}^{n} r_i \tag{6.16}$$

Let Z be the standardized form of $\bar{X}$, which is found by subtracting the mean from $\bar{X}$

and dividing by its standard deviation:

$$Z = \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{n}}} \tag{6.17}$$

We want to know the interval $[-z, z]$ into which the probability that Z belongs is 95 percent, which we define mathematically as follows.

$$\Pr(-z \leq Z \leq z) = 1 - \alpha = 0.95 \tag{6.18}$$

Z is a normally distributed random variable, since transforming a normal random variable by subtracting a constant from and dividing by a constant yields a normal random variable (22). Hence, we can use the cumulative distribution function for standard Normal random variables to find z:

$$\begin{aligned} \Phi(z) &= P(Z \leq z) = 1 - \frac{\alpha}{2} = 0.975 && (6.19) \\ \Rightarrow z &= \Phi^{-1}(\Phi(z)) \\ &= \Phi^{-1}(0.975) \\ &= 1.96 && (6.20) \end{aligned}$$

This gives us the following formulation for a 95 percent confidence interval on the mean (quality) of response X:

$$\begin{aligned} 0.95 &= 1 - \alpha \\ &= \Pr(-z \leq Z \leq z) \\ &= \Pr\left(-1.96 \leq \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{n}}} \leq 1.96\right) \\ &= \Pr\left(\bar{X} - 1.96\frac{\sigma}{\sqrt{n}} \leq \mu \leq \bar{X} + 1.96\frac{\sigma}{\sqrt{n}}\right) && (6.21) \end{aligned}$$

To solve for the lower bound of the confidence interval, we require an estimate of the variance (σ^2) of $\bar{X}$.

This model assumes that we know the variance (σ^2) of $\bar{X}$. Since we do not know the true value, we must come up with a statistical model for the variance and estimate it using the ratings data available. In the following section we derive a model for this purpose.

6.7.2 Statistical Model of Rating Data for Textual Responses in Opinion Space

Our goal is to parametrically model the distribution of insightfulness ratings for each response. We model the variable X as a spike at 0 with probability p_1, a spike at 1 with probability p_2, and then a mixture of two Normal variables $X_3 \sim N(\mu_3, \sigma_3)$ and

$X_4 \sim N(\mu_4, \sigma_4)$ with probabilities p_3 and p_4, respectively. Observe that the maximum likelihood estimates for p_1 and p_2 can easily be found by:

$$p_1 = \frac{\#\text{ of ratings for } X \text{ with a value of } 0}{\text{Total } \#\text{ of ratings for } X} \tag{6.22}$$

$$p_2 = \frac{\#\text{ of ratings for } X \text{ with a value of } 1}{\text{Total } \#\text{ of ratings for } X} \tag{6.23}$$

Unlike our previous models in which we empirically estimated the standard error, this model yields a parametric estimate of the same quantity. Below we derive $E(X)$ and $Var(X)$ in that order, as the former is required for the derivation of the latter. First we formally define the variable:

$$\begin{aligned} X = {} & I(0, p_1) \times 0 + I(p_1, p_1 + p_2) \times 1 + I(p_1 + p_2, p_1 + p_2 + p_3) \times X_3 \\ & + I(p_1 + p_2 + p_3, 1) \times X_4 \end{aligned} \tag{6.24}$$

Where $I()$ is an indicator variable corresponding to the event that the rating falls within the corresponding "bin" (was generated by the corresponding random variable). It then follows that:

$$\begin{aligned} E(X) &= E(I(0, p_1) \times 0 + I(p_1, p_1 + p_2) \times 1 \\ &\quad + I(p_1 + p_2, p_1 + p_2 + p_3) \times X_3 + I(p_1 + p_2 + p_3, 1) \times X_4) \\ &= p_1 E(0) + p_2 E(1) + p_3 E(X_3) + p_4 E(X_4) \\ &= p_2 + p_3 \mu_3 + p_4 \mu_4 \end{aligned} \tag{6.25}$$

Conditioned on a rating not belonging to either the "0" or "1" bins, let λ be the probability that the rating was generated by $N(\mu_3, \sigma_3)$. Then the mixing probabilities for X_3 and X_4 are given by

$$p_3 = \lambda(1 - (p_1 + p_2)) \tag{6.26}$$

$$p_4 = (1 - \lambda)(1 - (p_1 + p_2)) \tag{6.27}$$

Letting $\mu = p_2 + p_3 \mu_3 + p_4 \mu_4$, we then calculate the variance of X, σ_X^2, as:

$$\begin{aligned}
\sigma_X^2 &= \sum p_i E\left(X_i - E\left(X\right)\right)^2 \\
&= p_1\left(0-\mu\right)^2 + p_2\left(1-\mu^2\right) + p_3 E\left(X_3-\mu\right)^2 + p_4 E\left(X_4-\mu\right)^2 \\
&= p_1\mu^2 + p_2\left(1-\mu^2\right) + p_3 E\left(X_3^2\right) + p_3 E\left(-2X_3\mu+\mu^2\right) \\
&\quad + p_4 E\left(X_4^2\right) + p_4 E\left(-2X_4\mu+\mu^2\right) \\
&= p_1\mu^2 + p_2\left(1-\mu^2\right) + p_3\left(\sigma_3^2+\mu_3^2\right) + p_3 E\left(-2X_3\mu+\mu^2\right) \\
&\quad + p_4\left(\sigma_4^2+\mu_4^2\right) + p_4 E\left(-2X_4\mu+\mu^2\right) \\
&= p_1\mu^2 + p_2\left(1-\mu^2\right) + p_3\sigma_3^2 + p_3 E\left(\mu_3^2 - 2X_3\mu+\mu^2\right) \\
&\quad + p_4\sigma_4^2 + p_4 E\left(\mu_4^2 - 2X_4\mu+\mu^2\right) \\
&= p_1\mu^2 + p_2\left(1-\mu^2\right) + p_3\sigma_3^2 + p_3\left(\mu_3-\mu\right)^2 + p_4\sigma_4^2 + p_4\left(\mu_4-\mu\right)^2 \\
&= p_1\mu^2 + p_2\left(1-\mu^2\right) + p_3\sigma_3^2 + p_3\left(\mu_3-\mu\right)^2 + p_4\sigma_4^2 + p_4\left(\mu_4-\mu\right)^2 \qquad (6.28)
\end{aligned}$$

The Standard Error is computed by

$$SE_X = \sqrt{\frac{\sigma_X^2}{n}} \qquad (6.29)$$

and the error-adjusted final score for the response is

$$\text{Score } = \bar{X} - 1.96 \times SE_X \qquad (6.30)$$

6.7.3 Derivation of EM Algorithm for Estimating Parameters

Equations 6.22 and 6.23 give us maximum likelihood estimates for p_1 and p_2, the probabilities that a rating is a 0 or 1 respectively. However, computing the variance estimate of X according to the formula derived in Equation 6.28 necessitates empirical estimates of λ (to find p_3 and p_4), μ_3, σ_3^2, μ_4, and σ_4^2. To make these estimates, we use the derivation of the Expectation-Maximization (EM) method described in Algorithms 2, 3, and 4 below. This method is commonly used for parameter estimation for unobserved latent variables in statistical models. It is a two-step iterative method, where one step (the M step) is designed to find the maximum likelihood estimate of the parameters, and the other (the E step) is used to estimate the mixing probability between the two inner Normal distributions given those parameter estimates. This process is iterated until it converges and up to 1,000 times.

Before we run the algorithm, we pre-process the ratings data by removing all 0- and 1-valued ratings, which enables us to focus on finding parameters to describe the ratings in the range $(0, 1)$.

The Expectation (E) Step takes in as parameters estimates for the values of λ, μ_3, σ_3^2, μ_4, and σ_4^2 and the set of ratings $r = \{r_1, \ldots, r_n\}$ for the response in question. Let n be the number of ratings, and let $I = \{I_1, \ldots, I_n\}$ be a set of n variables, where I_j corresponds to the probability that rating r_j was generated by $N(\mu_3, \sigma_3)$ instead of $N(\mu_4, \sigma_4)$. Recall that λ is the probability that a randomly sampled rating

is generated by the left-most Normal distribution, $N(\mu_3, \sigma_3)$. Let $f(x|\mu, \sigma)$ be the probability density function of the Normal distribution with mean μ and standard deviation σ.

For each rating r_i collected for this response, we compute p_3 as the marginal probability that r_i was generated by $N(\mu_3, \sigma_3)$ and p_4 as the marginal probability that r_i was generated by $N(\mu_4, \sigma_4)$. Given these values, the probability that r_i was generated by $N(\mu_3, \sigma_3)$ is computed in step 7 and assigned to I_i. Once computed for each rating, the set of indicator probabilities are passed to the Maximization (M) Step, along with the original rating values.

The M Step uses I to update our estimates for all of the parameters used to describe our statistical model. Hence, λ is computed as the average of the values of $\{I_1, \ldots, I_n\}$. The mean μ_3 of the left-most Normal distribution is the average value of the ratings weighted by I. Similarly, the mean μ_4 of the right-most Normal distribution is the average values of the ratings weighted by $(1 - I)$. The estimates for the variances also follow the standard formula, weighted by I.

To run the algorithm, we first initialize the values for λ, μ_3, σ_3^2, μ_4, and σ_4^2. This can be done by making an educated guess based on our observations of the data. In this case, we initialize λ to be 0.5, μ_3 to be 0.25, μ_4 to be 0.75, and the variances to be 0.05.

Algorithm 2 Expectation (E) Step

Require: $r = \{r_1, \ldots, r_n\}$ is the set of ratings for the current response, excluding those with values of 0 or 1

1: **procedure** ESTEP($\lambda,\ \mu_3,\ \sigma_3^2,\ \mu_4,\ \sigma_4^2, r$)
2: n = number of ratings
3: $I = \{I_1, \ldots, I_n\}$
4: **for** $i = 1, \ldots, n$ **do**
5: $p_3 \leftarrow \lambda\, f(r_i \mid \mu_3, \sigma_3)$
6: $p_4 \leftarrow (1 - \lambda)\, f(r_i \mid \mu_4, \sigma_4)$
7: $I_i \leftarrow \frac{p_3}{p_3 + p_4}$
8: **end for**
9: **return** I
10: **end procedure**

6.7.4 Performance on Empirical Data

Now that we have a parametric statistical model for response ratings in Opinion Space, we require a way to evaluate how well the model describes the data. To do so, we derive a Chi-Squared goodness of fit test and run it on the various data sets we have accumulated. We define the following hypotheses:

- H_0 = the parametric distribution fits the data. That is, the observed data is a combination of two Normal distributions: $\alpha X_3 + (1 - \alpha) X_4$, where X_3 is described

Algorithm 3 Maximization (M) Step

Require: $r = \{r_1, \ldots, r_n\}$ is the set of ratings for the current response, excluding those with values of 0 or 1. $I = \{I_1, \ldots, I_n\}$ indicates the probability that the rating was generated by $N(\mu_3, \sigma_3)$ instead of $N(\mu_4, \sigma_4)$.

1: **procedure** MSTEP(I, r)
2: $\lambda \leftarrow \frac{\sum_i I_i}{n}$
3: $\mu_3 \leftarrow \frac{\sum_i I_i r_i}{\sum_i I_i}$
4: $\sigma_3^2 \leftarrow \frac{\sum_i I_i (r_i - \mu_3)^2}{\sum_i I_i}$
5: $\mu_4 \leftarrow \frac{\sum_i (1-I_i) r_i}{\sum_i (1-I_i)}$
6: $\sigma_4^2 \leftarrow \frac{\sum_i (1-I_i)(r_i - \mu_4)^2}{\sum_i (1-I_i)}$
7: **return** λ, μ_3, σ_3^2, μ_4, σ_4^2
8: **end procedure**

Algorithm 4 Running the iterations of the EM algorithm

Require: Parameters must be initialized to some reasonable estimated value.

1: **procedure** EM(λ, μ_3, σ_3^2, μ_4, σ_4^2)
2: **for** $i = 1, \ldots, 1000$ **do**
3: $I \leftarrow Estep\left(\lambda, \mu_3, \sigma_3^2, \mu_4, \sigma_4^2\right)$
4: $\left(\lambda, \mu_3, \sigma_3^2, \mu_4, \sigma_4^2\right) \leftarrow Mstep\left(I\right)$
5: **end for**
6: **return** λ, μ_3, σ_3^2, μ_4, σ_4^2
7: **end procedure**

by $N(\mu_3, \sigma_3)$ and X_4 is described by $N(\mu_4, \sigma_4)$.

- H_1 = the parametric distribution does not fit the data.

Working with only ratings in the range $(0, 1)$, we partition the distribution into m evenly-spaced bins: $(0, b_1)$, $[b_1, b_2)$, ..., $[b_{m-1}, 1)$. Let O_k be the observed number of response ratings that fall in the range corresponding to bin k. For convenience, we normalize the counts by dividing O_k by the total number of observed ratings; this forces $\sum_k O_k = 1$.

E_k is the expected number of response ratings to fall in the range corresponding to bin k as determined by our parametric model. (However, since we normalized the observed counts to sum to 1, E_k is also the probability that a rating falls in bin k.) Let $F(x|\mu, \sigma)$ be the cumulative distribution function (CDF) for a Normally distributed random variable with mean μ and standard deviation σ. Then E_k is computed as follows:

$$\begin{aligned} E_k &= \alpha \left[F\left(b_{k+1}|\mu_3, \sigma_3\right) - F\left(b_k|\mu_3, \sigma_3\right) \right] \\ &\quad + (1-\alpha) \left[F\left(b_{k+1}|\mu_4, \sigma_4\right) - F\left(b_k|\mu_4, \sigma_4\right) \right] \end{aligned} \tag{6.31}$$

Since the X_3 and X_4 distributions are truncated Normals, we need to ensure that the sum of the E_k's is 1. This can be done by dividing each E_k by $\sum_k E_k$ in a subsequent step.

Given the observed and estimated data, we can compute the associated χ^2 statistic with:

$$\chi^2_k = \sum_{k=1}^{m} \frac{(O_k - E_k)^2}{E_k} \tag{6.32}$$

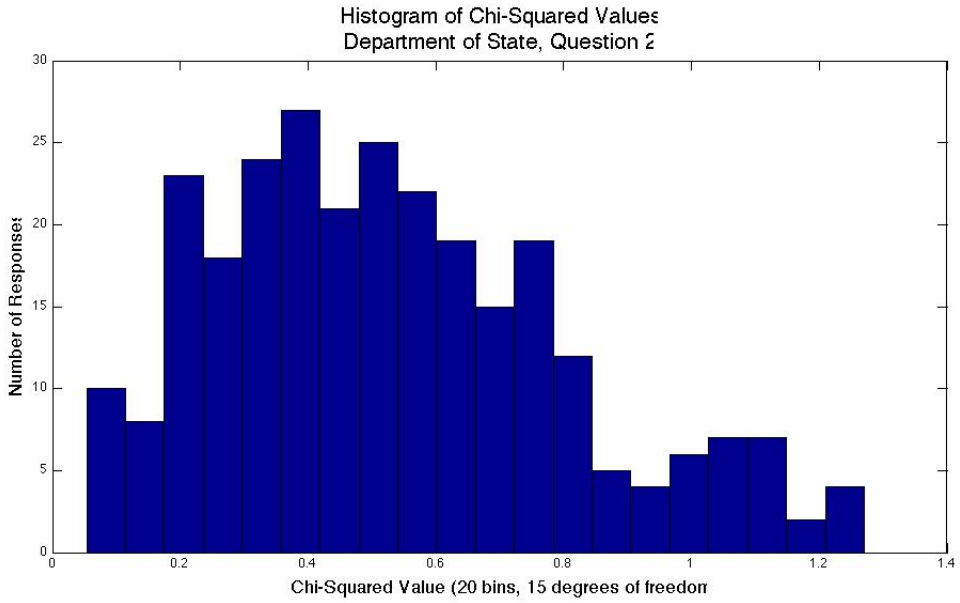
Since our statistical model estimates four parameters, the number of degrees of freedom is $m-5$. If we let $m = 20$, then the critical upper value for the Chi-Squared distribution with probability 0.001 and 15 degrees of freedom is 3.48. The results from this test are given in Table 6.9 below. We find that for all responses in both data sets, the Chi-Squared value is well under the critical upper limit, and thus we accept the Null Hypothesis that our parametric model describes the data. Figure 6.7 illustrates the distribution of the Chi-Squared values. As expected, there is less variability in the values for the Automotive Industry data, since the data set is significantly more dense than that of the Department of State.

Figure 7.6 shows scatter plots for two different data sets of the mean insightfulness rating for each response versus the computed standard error. We observe that there is no discernible correlation between the two measures, which implies that with our parametric model, the quality of a response is independent of the variability in the ratings it receives.

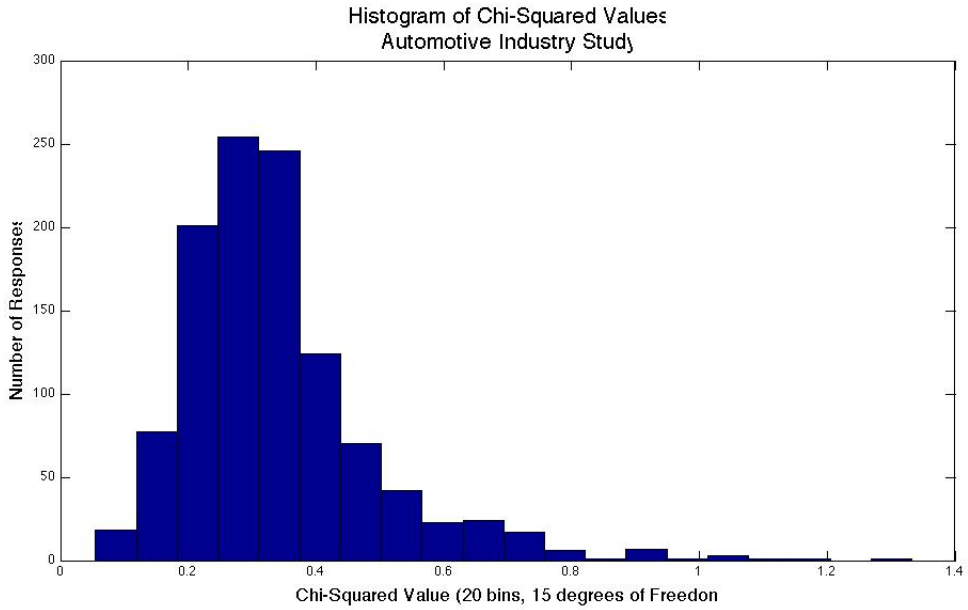
6.8 Empirical Data and Results

In Section 6.4 we described the properties of the data sets we have available. The two data sets with the highest activity levels are 1) the Department of State, Question 2 (DSQ2), and 2) the Automotive Industry (AI) study. As previously mentioned, the DSQ2 instance did not provide participants with monetary or tangible incentives for participation; that is, participation was entirely self-motivated. The AI instance of Opinion Space, however, promised 100 dollar gift certificates to the ten most insightful participants. This difference in motivations resulted in data sets with significantly different properties, especially in terms of participation. Hence, we consider both data sets for our analysis to get a better idea of the behavior of our algorithms and models under these different circumstances. We consider the top 10 responses returned by five different methods:

1. The Spatial approach outlined in Section 6.5.

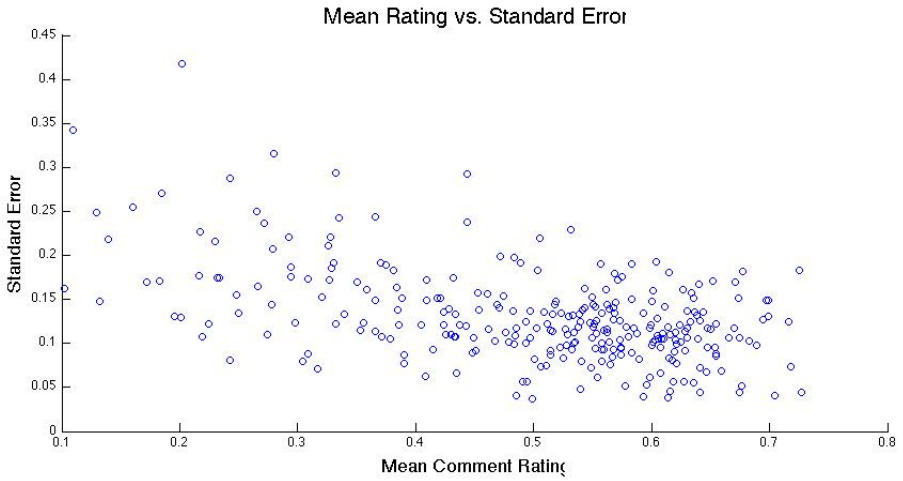


(a) Department of State

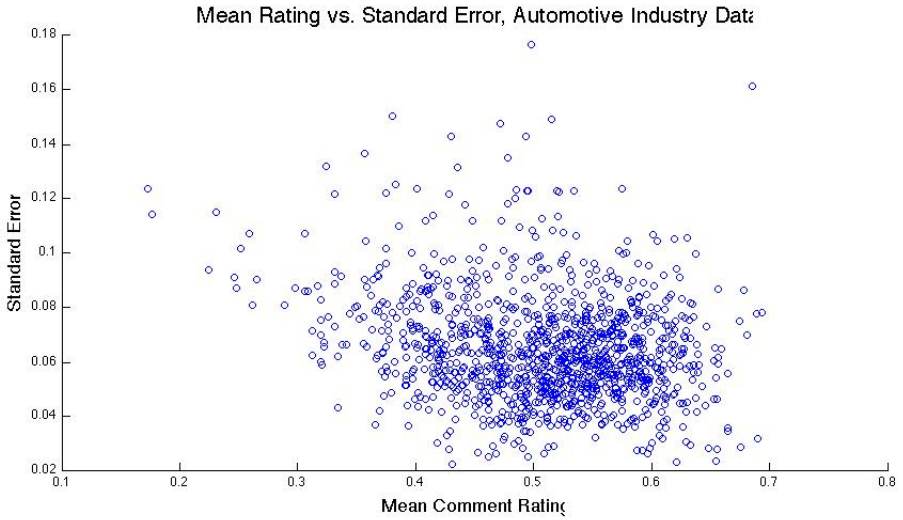


(b) Automotive Industry

Figure 6.7: Distribution of Chi-Squared values for data from (a) the Department of State Question 2, and (b) the Automotive Industry study.



(a) Department of State



(b) Automotive Industry

Figure 6.8: Scatter plot of the mean rating for each response as compared to the standard error of the ratings for that response for (a) data from the Department of State study, discussion question 2, and (b) the Automotive Industry study.

Data Set	Question	Avg χ^2	Std Dev	Min	Max
DoS	2	0.53	0.27	0.05	1.27
Auto	1	0.34	0.15	0.06	1.33

Table 6.9: Results from Chi-Squared goodness of fit test.

2. Combining reviewer and author scores, as described in Section 6.6.

3. Combining the Spatial algorithm with the Standard Error computed in Section 6.7: the score of a response is equal to its Spatial author score divided by the standard error.

4. Combining all three: the reviewer score, Spatial author score, and standard error.

5. The Confidence Interval method of Section 6.7.

Figure 6.9 depicts a bump chart comparing the ranking top ten responses for each method run on the Automotive Industry data set. Each column corresponds to a different method, and the boxes in each column reflect the responses returned by the corresponding method. Two boxes with the same label correspond to the same textual response. The boxes that are shaded gray reflect responses that were ranked in the top ten by that method only; that is, no other method found that response to be in the top ten.

As can be seen, there is a significant amount of overlap between the methods. All three of the methods that extend the Spatial method (columns 2 - 4) return the exact same set of responses; the only difference is the ordering. Further, these three methods all seem to dampen the ranks of the top responses returned by the Spatial method alone.

Tables 6.10 and 6.11 give us a closer look at what is going on. In Table 6.10 we give the Levenshtein edit distances between the rankings of the top ten responses from each method; this measures the number of edits required to transform one ranking into another, where possible operations include insertions, deletions, and substitutions. Table 6.11 gives the Spearman rank correlation coefficient for each pair of methods, which measures the statistical dependence between ranked lists. We find that methods (2) and (3) yield the most similar ranks, but methods (1) and (3) have the highest correlation. Overall, the Confidence Interval (CI) method gives the least similar and least correlated ranking in comparison to the other four methods. This is likely because the scores are largely dependent on the average rating, whereas the scores for the other methods are dependent on the sum of the transformed ratings. Although these two classes of methods differ so drastically, they still return a significant number of responses in common. The fact that two independent (types of) methods return similar top-10 lists indicates that our methods are working reasonably well.

Method	(2)	(3)	(4)	(5) CI
(1) Spatial	8	7	9	10
(2) Rev Score + Spatial		4	6	10
(3) Spatial + SE			5	10
(4) Rev Score + Spatial + SE				10

Table 6.10: Levenshtein edit distances between rankings given by the methods under consideration.

Method	(2)	(3)	(4)	(5) CI
(1) Spatial	0.80	0.8361	0.80	0.68
(2) Rev Score + Spatial		0.69	0.77	0.64
(3) Spatial + SE			0.62	0.64
(4) Rev Score + Spatial + SE				0.65

Table 6.11: Spearman's rank correlation coefficient between the methods under consideration, measured on the ranks of the top 20 responses returned by each method.

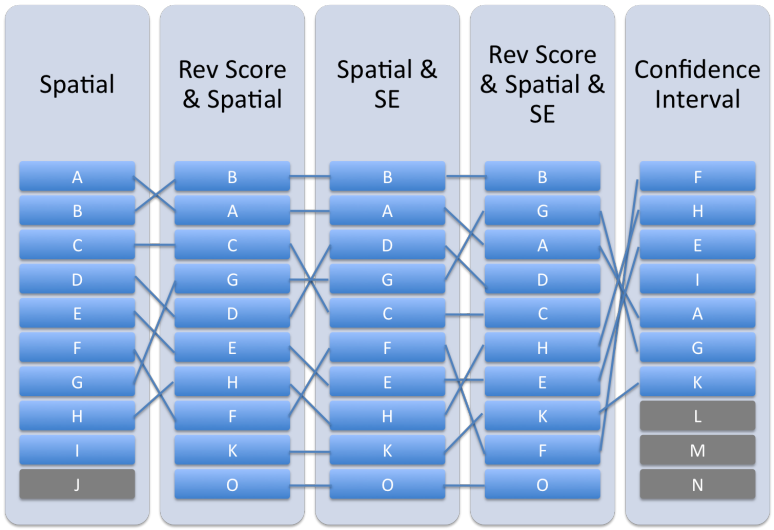


Figure 6.9: Bump chart comparing the top ten responses found by the various methods we considered on the Automotive Industry data set. Each box corresponds to a different response, and boxes with the same label correspond to the same response. Gray boxes reflect responses that were unique to that model (i.e., they were not returned in the top ten by any other method).

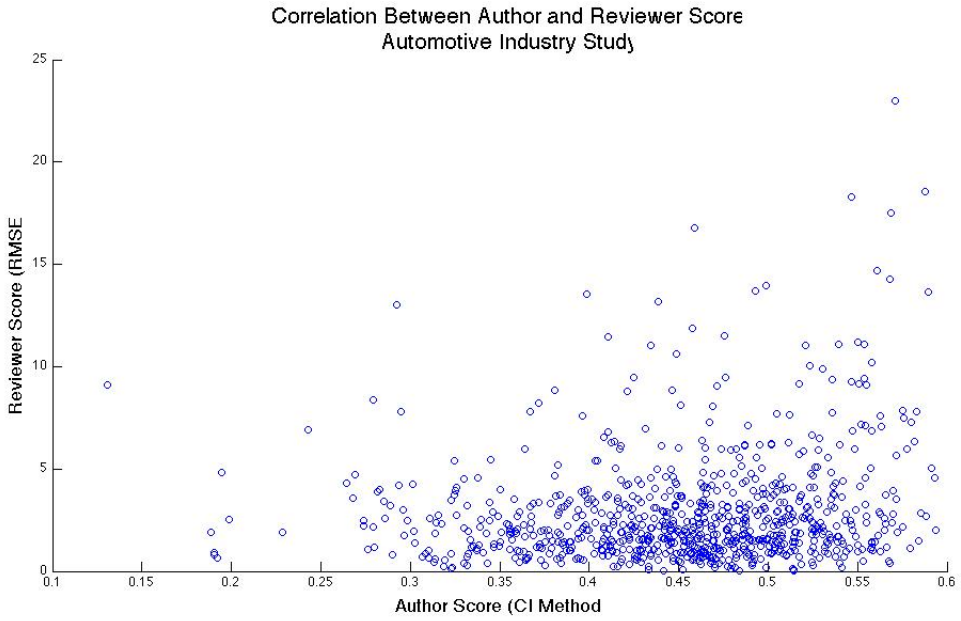


Figure 6.10: Scatter plot depicting the small amount of correlation between Author and Reviewer scores in the Automotive Industry data set. The Author scores were computed using the Confidence Interval Method, and Reviewer Scores were computed using Root Mean Squared Error. In this case, the Pearson correlation is 0.1612.

6.8.1 Ranking Responses by Average Insightfulness Rating

Many user-generated content sites on the web employ some protocol for managing information that asks users to rate comments, reviews, or story submissions. The ratings for each item are typically averaged to form a ranking (Josang et. al, 2007). While this is a simple and intuitive method for filtering out irrelevant or inappropriate material, it is subject to many known problems, including rank reversal in the presence of irrelevant alternatives (Hochbaum and Levin, 2006). By definition, the averaging model for ranking also tends to be dominated by extreme behavior, which is precisely what we wish to filter out. Since humans are ultimately the ones evaluating the ranked outputs of these systems, a natural measure of quality should also be based on human evaluation.

In a preliminary study, we used the standard averaging model on insightfulness ratings to rank the comments for four different discussion questions that had been hosted on Opinion Space. While in all cases the averaging measure did return some insightful responses to the top ten, there were some disappointingly unintelligent responses as well. For example, the top responses to the discussion question: *If you met U. S. Secretary of State Hillary Clinton, what issue would you tell her about, why is it important to you, and what specific suggestions do you have for addressing it?* include

- keep on working hard

- Encourage a strong European Union and work in a confident way with it

- The rest of the rest of the world doesn't want to be like you ie. U.S. of A. Stop using your weapons and weapon sales to subjugate others. Only through diversity will we survive as species.

These responses are short and difficult to justify as thoughtful, insightful, and in some cases, relevant. They do not provide any new insights or ideas that the Department of State could realistically consider, which defies the original purpose of the system.

To contrast, the top response for the Spatial Reputation model in the same data set is the following:

> *One of the big problems right now is the lack of visibility of the Foreign Service as an arm of American foreign policy, as compared to the military. This results in a popular misconception that diplomacy is "only talking," or that the military should be the first resort in a conflict. On the other hand, it is true that the military in Iraq and Afghanistan *are* the face of America to many civilians. Effectively, they are doing public diplomacy. It seems like the State Department could do more a) to educate Americans and others about what diplomacy does and does not do, and b) educate military personnel about how to be more skillful public diplomats. Every time I read about a military class on "cultural awareness" led by a military trainer, I wonder: Where is the Foreign Service in the picture?*

Many would agree that this response is significantly more thought-provoking and insightful. Further, the top ten responses according to the Spatial model does not include any clearly disappoint responses such as those returned by the averaging model.

Curious as to why we were getting unreliable results with the averaging model, we took a closer look at the distributions of the ratings we were collecting. Interestingly, the ratings for a comment tend to be the exact opposite of a Gaussian, with peaks on the left and right extremes and two humps in between. As a result, the mean is an insufficient statistic to describe the distribution, and comparing two responses based only on their mean ratings will be subject to high degrees of noise. To better quantify the variability in the quality of the top responses found by average insightful ratings, we computed the average standard error of the top 10 comments. The standard error is defined as the standard deviation divided by the square root of the sample size. Mathematically, it equates to the standard deviation of the sample mean estimate of a population mean. The average standard error of the responses returned for the second question was 0.061 with a standard deviation of 0.019. In comparison, the spatial ranking model that we use (described below), returned a set of responses from the same data set with an average standard error of 0.029 and a standard deviation of 0.009, which is significantly lower.

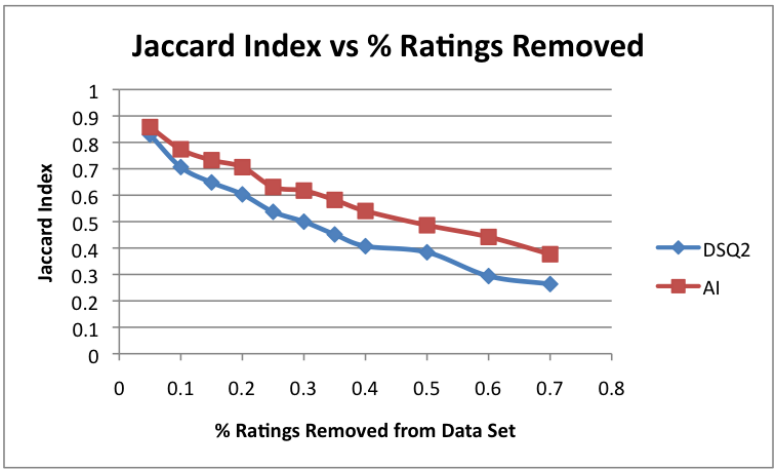


Figure 6.11: Plot of the convergence behavior of the Spatial Reputation model as a percentage of the ratings data are removed from the data set. The analysis was performed on both the Department of State question 2 (DSQ2) and Automotive Industry (AI) data sets.

6.8.2 Sensitivity to Number of Ratings

We evaluated the convergence properties of the Spatial Reputation model by comparing the ranking results when a percentage of the response ratings are removed from the data set. The Jaccard Index is used to compare the overlap of the top 30 responses found in this way with the top 30 responses found by running the model on the original, complete data set. This metric computes the size of the intersection between two sets over the size of their union; a value of 1 indicates complete agreement between the sets, and a value of 0 indicates complete disagreement. The specific steps of our analysis can be described as follows.

1. Randomly remove x percent of the response ratings from the data set.
2. Find the top 30 responses according to the Spatial Reputation model.
3. Compute the value of the Jaccard Index with the top 30 responses determined by the complete data set.
4. For statistical significance, repeat 40 times and take the average.

Figure 6.11 shows the convergence rate of the Spatial Reputation model on two data sets: the question 2 of the State Department data (DSQ2) and the Automotive Industry study (AI). Since the AI data set is significantly more dense than DSQ2 in terms of the number of ratings per response, it performs slightly better when a fixed percentage of the ratings are removed. As can be seen, the convergence rate is fairly fluid; removing 50 percent of the ratings results in approximately 50 percent overlap

in the top 30 responses. For both data sets, there are significant drops when 5 and 10 percent of the ratings are removed, indicating that both data sets have not yet reached saturation. That is to say, the ranking of the responses have not yet reached a stable, steady state.

6.9 Summary

In this chapter we presented three different mathematical models for participant reputation in Opinion Space: a Spatial Reputation model that corrects for participant tendencies to give higher ratings to those that are closer to themselves in the Opinion Space map, a Reviewer Reputation model assessing the ability of a participant to accurately rate a response, and a statistical model based on confidence intervals for determining response quality. The Reviewer Reputation model can be combined with either of the other models as an intelligent way to weight response ratings according to their reliability. The Spatial model is appropriate for use when participants exhibit significant spatial bias towards those with similar opinions, whereas the Statistical model may be more appropriate in situations where this effect is not present.

We showed that the results returned by the Spatial Reputation model converges smoothly as more response ratings are collected and that it returns significantly better results than the simple average model. We also showed that our statistical model for response ratings is extremely accurate, passing the Chi-squared goodness of fit test with a p-value less than 0.001 for all responses with at least 20 ratings.

6.10 Future Work: Using Ensemble Learning Theory

In this chapter we have presented a variety of methods for ranking textual responses in Opinion Space. While there are many responses on which the methods agree should be in the top ten, there are also a handful of responses on which they disagree. We assume that if two independent methods identify the same response as highly insightful, there is a greater likelihood that this is the case. This motivates the design of a meta-method that aggregates the results from the different algorithms based on consensus.

"Ensemble systems" or "multiple classifier systems" or "a mixture of experts" are a class of models in theoretical computer science for classification that have been shown to be stronger predictors than single-classifier models (111). These models take as input predictions from a set of various classifiers, combine the predictions in some way, and then output a single prediction. When the truth is revealed, the models adjust the weights given to the different classifiers based on performance, and then a new prediction is made. Unfortunately, with Opinion Space we don't have the luxury of learning the ground truth ranking. However, we may still be able to borrow concepts from ensemble learning theory to design a meta-algorithm. One idea is to iteratively apply Adaptive Boosting (50) to identify the responses that require further review, as they may have been mis-classified.

7

Concluding Remarks: Reputation and Filtering from a New Media Perspective

7.1 Introduction

In this chapter we provide concluding remarks on reputation and filtering from a New Media perspective. We consider the social responsibility of web designers when building dynamic, social spaces for information/opinion gathering and dissemination. The chapter begins with a a closer look at the consequences of various design decisions often made, which leads into a discussion of the requirements and challenges of enabling deliberative democracy in an online setting. We then propose future work along two directions: robust methods for modeling the reputation of individual participants, and collaboratively filtering the opinions of others while still upholding the ideals of deliberative democracy and the public sphere.

7.1.1 A Closer Look at Online Discussion Forums

The communities that grow around many online discussion forums are self-selected in that each website tends to attract a certain type of participant. The Huffington Post (`www.huffingtonpost.com/`), for example, is overwhelmingly dominated by liberals, and any conservative opinions are often met with hostility from the community. While self-selected discussion groups are particularly valuable for minority groups seeking to deliberate in order to form a unified identity or front, as noted by Fishkin they fail to serve as an open platform for deliberative democracy:

> Discussion groups achieve deliberation among unrepresentative groups. For that reason they serve the enlightenment of the participants, but they do not offer a voice for we the people. (46)

Several studies of online discussion forums have shown that the traditional setup of linear or threaded comment lists are insufficient to create a positive learning environment (122; 143). Bishop (9) presents a theoretical framework for understanding what factors encourage visitors to participate in online discussion forums, and Brandtzaeg and Heim (14) describe a user study of participation in several popular Norwegian online forums. Dahlgren argues in (31) that one of the dangers of online deliberations is fragmentation of the participants. While Berinsky (8) lauds public opinion polling as one of the most inclusive means for participating in political discussions, he is critical of its inherent bias. He argues that one of the contributing factors to such bias is that a portion of the population may simply not know how to respond to a particular question and thus abstain.

Sack (125) provides a review of theories in discourse analysis and very large-scale conversation (VLSC). He developed the Conversation Map to output qualitative diagrams rather than quantitative summaries of online conversations. Unlike Opinion Space, Conversation Map is designed to help users navigate a VLSC via a *semantic map* of key words in the conversation which are determined via statistical and linguistic analysis.

7.1.2 Setting the Stage for Deliberative Democracy Online

The notion of the Internet as a facilitator of deliberative democracy is riddled with contradictions. On one hand, cyberspace is a realm of abstractions that removes prejudice-inducing components such as class and appearance from social interactions. Because pseudonyms are cheap, and often free, on the Web and true identities can be kept private with little effort, participants do not see any real-world social repercussions for their behavior in online discussions. In theory, this is the perfect setting for eliciting open and honest deliberation on controversial issues that do not have a clear or determinable solution. (112)

At the same time, Dahlberg argues in (30) that there are several factors that counteract the open nature of online discourse, which include disrespect for dissenting opinions and the tendency for mob behavior to dominate discussions. These factors contribute to the phenomenon known as the *spiral of silence*. First coined by Noelle-Neumann in (106), the spiral of silence refers to the inclination for individuals holding a minority opinion to remain silent out of fear of humiliation or isolation. Miller explains in (98) that to break free of this trend towards conformity there must be individuals that are willing to speak up without fearing adverse social consequences. Although the Internet is often lauded as a space where people can feel relatively more comfortable sharing their opinions than they feel in face-to-face interactions, with the deluge of information posted and the use of popularity-based filtering systems minority opinions are less likely to stand out in online discussions. This effect can be exacerbated by the absence of social cues such as facial expressions, body language, gender, age range, and appearance. (139) Consequently, it is difficult for participants to quickly understand the *true* breadth of the diversity of opinions held by others in the discussion, and as a result they may be less likely to voice seemingly less-popular views.

Dahlberg (30) extends Habermas' theory on communicative rationality (65) to propose six requirements for deliberative democracy to succeed. These include: independence from state and economic power, exchange and critique of criticizable moral-practical validity claims, reflexivity, ideal role-taking, sincerity, and discursive inclusion and equality. It can be argued that the most crucial requirement is for a *diversity* of opinions to be presented and considered in ernest, for without this the online component of the public sphere cannot exist. Instead, the spiral of silence encourages those that are "silenced" to form self-selected subgroups where they feel more comfortable speaking and more secure that their opinions will be heard; consequently, these subgroups only serve to reinforce their own, collective opinions and are not necessarily constructive towards finding a unifying solution to the problem at hand. We argue that for the public sphere to truly exist online, these groups, or the larger groups from which they splintered off, must find a way to engage in earnest debate. People must be willing to respect and consider opposing opinions in online forums, otherwise the dividing lines between different groups in a given Internet discussion are likely to become rigid and opaque.

Unfortunately, most social media websites today are unable to create a space that encourages participants to explore, empathize with, and respond to a diversity of opinions. These sites typically consist of a linear list of textual responses that may or may not be threaded, and a single topic or thread can receive upwards of thousands of responses within a matter of hours. To help participants cope with information overload, many sites ask them to rate the responses they read and then highlight those with the highest average rating. The problem with this approach is that only the responses reflecting the most popular points of view are emphasized, essentially eliminating from the equation any dissenting view regardless of quality; this serves to silence the minority and encourages conformist behavior from the participants.

In the following section we delve deeper into the design requirements for a well-functioning public sphere and deliberative democracy in an online setting.

7.1.3 The Public Sphere and the Internet

Habermas (66) defines the public sphere as "a realm of our social life in which something approaching public opinion can be formed." Under this definition, he requires that all members of society have access to the public sphere and the freedom to assemble and express their opinions. Dahlgren (31) deconstructs the public sphere into three dimensions:

1. *Structural:* The structural dimension highlights the need for a structured means of communication and deliberation, particularly in terms of access and freedom of speech.

2. *Representational:* This dimension considers the dispersion of information and whether it is accurate, fair, and representational of multiple views.

3. *Interactional:* Finally, the interactional dimension of the public sphere is a consideration of citizens as sharers of opinions and information rather than simply consumers.

When it comes to the Internet, Dahlgren notes that the representational and interactional dimensions are *not* orthogonal in that the separation between citizens and the media by which the opinions of the public sphere are defined becomes blurred.

Traditionally, the public sphere has allowed for "one-to-one" and "one-to-many" forms of communication between citizens. Prior to the advent of the Internet, the primary media of the public sphere consisted of radio, television, and print media such as newspapers and magazines; these forms are inherently unidirectional streams of information, where citizens are unable to respond in a public or timely manner. In that sense, the interactional component was minimal and hence there was no guarantee that the representational dimension was intact.

The Internet today provides, for the first time in history, a structured and efficient environment for communication on a "many-to-many" level. The interactional dimension of the public sphere flourishes on the Internet, where participating citizens can express their opinions freely, be heard by many, and respond to each other in real-time.

On the other hand, many traditional discussion tools on the Internet have serious shortcomings when it comes to supporting the representational and structural dimensions of the public sphere. Structurally, websites act as institutions that facilitate large-scale discussions between thousands or more people. At the same time, when these thousands of people are expressing their opinions through this medium, the challenge that many websites and forums have failed to overcome is to organize the discussion in a meaningful way that grants equal access to all participants and viewpoints. Often times, the majority opinion is deemed by the community as the "ground truth," and any further deliberation is in response to extreme, provocative comments that lack any merit. This phenomenon is commonly referred to as *flaming*, and it plagues many communities not moderated by humans. Another vulnerability faced by these forums is to *trolling*, which is when a user (*troll*) intentionally posts "messages that lure members of the community into fruitless argument." (71) By posting comments that are "intentionally incorrect but not overly controversial," trolls seek entertainment at the expense of the naive and vulnerable. (38; 71) Both flaming and trolling often result in a "disrupt[ion to] the on-going conversation, and both can lead to extended aggravated argument." (71)

From the representational perspective, first there is the "digital divide" that prevents the poor, uneducated, and/or elderly from accessing the Internet and hence participating in online dialogue. Second, because the organization of information is in the hands of private companies and web developers, there is no guarantee that the output of these media are in accordance with the criteria outlined by Dahlgren for political communication; these include "fairness, accuracy, completeness, [and] pluralism of views." (31)

Opinion Space was built to enhance all three dimensions of the public sphere. The system provides a *structure* for visualizing and understanding the spread of opinions

and *interact* with a two-dimensional slice of the public sphere. Although there is not much we can do to give greater access to the site, it enhances the *representational* dimension by giving all viewpoints equal value and opportunity in the Space.

It can be argued that Opinion Space is less of a facilitator of discussion and more of a way to visualize the diversity of opinions. This is certainly true to the extent that we do not (yet) allow users to respond directly to each other's comments. Rather, Opinion Space is more of an abstract discussion where opinions are presented and the most insightful ones are "voted" to the top.

7.2 Reputation in an Online Setting

When it comes to evaluating a textual response in the absence of nonverbal cues, one of the greatest challenges for users of any social media system is to determine who is reliable and who is not. This is even more necessary when the number of responses is so large that it becomes overwhelmingly frustrating to sift through them in search of the most insightful ones.

In designing Opinion Space, our goal was to automate this process in the form of a *reputation system* (118). These systems take as input the history of actions for each participant and output a ranking of the participants based on their trustworthiness; rankings are either global or personalized with respect to a particular individual. There are several challenges in creating such systems:

1. Providing users with an incentive to cooperate and strive to improve their reputation or rank.

2. Aggregating the data to form a fair and nontrivial ranking of the users.

3. Introducing mechanisms that make the system resistant to manipulation by malicious users.

We now discuss each in further detail.

7.2.1 Incentive Structure

We observe that participants would only want to build a good reputation for themselves if there is a promise of future gains as a result. (34; 118) For example, with one-time transactions such as on Craigslist, it is not economically advantageous to take an initial loss in order to build a good reputation. On the other hand, merchants on eBay desire better reputations because it will give consumers more confidence in making a purchase and hence make them more likely to do so. (120)

By design, participants of a system with a meaningful reputation economy are encouraged to contribute constructively in order to improve their reputability and, subsequently, their influence or relevance. Therefore, when it comes to building an effective reputation system for users of large-scale online discussion tools, the *key* challenge is

in designing the proper incentives (i.e. future gains) for building a good reputation; for without this component, the reputation system is rendered useless.

In the version of Opinion Space released with the US Department of State, user participation was entirely self-motivated. That is, no promises were made to participants in reward of good behavior. Participants contributed to the discussion purely out of interest or curiosity. With the Automotive Industry study, on the other hand, we provided monetary incentives for constructive participation. Specifically, we promised to reward the authors of the ten most insightful responses 100 dollar gift certificates. We found that this greatly motivated participants, and they contributed at significantly higher levels and more insightful responses than participants of the State Department version. Further study will be required to accurately characterize the motivational factors for constructive participation, which will help inform the design of future iterations of the site.

7.2.2 Ranking Participants

While removing hierarchical social structures and prejudices from the picture has been shown to encourage individuals to be more forthcoming with their opinions, anonymity on the web exacerbates the problem of information overload. Naturally, as long as pseudonyms are free or cheap, any unsupervised reputation system is vulnerable to *whitewashing*, where participants with bad reputations can get a fresh start by simply taking a few minutes to sign up for a new account. This introduces serious challenges to dealing with malicious participants seeking to sidetrack or even sabotage the discussion. The contradiction posed by the current online culture of free access to information and interaction is that the very people we are trying to filter out of the conversation can easily mask their true identities by creating new ones as necessary. Some websites have combatted this by hiring human moderators or requiring textual responses by new users to be pre-approved, but what we are truly interested in is creating a self-organizing system armed with the proper mechanisms for preventing malicious participants from causing too much damage.

Hence, rather than giving each response a blank slate, we propose a mathematical model for participant reputation that considers the actions and contributions made by each participant over the history of the participant's account. This would afford any new participant, regardless of social class, the opportunity to establish her or himself as a reputable contributor to the discussion while at the same time allowing the system to filter out inappropriate material. Participants with the best reputation would be given the loudest voice in the space, and malicious participants seeking to seriously disrupt the system would first need to invest the time into building a good reputation.

The reasoning behind our model can be described as follows. By giving users the opportunity to rate responses in a discussion forum, we allow for the very likely possibility that users will only promote their own interests and rate opposing opinions poorly, even if it is a well-written and pointed response. We claim that this behavior is of little value to the system, because all it indicates is that participants holding a particular opinion tend to agree with similar participants. (Note that this is also

an indication that our dimensionality reduction method is working well.) Visually, one can imagine that the space is partitioned into subgroups or smaller spheres of agreement. In terms of deliberative democracy, we are interested in emphasizing the responses where these spheres intersect. More specifically, we note that it is surprising (statistically) when a participant agrees with another that has a significantly different opinion profile. In this scenario, we have identified participants of different viewpoints that have potentially found a legitimate middle ground. The spatial reputation model we have designed highlights responses that elicit high levels of approval from a diversity of participants, as opposed to like-minded users.

Under this assumption, the question at hand is to determine what type of responses will propagate towards the top? Will they be interesting and hence more valuable in providing solid ground for debate between subgroups, or will they be the "safe" responses that represent a superficial or obvious consensus among diverse users?

By design, the reputation system filters the massive amount of responses received by highlighting those with the best score and downplaying or even removing the worst ones. Hence, perhaps an even more important question to ask is who is being silenced by this system, and what does it mean for deliberative democracy? Can a public sphere be sustained and nurtured with online discussion tools that implement unsupervised methods for filtering opinions? How does the spiral of silence factor in? Is the phenomenon made worse by reputation / filtering methods? If so, how can we design new reputation models so that this effect is minimized?

7.2.3 Resisting Manipulation

We consider three methods for manipulating the rank of a response in Opinion Space and describe how we propose to handle each form of attack.

7.2.3.1 Whitewashing

With the power of digital communication over the Internet comes anonymity. Because interactions are never face-to-face, it is incredibly easy for people to assume new or false identities without being detected. *Whitewashing* refers to the user's ability to erase a bad reputation by registering for a new account.

This sort of behavior can be mitigated by introducing a cost to entry, also called an "initiation fee." If the cost is sufficiently high, users will not have enough incentive to whitewash their accounts. This solution is not practical in real-world applications, since the culture of the Internet has evolved towards the principles of free and unlimited access to content. (42)

Friedman et al. (51) argue that a better model is to give a reputation penalty to new users, which they call the *Pay Your Dues* (PYD) strategy. Users are partitioned into two categories: veterans are those who have played at least one round of the game, and newcomers are those who haven't yet participated. The PYD strategy is similar to Reputational-Grim in that newcomers cooperate with each other and all users defect against those who deviate from the strategy. The difference is that veterans choose to

defect against newcomers instead of cooperating with. This results in forcing newcomers to pay an initial reputational fee, which they can only gain back by cooperating with the PYD strategy.

It is interesting to note that the PYD strategy maximizes social welfare when whitewashing is present, where social welfare is defined as the sum of the utilities for all participating users. (53) However, this value is strictly less than the total social welfare of the users when whitewashing is not possible, and so we conclude that whitewashing becomes a financial burden to society.

Opinion Space currently does not rely on any protocols for detecting whitewashing. Working with the Department of State imposed severe restrictions on the types of data that we can collect from participants, and we were not allowed to store IP information, even for security and tracking purposes. Consequently, the only cost for creating a new account is time, since the site is entirely free and available to anyone with an email address. In the future, if we find whitewashing to be a problem on the site, we may wish to employ more sophisticated techniques for detecting and denying new accounts to whitewashers. However, on the whole this is a generally unsolved problem in practical Internet applications.

7.2.3.2 False Feedback

For many applications on the web, it is not possible to determine the true outcome of an interaction between two individuals. Instead, we rely on user-reported feedback, which may of course be subject to emotional whims, misjudgment, or even malice. According to Friedman et al. in (51), there are two key challenges in such systems:

1. Giving participants incentive to take the time to formulate and express an opinion.

2. Encouraging users to provide honest feedback.

One solution to these problems could be in the form of a points or reward system. Assuming that rewards are incentive enough to elicit feedback from participants, the main challenge is to introduce an incentive compatible or *truthful* mechanism for distributing rewards, where truthful is defined as follows.

Definition. A reward mechanism is **truthful** or incentive compatible if it is a dominant strategy for every user to provide honest ratings. (105)

The difficulty here is to identify honest feedback when we don't have an objective way to compare participant feedback with an outcome.

The first model that comes to mind is to test for and reward agreement among different participants. So, for example, if the majority of participants agree that the quality of response A is high, then we would reward users who give A positive reviews. Unfortunately, this model has a major flaw in that it gives participants incentive to provide feedback in agreement with the majority. One way to counter this would be to hide the score or reputation of the response until the participant provides a rating.

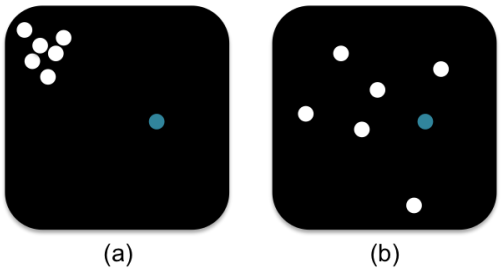


Figure 7.1: Illustration of two possible sybil attacks on Opinion Space. In (a) the attacker seeks to have the greatest impact with each positive rating of the target, and hence creates multiple phantom profiles as far away from the target as possible. In (b), the attacker wishes to minimize the chance of detection, and thus creates multiple phantom profiles at uniformly random locations. The former strategy is easier to detect, while the latter requires significantly more work on behalf of the attacker.

However, this would also prevent participants from using our scoring system to navigate the Space more efficiently.

Eliciting truthful feedback is one of the greatest challenges faced by Opinion Space and discussion forums in general. While in most cases it is impossible to prevent false ratings (52), our spatial ranking model is able to resist it by introducing the following tradeoff: If a participant wants to have the maximum impact when rating similar participants highly, she would be forced to misrepresent her ratings of the ve initial statements; hence, a participant cannot rate the statements truthfully and artificially inate a neighbors ranking at the same time. We make it further difficult for malicious participants to manipulate the rank of a response by only showing participants a randomized subset of the responses at any given time; this serves to make the Space more manageable for the participant, but it also makes it difficult to target a specific response.

7.2.3.3 Phantom Feedback (Sybil Attacks)

The last form of attack on reputation systems that we consider are referred to as sybil attacks. This describes the scenario where a participant creates a large number of fake accounts to provide false feedback that would improve her overall rank or reputation. There is no limit on either the number of fake accounts that a participant can create or the trust values reported in the fake reviews.

Definition. A reputation function F is (value-, rank-) sybilproof if for any participant, no sybil attack can strictly improve the user's (cardinal, ordinal) rank.

Sybilproofness is a very strong requirement on reputation systems. In fact, Cheng and Friedman (25) show that no symmetric (i.e. reliant only on the structure of the graph) reputation function can be rank-sybilproof. Even worse, they show that there cannot exist a reputation function that is sybilproof against just one sybil. From this

we learn that PageRank, being a symmetric reputation function, cannot be sybilproof. Our only hope in finding a sybilproof function is to create an assymetry by preemptively giving participants different levels of importance.

Since our Spatial ranking model is symmetric, we also conclude that it cannot be rank-sybilproof. In any case, let us consider two possible approaches to a sybil attack that seeks to promote the rank of a target response.

1. The sybil attacks are carried out by creating multiple phantom users with the opinion profile that is most distant from the target's profile.

2. The sybil attacks are carried out by creating phantom users with uniformly distributed opinion profiles on the five propositions.

Although phantom participants of the first type will have greater influence on the overall rank of the target, this type of attack is much easier to detect. The second type of attack is nearly impossible to detect without outside information (such as IP addresses, time stamps, etc.), however the phantom participants of this type will have significantly less influence on the rank of the target. Hence, attacks of the first type run a greater risk of being caught, while attacks of the second type require more effort to have the same amount of influence. (See Figure 7.1 for an illustration.

7.2.4 Reputation in Opinion Space: Who is Silenced?

The primary purpose of including a reputation system in Opinion Space is to help participants cope with information overload. Instead of having to sort through all of the responses in search of the most insightful, we use the *wisdom of crowds* to improve efficiency for the participant. The end result is that some individuals become "louder" in the space (bigger and brighter dots, name on the leaderboard), while the opinions of others become "quieter" or even silenced all together. In keeping with the principles of the public sphere and deliberative democracy, our challenge in designing an information filter is to ensure that all opinions are given equal opportunity to be heard; the goal of the reputation system is to propagate the most insightful opinions to the top, regardless of "popularity." That is to say, insightful responses of the unpopular opinion should be just as prominent in the space as insightful comments of the popular opinion.

7.3 Enhancements to Opinion Space in Support of Deliberative Democracy

While it is extremely important to design a system that gives upholds the ideals of the public sphere and deliberative democracy, such a system would be useless if it fails to retain participants. Hence, a delicate compromise must be made between these ideals and the desire of the participants. Munson et al. (100) hypothesize that participants seeking political commentary online fall into one of three categories:

1. Those that are *challenge-averse*: Participants who are not interested in exposing themselves to opinions that challenge their own.

2. *Diversity-seeking* participants, who are primarily interested in finding insightful opinions that challenge their own.

3. And those that are *support-seeking*: participants who claim to be diversity-seeking but in reality prefer to see a diversity of opinions so long as the majority of those opinions support their own.

Under this assumption, it is necessary to create a system that caters to these different personalities, which motivates the design of a recommendation engine that personalizes each participant's experience with the site. The goal is to present each individual with a set of responses of greatest possible diversity such that the participant will want to actively engage with the site.

In this section we propose two enhancements to the Opinion Space framework that work towards this goal. The first is an improvement to the dimensionality reduction technique used to build the Opinion Space map of participants. This method considers the content of the textual responses in addition to the statement ratings to create a more meaningful space; it indirectly serves to support deliberative democracy by giving participants more precise control over the diversity of responses they see.

The second enhancement is a new UI feature, the "Diversity Donut," that allows participants to explicitly control the diversity of responses they see. This is paired with a new recommendation algorithm designed to return insightful responses within the diversity constraints specified by the participant. Giving participants direct control over the diversity of recommended responses takes the guesswork out of catering to different participant personalities and minimizes the risk of frustration on behalf of the participants.

In the remainder of this chapter, we discuss both enhancements in further detail and preliminary results from a pilot study we performed.

7.3.1 Improving the Opinion Space Map with Text Analysis

The existing Opinion Space platform uses Principle Component Analysis (PCA) to project participants onto a two-dimensional plane. This method only utilizes the participants ratings of the initial set of statements and does not consider the textual response in the calculation of a participants position in the space. Canonical Correlation Analysis (CCA), on the other hand, is a statistical technique that uses both the participants statement ratings *and* her textual response to calculate the participants position. This enables us to capture the sentiment of the participants textual response as part of the projection. A participants response is stemmed, featurized, and fed into CCA along with their slider data to determine the participants position.

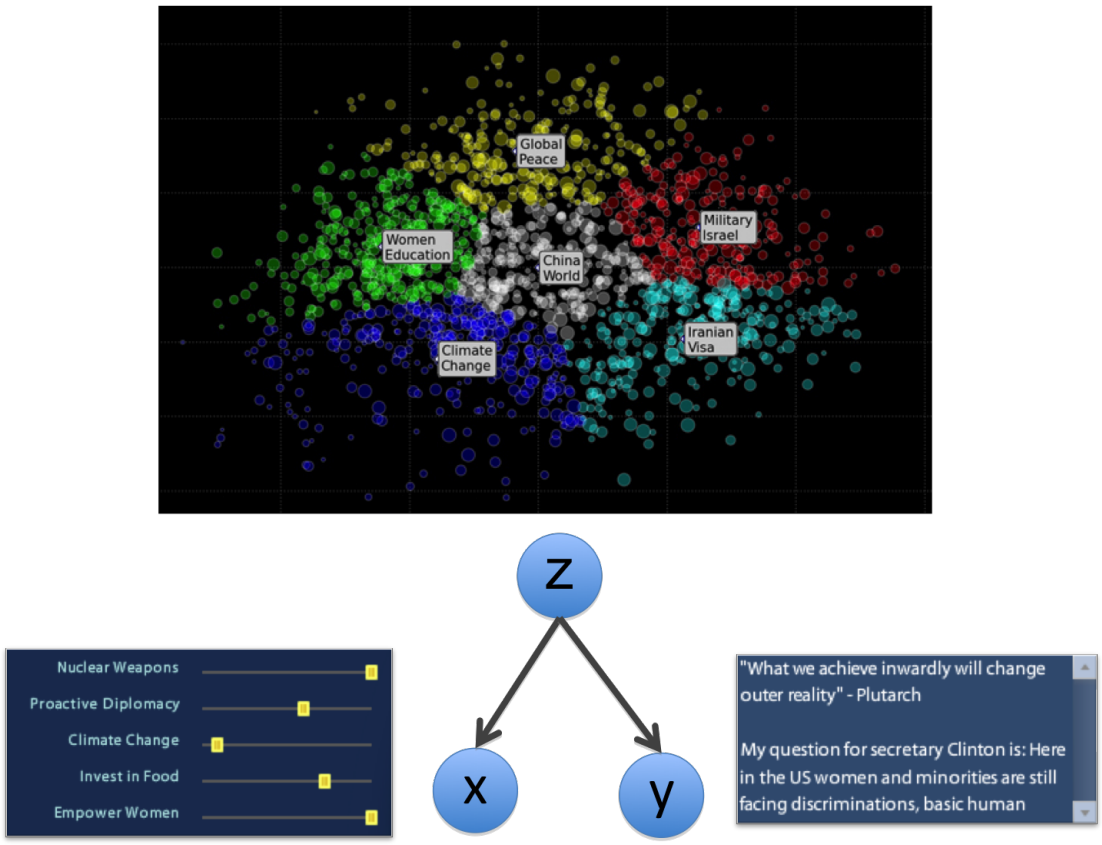


Figure 7.2: Region labeling of the topics of responses in CCA space.

7.3.1.1 Evaluating Projection Quality

We evaluate the quality of a dimensionality reduction method for Opinion Space by calculating the average Pearson correlation between the agreement rating for a response and the distance between the rater and author of the response for all ratings collected by the system. This method gives us a quantifiable measure of the significance of the spatial relationships mapped by different dimensionality reduction techniques. In accordance with the desired properties of the Opinion Space map, technique A is of higher quality than B if the correlation computed in the map produced by A is lower than the correlation for the map produced by B. Our results suggest that CCA is a better projection method than PCA for Opinion Space. (See Section 7.3.3.3.)

7.3.2 The Diversity Donut

The Diversity Donut is a new graphical feature for the Opinion Space interface that enables participants to explicitly indicate their desired level of diversity in the set of recommended responses. The tool consists of two adjustable rings that are centered

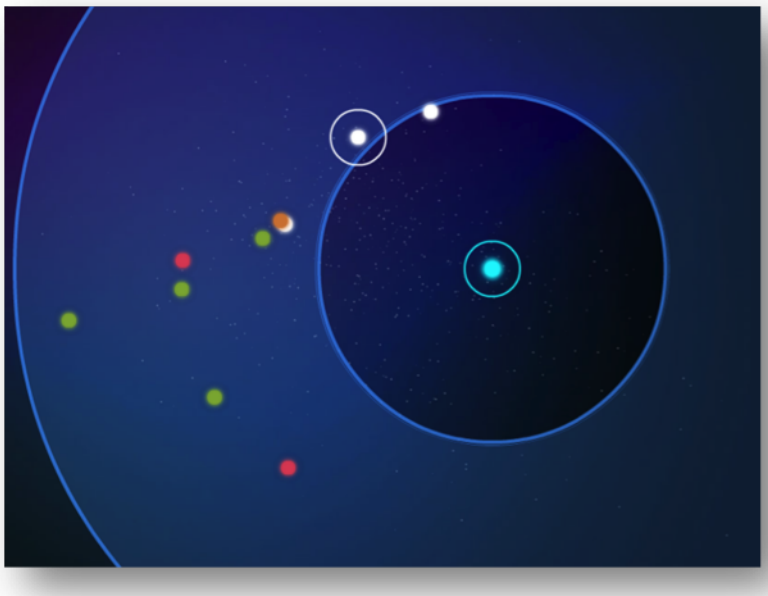


Figure 7.3: The Diversity Donut allows participants to indicate the level of diversity they would like to see in a recommended set of responses. Each point in the space represents a participants response. By changing the radii of the inner and outer rings of the Diversity Donut, a participant changes the degree of diversity in the responses that will be presented.

around the active participant's point in Opinion Space (Figure 7.3). The space defined between the inner and outer radii of the resulting donut-shaped image defines the spatial region that is queried for responses to recommend the participant. After the participant defines the region, the underlying recommendation system retrieves responses from participants whose points fall within that region. Allowing the participant to adjust the query space to suit her preferences gives her direct control over whether she is recommended responses from like-minded participants or from those with differing opinions.

We propose a basic recommendation engine to support the Diversity Donut that seeks to recommend participants a diverse set of responses given the spatial constraints defined by the Donut. This means that the algorithm may recommend a response with which the participant disagrees but may still find insightful. This approach differs from traditional collaborative filtering and recommendation systems (e.g. Amazon.com and Netflix), which recommend participants a set of items that they most likely would agree with or rate highly. Arguably, the application of a traditional recommender algorithm in Opinion Space would encourage cyberpolarization: if our goal was to recommend responses that participants are most likely to rate highly, then participants would only see responses that reinforce their own opinions.

Figure 7.4: Each participant in the study evaluated responses recommended by the three recommendation methods illustrated above: the spatial ranking method (a non-personalized approach), k Nearest Neighbors, and using the Diversity Donut.

The recommendation algorithm we propose first clusters all participants in CCA space using k-means clustering. For every cluster of participants, the system aggregates each participants agreement ratings and calculates the average agreement rating for all responses in the system. After a participant defines a spatial search region with the donut, the system returns the responses that have the lowest average agreement in the cluster and are within the search region.

7.3.3 System Evaluation

We ran a pilot study to build a preliminary understanding of how participants interact with the Diversity Donut feature. In this section we describe the design and results of that study.

7.3.3.1 User Study Design

To evaluate the Diversity Donut, we selected a set of 118 thoughtful responses that reflect a range of opinions from Question 2 of the Opinion Space 2.0 data set. We conducted a controlled user study with 13 participants. Five participants were Berkeley students and eight were recruited from Mechanical Turk. An experimenter was available online for any questions that the Mechanical Turk participants may have had. Before starting the experiment, participants completed a tutorial that used both images and text to explain the Opinion Space interface.

Participants were first asked to fill out a prescreening survey to determine whether they were challenge averse or diversity seeking. Next, they were asked to complete their opinion profile. As with the original version of Opinion Space, participants were asked to rate the five initial statements using a sliding scale that ranged from 'Strongly Disagree' to 'Strongly Agree' and to enter a textual response to a discussion question.

Each participant was then presented with three different (though not necessarily disjoint) sets of 10 or more recommended responses in a random, sequential order. Each set of recommendations was generated using a different algorithm: a) the Global

Spatial method described in Section 6.5, b) the k Nearest Neighbors (KNN) algorithm described in Section 4.1.1, and c) the Diversity Donut approach. (See Figure 7.4.) The KNN model is a benchmark recommendation system that makes personalized recommendations based on the preferences of those in the immediate neighborhood of the participant. Participants were asked to read the responses recommended from each method and to rate each response based on how much they agreed with it and how insightful they found it. After reviewing the responses returned by each recommendation method, the participants were asked to complete a survey regarding the quality of the responses and the participants satisfaction. At the end of the experiment, they were also asked to complete an exit survey.

All participant activity on the space was recorded, including the responses viewed, the agreement and insight rating values provided, the dwell time per response, and the radii of the Diversity Donuts inner and outer rings adjusted by each participant.

7.3.3.2 Hypotheses

We designed the pilot study to evaluate the following three hypotheses.

> *Hypothesis 1 (H1): Dimensionality Reduction Techniques.* CCA will outperform PCA as a dimensionality reduction technique according to the method of evaluation described in Section 7.3.1.
>
> *Hypothesis 2A (H2A): Insightfulness of Responses.* Participants will find the responses recommended by the Diversity Donut to be more insightful than those recommended by KNN.
>
> *Hypothesis 2B (H2B): Agreement with Responses.* Participants will agree more with the responses recommended by the KNN model than with those recommended by the Diversity Donut.
>
> *Hypothesis 3 (H3): Self-reported Data.* Participants will report that the responses recommended by the global ranking method and the Diversity Donut will be more diverse than those recommended by KNN and that there will be higher satisfaction in using the Diversity Donut.

7.3.3.3 Preliminary Results

In this section we provide the preliminary results collected with our pilot study, both in terms of the quality of the CCA projection and participant response to the Diversity Donut recommendation feature.

Projection Quality

We found that CCA had the greatest negative correlation between spatial distance and agreement rating of responses. This indicates that the spatial relationships between participants when using the CCA projection is more meaningful in terms of difference or similarity of opinion than with the PCA or a randomized projection. We therefore

Method	ρ
CCA	-0.352
PCA	-0.134
RND	0.0021

Table 7.1: Pearson's correlation between distance and agreement rating as measured by projections using CCA, PCA, and random placement in the space. We find that CCA provides the highest negative correlation, and so of the three methods it is best suited for visualizing the spread opinions in Opinion Space.

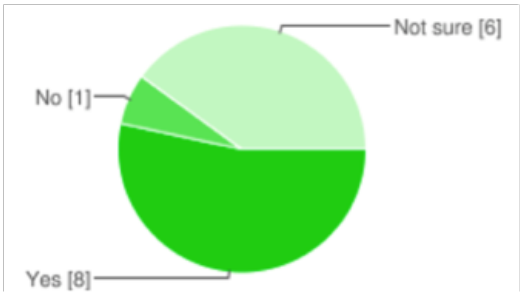


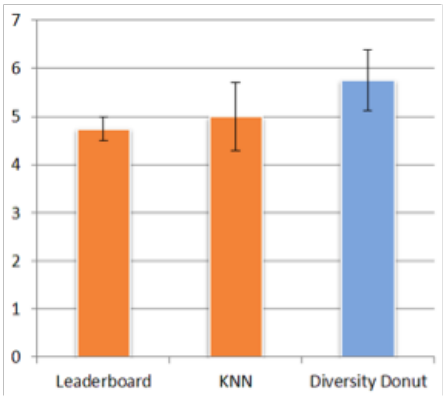
Figure 7.5: Responses to the survey question "Do you prefer the diversity donut to a purely automated approach?"

conclude that Hypothesis 1 is supported and CCA is the most effective dimensionality reduction method for Opinion Space out of those considered. Table 7.1 summarizes the results for our evaluation.

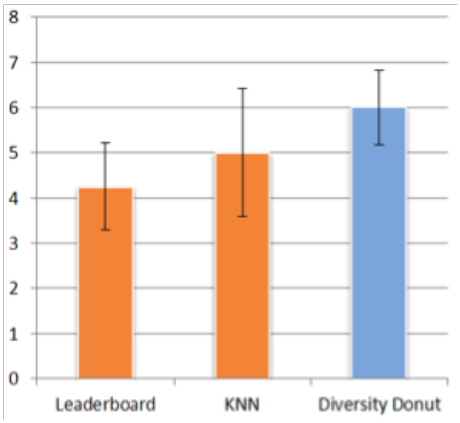
Diversity Donut

We performed a single factor ANOVA analysis with repeated measures on the insightfulness ratings ($p = 0.2167$) and agreement ratings ($p = 0.2047$) across all three recommendation methods. The results from this analysis were inconclusive in determining which recommendation method yields the highest agreement or insightfulness ratings. There are three possible explanations for this result: 1) 13 participants is not a sufficiently large sample to establish statistical significance, 2) there is no significant difference between the three algorithms in terms of the mean values of insightfulness and agreement ratings, or 3) the dataset of responses used was not sufficiently diverse in content and opinion. All of these possibilities require further statistical analysis. Hypotheses 2A and 2B cannot be supported based on this preliminary data.

Figures 7.5, 7.6a, and 7.6b show the self-reported data from the surveys. Figures 7.6a, and 7.6b illustrate the mean value and one standard deviation in each direction for the Likert scale data. These data suggest that participants generally preferred the Diversity Donut to an entirely automated approach. Additionally participants were more satisfied with the diversity of responses that were presented to them by the Diversity Donut, suggesting support for Hypothesis 3.



(a) Satisfaction with Diversity



(b) Perceived Diversity

Figure 7.6: Responses to the survey questions (a) "How satisfied were you with the diversity of the opinions expressed in the recommended set of responses?" and (b) "Did you see a good range of opinions in the recommended responses?"

7.3.3.4 Discussion

In this preliminary report, we compare CCA with PCA and investigate the effectiveness of the Diversity Donut as a personalized tool for finding a diversity of insightful responses. Our analysis of projection quality suggests that CCA is a more effective dimensionality reduction method for Opinion Space. While our data on the Diversity Donut is inconclusive, self-reported data suggests that participants found the Diversity Donut to be an effective tool for recommending diverse responses.

The data set of responses may not have been sufficiently diverse in content and opinion. In the next iteration of our study, we plan to it again using a data set of responses that has a wider range of diversity in content, possibly with a more controversial discussion question.

The inner and outer rings of the Diversity Donut allow a participant to select a

symmetric search region, but the CCA space is not symmetric around the participant. This suggests that a Diversity Lasso that is not centered on the participants point may be a more useful design. Since CCA provides us with strong topic modeling and region labeling capability, we can use this data to augment the space with topic labels, in which participants can see the main topic for each region in space. Participants can then use the lasso to select a region that they are interested in. We would like to analyze how more general tools such as the lasso perform in comparison to symmetric tools like the Diversity Donut.

References

[1] R.K. Ahuja, D.S. Hochbaum, and J.B. Orlin. Solving the convex cost integer dual network flow problem. *Management Science*, 49(7):950–964, 2003. 114

[2] K. Ali and W. van Stam. TiVo: making show recommendations using a distributed collaborative filtering architecture. *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 394–401, 2004. 68

[3] A. Altman and M. Tennenholtz. Axiomatic foundations for ranking systems. *Journal of Artificial Intelligence Research*, 31:473–495, 2008. 109

[4] Amelie Anglade, Marco Tiemann, and Fabio Vignoli. Complex-network theoretic clustering for identifying groups of similar listeners in p2p systems. In *RecSys '07: Proceedings of the 2007 ACM conference on Recommender systems*, pages 41–48, New York, NY, USA, 2007. ACM. 68

[5] K.J. Arrow. *Social Choice and Individual Values*. Yale University Press, 1970. 109, 110

[6] M. Baum and K. Passino. A search-theoretic approach to cooperative control for uninhabited air vehicles. *Proc. of the 2002 AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2002. 11

[7] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995. 35

[8] A.J. Berinsky. The two faces of public opinion. *American Journal of Political Science*, 43(4):1209–1230, 1999. 150

[9] J. Bishop. Increasing participation in online communities: A framework for human-computer interaction. *Computers in Human Behavior*, 23(4):1881–1893, 2007. 150

[10] E. Bitton. A spatial model for collaborative filtering of comments in an online discussion forum. In *Proceedings of the third ACM conference on Recommender systems*, pages 393–396. ACM, 2009. 93, 104, 125

[11] M. Booth. How do computers know so much about us? *The Denver Post*, page F01, January 30 2005. 54

[12] F. Bourgault, T. Furukawa, and HF Durrant-Whyte. Coordinated decentralized search for a lost target in a bayesian world. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1, 2003. 12

[13] H. E. Brady. Dimensional analysis of ranking data. *American Journal of Political Science*, 34(4):1017–1048, 1990. 89

[14] P.B. Brandtzaeg and J. Heim. Explaining participation in online communities. *Handbook of Research on Socio-Technical Design and Social Networking Systems*, page 167, 2009. 150

[15] J.S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. *Learning*, 9:309–347, 1992. 75

[16] Derek Bridge and Francesco Ricci. Supporting product selection with query editing recommendations. In *RecSys '07: Proceedings of the 2007 ACM conference on Recommender systems*, pages 65–72, New York, NY, USA, 2007. ACM. 69

[17] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998. 114

[18] Kenneth Bryan, Michael O'Mahony, and Pádraig Cunningham. Unsupervised retrieval of attack profiles in collaborative recommender systems. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 155–162, New York, NY, USA, 2008. ACM. 77

[19] L. Caffarelli, V. Crespi, G. Cybenko, I. Gamba, and D. Rus. Stochastic Distributed Algorithms for Target Surveillance. *Intelligent Systems Design and Applications*, 2003. 12

[20] D.J. Cannon. *Point-and-direct telerobotics: object level strategic supervisory control in unstructured interactive human-machine system environments*. PhD thesis, Stanford University Department of Mechanical Engineering, 1992. 11

[21] P.J. Carrington, J. Scott, and S. Wasserman. *Models and methods in social network analysis*. Cambridge University Press, 2005. 90

[22] G. Casella and R.L. Berger. *Statistical inference*. Duxbury Press, 2 edition, 2001. 133, 134

[23] JL Casper and RR Murphy. Workflow study on human-robot interaction in USAR. *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, 2, 2002. 11

[24] L. Chaimowicz and V. Kumar. Aerial shepherds: Coordination among uavs and swarms of robots. *Proceedings of the 7th International Symposium on Distributed Autonomous Robotic Systems (DARS 2004*, pages 231–240, 2004. 11

[25] A. Cheng and E. Friedman. Sybilproof reputation mechanisms. In *Proceedings of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, pages 128–132. ACM New York, NY, USA, 2005. 108, 157

[26] WJ Conover and R.L. Iman. Rank transformations as a bridge between parametric and nonparametric statistics. *American Statistician*, 35(3):124–129, 1981. 98

[27] D. Cosley, S.K. Lam, I. Albert, J.A. Konstan, and J. Riedl. Is seeing believing?: how recommender system interfaces affect users' opinions. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 585–592. ACM New York, NY, USA, 2003. 57, 69

[28] M.A.A. Cox. *Multidimensional scaling*. CRC Press, 2000. 89, 90

[29] A. Dahl. Implementation of a collaborative observatory for natural environments. Master's thesis, University of California at Berkeley, 2007. 27

[30] L. Dahlberg. Computer-mediated communication and the public sphere: A critical analysis. *Journal of Computer-Mediated Communication*, 7(1):27, 2001. 150, 151

REFERENCES

[31] P. Dahlgren. The Internet, public spheres, and political communication: Dispersion and deliberation. *Political Communication*, 22(2):147–162, 2005. 150, 151, 152

[32] A.S. Das, M. Datar, A. Garg, and S. Rajaram. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web*, pages 271–280. ACM, 2007. 52

[33] C. Dellarocas. The digitization of word of mouth: Promise and challenges of online feedback mechanisms. *Management Science*, pages 1407–1424, 2003. 108

[34] C. Dellarocas. The digitization of word of mouth: Promise and challenges of online feedback mechanisms. *Management Science*, 49(10):1407–1424, 2003. 153

[35] M. Deshpande and G. Karypis. Item-based top-N recommendation algorithms. *ACM TOIS*, 22(1):143–177, 2004. 57

[36] I.S. Dhillon, Y. Guan, and B. Kulis. Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 551–556. ACM, 2004. 38

[37] C. Ding, X. He, P. Husbands, H. Zha, and H.D. Simon. PageRank, HITS and a unified framework for link analysis. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 353–354. ACM New York, NY, USA, 2002. 111

[38] J.S. Donath. Identity and deception in the virtual community. *Communities in cyberspace*, pages 29–59, 1999. 152

[39] S. Faridani, E. Bitton, K. Ryokai, and K. Goldberg. Opinion space: a scalable tool for browsing online comments. In *Proceedings of the 28th international conference on Human factors in computing systems*, pages 1175–1184. ACM, 2010. 91

[40] S. Faridani, B. Lee, S. Glasscock, J. Rappole, D. Song, and K. Goldberg. A networked telerobotic observatory for collaborative remote observation of avian activity and range change. *IFAC Conference on Networked Robotics (NetRob)*, 2009. 11

[41] U. Feige, S. Goldwasser, L. Lovasz, S. Safra, and M. Szegedy. Approximating clique is almost NP-complete. In *Foundations of Computer Science, 1991. Proceedings., 32nd Annual Symposium on*, pages 2–12. IEEE, 2002. 21

[42] M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica. Free-riding and whitewashing in peer-to-peer systems. *IEEE Journal on Selected Areas in Communications*, 24(5):1010–1019, 2006. 155

[43] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta. A survey of kernel and spectral methods for clustering. *Pattern Recognition*, 41(1):176–190, 2008. 38

[44] J.S. Fishkin. *Democracy and Deliberation: New Directions for Democratic Reform*. Yale University Press, 1991. 83

[45] J.S. Fishkin and R.C. Luskin. Experimenting with a democratic ideal: Deliberative polling and public opinion. *Acta Politica*, 40(3):284–298, 2005. 83

[46] J.S. Fishkin, R.C. Luskin, and R. Jowell. Deliberative polling and public consultation. *Parliamentary Affairs*, 53(4):657–666, 2000. 83, 149

[47] C. Fowlkes, Q. Shan, S. Belongie, and J. Malik. Extracting global structure from gene expression profiles. *Methods of Microarray Data Analysis II*, pages 81–90, 2002. 38

[48] J.B. Frazier. Weather Stymies Mount Hood Search. *The Associated Press*, December 2006. 8

[49] L.C. Freeman. Visualizing Social Networks. *Journal of social structure*, 1(1):4, 2000. 90

[50] Y. Freund and R. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory*, pages 23–37. Springer, 1995. 147

[51] E. Friedman, P. Resnick, and R. Sami. Manipulation-Resistant Reputation Systems. In N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, pages 677–697. Cambridge University Press, 2007. 109, 116, 155, 156

[52] E. Friedman, P. Resnick, and R. Sami. Manipulation-resistant reputation systems. *Algorithmic Game Theory*, pages 677–697, 2007. 157

[53] E.J. Friedman and P. Resnick. The Social Cost of Cheap Pseudonyms. *Journal of Economics and Management Strategy*, 10:173–199, 2001. 107, 156

[54] S. Friess. Searching by Land, Air and the Web. *The New York Times*, September 2007. 9

[55] T. Furukawa, F. Bourgault, B. Lavis, and H.F. Durrant-Whyte. Recursive Bayesian Search-and-Tracking Using Coordinated UAVs for Lost Targets. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2521–2526, 2006. 12

[56] T. George and S. Merugu. A scalable collaborative filtering framework based on co-clustering. *Proc. of the 5th IEEE Int'l Conf. on Data Mining*, pages 625–628, 2005. 57

[57] D. Goldberg, D. Nichols, B.M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992. 54

[58] K. Goldberg. *The Robot in the Garden: Telerobotics and Telepistemology in the Age of the Internet*. The MIT Press, 2001. 9

[59] K. Goldberg and B. Chen. Collaborative control of robot motion: Robustness to error. *International Conference on Intelligent Robots and Systems (IROS)*, 2:655–660, 2001. 11

[60] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste: A Constant Time Collaborative Filtering Algorithm. *Information Retrieval*, 4(2):133–151, 2001. 51, 55, 72, 104

[61] K. Goldberg, J. Santarromana, G. Bekey, S. Gentner, R. Morris, J. Wiegley, and E. Berger. The Telegarden. In *Proceedings of ACM SIGGRAPH*, pages 135–1140, 1995. 9

[62] K. Goldberg, D. Song, and A. Levandowski. Collaborative teleoperation using networked spatial dynamic voting. *Proceedings of the IEEE*, 91(3):430–439, 2003. 10

[63] O. Goldschmidt and D.S. Hochbaum. A polynomial algorithm for the k-cut problem for fixed k. *Mathematics of Operations Research*, 19(1):24–37, 1994. 33

[64] GuideStar. Analyze nonprofit data: Get nonprofit data the way you want it. `http://www2.guidestar.org/Home.aspx`, July 2009. 67

[65] J. Habermas. The theory of communicative action, Vol. I. *Boston: Beacon Press*, 1984. 151

[66] J. Habermas. The public sphere: An encyclopedia article. *Media and cultural studies: keyworks*, page 102, 2001. 151

[67] Tarik Hadzic and Barry O'Sullivan. Critique graphs for catalogue navigation. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 115–122, New York, NY, USA, 2008. ACM. 69

[68] H.H. Harman. *Modern factor analysis*. University of Chicago Press, 1976. 89

[69] J. Heer and D. Boyd. Vizster: Visualizing Online Social Networks. In *Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization*, page 5. IEEE Computer Society, 2005. 90

[70] J.L. Herlocker, J.A. Konstan, L.G. Terveen, and J.T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004. 53, 54, 55, 57

[71] S. Herring, K. Job-Sluder, R. Scheckler, and S. Barab. Searching for safety online: Managing" trolling" in a feminist forum. *The Information Society*, 18(5):371–384, 2002. 152

[72] D.S. Hochbaum. The pseudoflow algorithm: A new algorithm for the maximum flow problem. *Operations Research*, 56(4):992–1009, 2008. 33

[73] D.S. Hochbaum. Polynomial Time Algorithms for Ratio Regions and a Variant of Normalized Cut. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(5):889–898, 2010. 33, 34

[74] D.S. Hochbaum and A. Levin. Methodologies and Algorithms for Group-Rankings Decision. *Management Science*, 52(9):1394, 2006. 109, 111, 113

[75] G.M. Hoffmann, S.L. Waslander, and C.J. Tomlin. Mutual Information Methods with Particle Filters for Mobile Sensor Network Control. *45th IEEE Conference on Decision and Control*, pages 1019–1024, 2006. 12

[76] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1):177–196, 2001. 77

[77] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979. 35

[78] H. Hu, X. Yan, Y. Huang, J. Han, and X.J. Zhou. Mining coherent dense subgraphs across massive biological networks for functional discovery. *Bioinformatics*, 21(Suppl 1):i213, 2005. 38

[79] J. Hu, M. Prandini, and C. Tomlin. Conjugate points in formation constrained optimal multi-agent coordination: A case study. *SIAM J. Control and Optimization*, 2006. 11

[80] EC Huskisson. Visual analogue scales. *Pain measurement and assessment*, pages 33–37, 1983. 80

[81] IT Jolliffe. *Principal component analysis*. Springer Verlag, 2002. 88

[82] A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, 2007. 107, 109, 111, 112, 114

[83] S. Jung, D. Unutmaz, P. Wong, G.I. Sano, K. De los Santos, T. Sparwasser, S. Wu, S. Vuthoori, K. Ko, F. Zavala, et al. In vivo depletion of CD11c+ dendritic cells abrogates priming of CD8+ T cells by exogenous cell-associated antigens. *Immunity*, 17(2):211–220, 2002. 30

[84] R. Kannan, S. Vempala, and A. Vetta. On clusterings: Good, bad and spectral. *Journal of the ACM (JACM)*, 51(3):497–515, 2004. 37

[85] M.G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1-2):81, 1938. 33

[86] D. Kim and B.J. Yum. Collaborative filtering based on iterative principal component analysis. *Expert Systems with Applications*, 28(4):823–830, 2005. 57

[87] J. Konstan, S. McNee, C.N. Ziegler, R. Torres, N. Kapoor, and J. Riedl. Lessons on Applying Automated Recommender Systems to Information-Seeking Tasks. *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, 2006. 56, 57

[88] S. Kullback and RA Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951. 17

[89] S. Kullback and R.A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, pages 79–86, 1951. 69

[90] L.I. Kuncheva and C.J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2):181–207, 2003. 109

[91] C.A.C. Lampe, E. Johnston, and P. Resnick. Follow the reader: filtering comments on slashdot. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1253–1262. ACM New York, NY, USA, 2007. 117

[92] D. Lemire. Scale and translation invariant collaborative filtering systems. *Information Retrieval*, 8(1):129–150, 2005. 57

[93] G. Linden, B. Smith, and J. York. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 2003. 57

[94] P.J. Ludford, D. Cosley, D. Frankowski, and L. Terveen. Think different: increasing online community participation using uniqueness and group dissimilarity. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 631–638, 2004. 100

[95] M. McDonald, D. Small, C. Graves, and D. Cannon. Virtual collaborative control to improve intelligent robotic system efficiency and quality. *IEEE International Conference on Robotics and Automation*, 1:418–424, 1997. 11

[96] L. McGinty and B. Smyth. Comparison-Based Recommendation. *Proceedings of the 6th European Conference on Advances in Case-Based Reasoning*, pages 575–589, 2002. 69

[97] S. McKillup. *Statistics explained: an introductory guide for life scientists*. Cambridge Univ Pr, 2006. 97, 98

[98] K. Miller. *Communication theories: Perspectives, processes, and contexts*. McGraw-Hill New York, 2005. 150

[99] Bamshad Mobasher, Robin Burke, Runa Bhaumik, and Chad Williams. Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. *ACM Trans. Internet Technol.*, 7(4):23, 2007. 77

[100] S.A. Munson and P. Resnick. Presenting diverse political opinions: how and how much. In *Proceedings of the 28th international conference on Human factors in computing systems*, pages 1457–1466. ACM, 2010. 158

REFERENCES

[101] RR Murphy. Human-robot interaction in rescue robotics. *Systems, Man and Cybernetics, Part C, IEEE Transactions on*, 34(2):138–153, 2004. 11

[102] RR Murphy, CL Lisetti, R. Tardif, L. Irish, and A. Gage. Emotion-based control of cooperating heterogeneous mobile robots. *Robotics and Automation, IEEE Transactions on*, 18(5):744–757, 2002. 11

[103] H. Nagamochi and T. Ibaraki. Computing edge-connectivity in multigraphs and capacitated graphs. *SIAM Journal on Discrete Mathematics*, 5(1):54–66, 1992. 33

[104] M. Narayanan, A. Vetta, EE Schadt, and J. Zhu. Simultaneous Clustering of Multiple Gene Expression and Physical Interaction Datasets. *Public Library of Science: Computational Biology*, 6(4):e1000742, 2010. 37, 38

[105] N. Nisan. Introduction to Mechanism Design (for Computer Scientists). In N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, pages 209–241. Cambridge University Press, 2007. 156

[106] E. Noelle-Neumann. The Spiral of Silence: A Theory of Public Opinion. *Journal of Communication*, 24(2):43–51, 1974. 150

[107] S.T. Park, D. Pennock, O. Madani, N. Good, and D. DeCoste. Naïve filterbots for robust cold-start recommendations. *Proc. of the 12th ACM SIGKDD Int'l Conf.*, pages 699–705, 2006. 57

[108] D.M. Pennock, E. Horvitz, S. Lawrence, and C.L. Giles. Collaborative filtering by personality diagnosis: a hybrid memory and model-based approach. *Proc. of the 16th Conf. on Uncertainty in Artificial Intelligence*, pages 473–480, 2000. 57

[109] A. Perer and B. Shneiderman. Balancing systematic and flexible exploration of social networks. *IEEE Transactions on Visualization and Computer Graphics*, pages 693–700, 2006. 90

[110] V. Plaks, T. Birnberg, T. Berkutzki, S. Sela, A. BenYashar, V. Kalchenko, G. Mor, E. Keshet, N. Dekel, M. Neeman, et al. Uterine DCs are crucial for decidua formation during embryo implantation in mice. *The Journal of Clinical Investigation*, 118(12):3954, 2008. 30, 40, 41

[111] R. Polikar. Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, 6(3):21–45, 2006. 109, 147

[112] M. Poster. Cyberdemocracy: Internet and the public sphere. *Reading digital culture*, pages 259–271, 2001. 150

[113] T.K. Quan, I. Fuyuki, and H. Shinichi. Improving accuracy of recommender system by clustering items based on stability of user similarity. *IAWTIC'2006 Proc.*, pages 61–61, 2006. 57

[114] A.M. Rashid, I. Albert, D. Cosley, S.K. Lam, S.M. McNee, J.A. Konstan, and J.T. Riedl. Getting to know you: learning new user preferences in recommender systems. *Proc. of the 7th Int'l Conf. on Intelligent User Interfaces*, pages 127–134, 2002. 57

[115] A.M. Rashid, S.K. Lam, G. Karypis, and J.T. Riedl. ClustKNN: a highly scalable hybrid model-& memory-based CF algorithm. *Proc. of WebKDD 2006*, 2006. 57

[116] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. GroupLens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, pages 175–186. ACM, 1994. 54

[117] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman. Reputation systems. *Communications of the ACM*, 43(12):45–48, 2000. 108

[118] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman. Reputation systems. *Communications of the ACM*, 43(12):48, 2000. 153

[119] P. Resnick and R. Zeckhauser. Trust among strangers in Internet transactions: Empirical analysis of eBay's reputation system. *Advances in Applied Microeconomics: A Research Annual*, 11:127–157, 2002. 108

[120] P. Resnick and R. Zeckhauser. Trust among strangers in Internet transactions: Empirical analysis of eBay's reputation system. *Advances in Applied Microeconomics: A Research Annual*, 11:127–157, 2002. 153

[121] Paul Resnick and Rahul Sami. The information cost of manipulation-resistance in recommender systems. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 147–154, New York, NY, USA, 2008. ACM. 77

[122] L. Rourke and H. Kanuka. Barriers to online critical discourse. *International Journal of Computer-Supported Collaborative Learning*, 2(1):105–126, 2007. 150

[123] A. Ryan and JK Hedrick. A mode-switching path planner for UAV-assisted search and rescue. *IEEE Conference on Decision and Control and the European Control Conference*, pages 1471–1476, 2005. 11

[124] A.D. Ryan, D.L. Nguyen, and J.K. Hedrick. Hybrid control for uav-assisted search and rescue. *the 2005 International Mechanical Engineering Congress and Exposition, Orlando, FL*, 2005. 11

[125] W. Sack. Conversation map: An interface for very large-scale conversations. *Journal of Management Information Systems*, 17(3):73–92, 2000. 90, 150

[126] A.I. Schein, A. Popescul, L.H. Ungar, and D.M. Pennock. Methods and metrics for cold-start recommendations. *Proc. of the 25th Annual Int'l ACM SIGIR Conf.*, pages 253–260, 2002. 57

[127] B. Schlkopf, A. Smola, and K. R. Mller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998. 88

[128] CE Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, 27:379–423, 1948. 17

[129] R. Sharan, A. Maron-Katz, and R. Shamir. CLICK and EXPANDER: a system for clustering and visualizing gene expression data. *Bioinformatics*, 19(14):1787, 2003. 35

[130] R. Sharan and R. Shamir. CLICK: A clustering algorithm with applications to gene expression analysis. *Proceedings of the International Conference on Intelligent Systems for Molecular Biology*, 8:307–316, 2000. 35

[131] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. 34, 38

[132] J. Shlens. A tutorial on principal component analysis. *Systems Neurobiology Laboratory, University of California at San Diego*, 2005. 82

[133] G. Smyth. Limma: linear models for microarray data. *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*, pages 397–420, 2005. 42

[134] G.K. Smyth. Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, 3(1):1027, 2004. 35

[135] D. Song and K. Goldberg. Sharecam part i: Interface, system architecture, and implementation of a collaboratively controlled robotic webcam. *IEEE/RSJ International Conference on Intelligent Robots (IROS)*, 2003. 10

[136] D. Song and K. Goldberg. Approximate Algorithms for a Collaboratively Controlled Robotic Camera. *IEEE Transactions on Robotics*, 23(5):1016–1070, 2007. 10, 18, 19

[137] D. Song, A.F. van der Stappen, and K. Goldberg. Exact Algorithms for Automated Satellite Frame Selection. *IEEE Transactions on Automation Science and Engineering*, 3(1):16–28, 2006. 10, 18, 19

[138] J. V. Stone. *Independent component analysis*. MIT Press, 2004. 89

[139] J. Stromer-Galley. New voices in the public sphere: A comparative analysis of interpersonal and online political talk. *Javnost - The Public*, 9(2):23–42, 2002. 150

[140] C.R. Sunstein. *Republic. com 2.0*. Princeton Univ Pr, 2007. 80

[141] Y. Takane, F.W. Young, and J. De Leeuw. Nonmetric individual differences multidimensional scaling: an alternating least squares method with optimal scaling features. *Psychometrika*, 42(1):7–67, 1977. 90

[142] Adi L. Tarca, Roberto Romero, and Sorin Draghici. Analysis of microarray experiments of gene expression profiling. *American Journal of Obstetrics and Gynecology*, 195(2):373–88, 8 2006. 35

[143] M.J.W. Thomas. Learning within incoherent structures: the space of online discussion forums. *Journal of Computer Assisted Learning*, 18(3):351–366, 2002. 150

[144] D. Verma and M. Meila. Comparison of spectral clustering methods. *Advances in Neural Information Processing Systems (NIPS 15)*, 2003. 38

[145] Paolo Viappiani, Pearl Pu, and Boi Faltings. Conversational recommenders with adaptive suggestions. In *RecSys '07: Proceedings of the 2007 ACM conference on Recommender systems*, pages 89–96, New York, NY, USA, 2007. ACM. 69

[146] F.B. Viégas and J. Donath. Social network visualization: Can we go beyond the graph. In *Workshop on Social Networks, CSCW*, volume 4, pages 6–10. Citeseer, 2004. 90

[147] L. Von Ahn. Games with a purpose. *Computer*, 39(6):92–94, 2006. 87

[148] L. Von Ahn, R. Liu, and M. Blum. Peekaboom: a game for locating objects in images. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 55–64. ACM, 2006. 87

[149] M. Vozalis and K. Margaritis. On the combination of user-based and item-based collaborative filtering. *International Journal of Computer Mathematics*, 81(9):1077–1096, 2004. 57

[150] J. Wang, A.P. de Vries, and M.J.T. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. *Proc. of the 29th Annual Int'l ACM SIGIR Conf.*, pages 501–508, 2006. 57

[151] J. Wang and J. Zhu. Portfolio theory of information retrieval. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 115–122. ACM, 2009. 68

[152] DC Wilson, B. Smyth, and DO Sullivan. Sparsity reduction in collaborative recommendation: a case-based approach. *Int'l Journal of Pattern Recognition and Artificial Intelligence*, 17(5):863–884, 2003. 57

[153] Hu Wu, Yongji Wang, and Xiang Cheng. Incremental probabilistic latent semantic analysis for automatic question recommendation. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 99–106, New York, NY, USA, 2008. ACM. 69

[154] S. Xiaoyuan and M. Taghi. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009, 2009. 53

[155] E.P. Xing and R.M. Karp. CLIFF: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. *Bioinformatics*, 17(Suppl 1):S306, 2001. 38

[156] Y. Xu, V. Olman, and D. Xu. Clustering gene expression data using a graph-theoretic approach: an application of minimum spanning trees. *Bioinformatics*, 18(4):536, 2002. 37

[157] Y. Xu, D. Song, J. Yi, and A.F. van der Stappen. An Approximation Algorithm for the Least Overlapping p-Frame Problem with Non-Partial Coverage for Networked Robotic Cameras. *IEEE International Conference on Robotics and Automation (ICRA)*, 2008. 11

[158] G.R. Xue, C. Lin, Q. Yang, W.S. Xi, H.J. Zeng, Y. Yu, and Z. Chen. Scalable collaborative filtering using cluster-based smoothing. *Proc. of the 28th Annual Int'l ACM SIGIR Conf.*, pages 114–121, 2005. 57

[159] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H.G. Okuno. An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(2):435–447, 2008. 68

[160] Mi Zhang and Neil Hurley. Avoiding monotony: improving the diversity of recommendation lists. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 123–130, New York, NY, USA, 2008. ACM. 68

[161] C.N. Ziegler, S.M. McNee, J.A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. *Proceedings of the 14th international conference on World Wide Web*, pages 22–32, 2005. 56, 57, 68