Title
Models and Algorithms for Crowdsourcing Discovery

Author
Faridani, Siamak

Publication Date
2012

Peer reviewed|Thesis/dissertation

Models and Algorithms for Crowdsourcing Discovery

by

Siamak Faridani

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

in

Engineering — Industrial Engineering and Operations Research

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Ken Goldberg, Chair
Professor Ilan Adler
Professor Laurent El Ghaoui
Professor Bjorn Hartmann

Fall 2012

Models and Algorithms for Crowdsourcing Discovery

Abstract

Models and Algorithms for Crowdsourcing Discovery

by

Siamak Faridani

Doctor of Philosophy in Engineering — Industrial Engineering and Operations Research

University of California, Berkeley

Professor Ken Goldberg, Chair

The internet enables us to collect and store unprecedented amounts of data. We need better models for processing, analyzing, and making conclusions from the data. In this work, crowdsourcing is presented as a viable option for collecting data, extracting patterns and insights from big data. Humans in collaboration, when provided with appropriate tools, can collectively see patterns, extract insights and draw conclusions from data. We study different models and algorithms for crowdsourcing discovery.

In each section in this dissertation a problem is proposed, the importance of it is discussed, solutions are proposed and evaluated. Crowdsourcing is the unifying theme for the projects that are presented in this dissertation. In the first half of the dissertation we study different aspects of crowdsourcing like pricing, completion times, incentives, and consistency with in-lab and controlled experiments. In the second half of the dissertation we focus on Opinion Space[1] and the algorithms and models that we designed for collecting innovative ideas from participants. This dissertation specifically studies how to use crowdsourcing to discover patterns and innovative ideas.

We start by looking at the CONE Welder project[2] which uses a robotic camera in a remote location to study the effect of climate change on the migration of birds. In CONE, an amateur birdwatcher can operate a robotic camera at a remote location from within her web browser. She can take photos of different bird species and classify different birds using the user interface in CONE. This allowed us to compare the species presented in the area from 2008 to 2011 with the species presented in the area that are reported by Blacklock in 1984 [Blacklock, 1984]. Citizen scientists found eight avian species previously unknown to have breeding populations within the region. CONE is an example of using crowdsourcing for discovering new migration patterns.

Crowdsourcing can also be used to collect data on human motor movement. Fitts' law is a classical model to predict the average movement time for a human motor motion. It has been traditionally used in the field of human-computer interaction (HCI) as a model

[1] opinion.berkeley.edu

[2] Available at http://cone.berkeley.edu/ from 2008 to 2011

that explains the movement time from an origin to a target by a pointing device and it is a logarithmic function of the width of the target (W) and the distance of the pointer to the target (A). In the next project we first present the square-root variant of the Fitts' law similar to Meyer et el. [Meyer et al., 1988]. To evaluate this model we performed two sets of studies, one uncontrolled and crowdsourced study and one in-lab controlled study with 46 participants. We show that the data collected from the crowdsourced experiment accurately follows the results from the in-lab experiments. For Homogeneous Targets the Square-Root model ($T = a + b\sqrt{\frac{A}{W}}$) results in a smaller ERMS error than the two other control models, LOG ($T = a + b \log \frac{2A}{W}$) and LOG' ($T = a + b \log \frac{A}{W} + 1$) for $A/W < 10$. Similarly for Heterogeneous Targets the Square-Root model results in a significantly smaller ERMS error when compared to the LOG model for $A/W < 10$. The LOG model resulted in significantly smaller ERMS error in the $A/W > 15$. In the Heterogeneous Targets the LOG' model consistently resulted in a significantly smaller error for $0 < A/W \leq 24$. These sets of experiments showed that the crowdsourced and uncontrolled experiment was consistent with the controlled in-lab experiment. To the best of our knowledge this is the largest experiment for evaluating a Fitts' law model. The project demonstrates that in-the-wild experiments, when constructed properly, can be used to validate scientific findings.

Opinion Space is a system that directly elicits opinions from participants for idea generation. It uses both numerical and textual data and we look at methods to combine these two sets of data. Canonical Correlation Analysis, CCA, is used as a method to combine both the textual and numerical inputs from participants. CCA seeks to find linear transformation matrices that maximize the lower dimension correlation between the projection of numerical rating (Xw_x) and textual comments onto the two dimensional space (Yw_y). In other words it seeks to solve the following problem $argmax_{w_x,w_y} corr(Xw_x, Yw_y)$ in which X and Y are representations of the numerical rating and textual comments of participants in high dimensions and Xw_x and Yw_y are their lower dimension representations. By using participants' numerical feedbacks on each others' comments, we then develop an evaluation framework to compare different dimensionality reduction methods. In our evaluation framework a dimensionality reduction is the most appropriate for Opinion Space when the value of $\gamma = -corr(r, D)$ has the largest value. In $\gamma = -corr(R, D)$, R is the set of r_{ij} values. r_{ij} is the rating that the participant i is giving to the textual opinion of participant j. Similarly D is the set that contains d_{ij} values. d_{ij} is the Euclidean distance between the locations of participant i and j. In this dissertation we provide supporting argument as to why this evaluation framework is appropriate for Opinion Space. We have compared different variations of CCA and PCA dimensionality reductions on different datasets. Our results suggests that the γ values for CCA are at least %169 larger than the γ values of PCA, making CCA a more appropriate dimensionality reduction model for Opinion Space.

A product review on an online retailer website is often accompanied with numerical ratings for the product on different scales, a textual review and sometimes information on whether or not the review is helpful. Generalized Sentiment Analysis looks at the correlation between the textual comment and numerical rating and uses that to infer the numerical

ratings on different scales from the textual comment. We provide the formulations for using CCA for solving such a problem. We compare our CCA model with Support Vector Machine, Linear Regression, and other traditional machine learning models and highlight the strengths and weaknesses of this model. We found that training the CCA formulation is significantly faster than SVM which is traditionally used in this context (the fastest training time for SVM in LibSVM was 1,126 seconds while CCA took only 33 seconds for training). We also observed that the Mean Squared Error for CCA was smaller than other competing models (The MSE for CCA with tf-idf features was 1.69 while this value for SVM was 2.28). Linear regression was more sensitive to the featurization method. It resulted in larger MSE when used on multinomial ($MSE = 8.88$)and Bernoulli features ($MSE = 4.21$) but smaller MSE when tf-idf weights were used ($MSE = 1.47$).

To Marjan

For keeping me sane and giving me hope when I was ready to give up.

Contents

List of Figures

List of Tables

Acknowledgments

I am indebted to my advisor, Professor Ken Goldberg. Without his guidance and scientific wisdom this dissertation would not have been possible. His passion for science and love for new ideas inspired many aspects of this dissertation. I am grateful for his mentorship during my PhD education.

I would like to thank my committee, Professor Ilan Adler, Professor Laurent El Ghaoui, and Professor Bjorn Hartmann whose constructive feedback encouraged me to do more. I am also grateful for my family who supported me and encouraged me during my many years at Berkeley.

I am also grateful to the members of Opinion Space and CONE team: Yan Zhang, Ephrat Bitton, Tavi Nathanson, David Wong, Elizabeth Goodman, Gail de Kosnik, Christopher Goetz, Dhawal Mujumdar, Meghan Laslocky, Rupa Saheli Datt, Susan Miller, Linda Kelley, Kay Loughman, Eleanor Pirry and Brooke Miller. Many thanks to Bobby Nyotta, Maneesh Agrawala, Taylor Berg-Kirkpatric, Timmy Siauw, Dmitry Berenson, Jun Wang and John Canny for their insightful feedback.

My research was partially supported by the Berkeley Center for New Media, Fujitsu Labs of America, Berkeley AMPlab, Panasonic, Microsoft and Google. I would like to thank these organizations for their generous support.

Chapter 1

Introduction

The internet enables us to collect and store unprecedented amounts of data. We need better models for processing, analyzing, and making conclusions from the data. In this work, crowdsourcing is considered as a viable option for collecting data and extracting patterns and insights from big data. Humans in collaboration, when provided with appropriate tools, can collectively see patterns, extract insights and draw conclusions from data. We study different models and algorithms for crowdsourcing discovery.

Crowdsourcing is the process of outsourcing a task to a group of people [Howe, 2006]. Coined in 2006 by Jeff Howe, the term "crowdsourcing" refers to both online and offline outsourcing of tasks. However, in recent years it is more often referred to online and web-based processes when the group of people who are working on a specific task is not known to the task requester beforehand. Today, crowdsourcing is used in labeling images (e.g., The ESP game [Von Ahn, 2006] and CONE [Faridani et al., 2009]), eliciting creative work (e.g., 99 designs[1] and Jade Magnet[2], answering questions (e.g., Stack Overflow[3] and Quora[4]) and finding solutions to challenging scientific problems (e.g., Innoventive[5]) in addition to many more different micro tasks that requesters may post on crowdsourcing markets like Amazon Mechanical Turk[6]. While many crowdsourcing platforms like MobileWorks use dedicated crowds and automatically direct works to specific workers, an open call to invite everyone eligible to work on the task appears to be the most promising first step to crowdsourcing idea generation. We evaluate this assumption later in this dissertation when we compare the results of a crowdsourced web-based user study around the classical Fitts' law with a similar in-lab study.

In each section of this dissertation a problem is proposed, the importance of it is discussed, solutions are proposed and evaluated. Crowdsourcing is the unifying theme for the

[1] www.99designs.com
[2] www.jademagnet.com
[3] www.stackoverflow.com
[4] www.quora.com
[5] www.innocentive.com
[6] www.mturk.com

projects that are presented in this dissertation. In every chapter we highlight the shortcomings of our models so future researchers can improve these methods. In the first half of the dissertation we study different aspects of crowdsourcing like pricing, completion times, incentives, and consistency with in-lab and controlled experiments. In the second half of the dissertation we focus on Opinion Space[7] and the algorithms and models that we designed for collecting innovative ideas from participants. This dissertation specifically studies how to use crowdsourcing to discover patterns and innovative ideas.

We start by looking at the CONE Welder project which uses a robotic camera in a remote location to study the effect of climate change on the migration of birds. In CONE[8], an amateur bird watcher can operate a robotic camera at Sinton, Texas from within her web browser. She can take photos of different bird species presented in the area on that day. She or other participants of the system can later classify birds in the pictures using the user interface in CONE and the system will record each classification and will save that in the logs as soon as amateur scientists arrive at a consensus about the species of a bird. Sinton has the highest diversity of bird species in North America outside of the tropics. Because of that the presence of birds in that area is well documented over the past 30 years. This allowed us to compare the species presented in the area from 2008 to 2011 with the species presented in the area that are reported by Blacklock in 1984 [Blacklock, 1984]. Citizen scientists found 8 avian species previously unknown to have breeding populations within the region. CONE is an example of using crowdsourcing for discovering new migration patterns.

Crowdsourcing can also be used to collect data on human motor movement. Fitts' law is a classical model to predict the average movement time for a human motor motion. It has been traditionally used in the field of human-computer interaction (HCI) as a model that explains the movement time from an origin to a target by a pointing device and it is a logarithmic function of the width of the target (W) and the distance of the pointer to the target (A). In the next project we first present a succinct derivation of the square-root variant of the Fitts' law presented in Meyer et el. [Meyer et al., 1988]. The dataset that is used for this chapter is available online[9]. It contains data for two sets of experiments. An in-lab experiment and a web-based experiment. Each one of the experiments consists of data for two conditions: *Homogeneous Cursor Motion (Fixed Rectangles)* in which the participant repeatedly clicks back and forth between two vertical rectangles of fixed width and amplitude. After 11 repetitions, the width and amplitude are changed. The second condition is called *Heterogeneous Cursor Motion (Variable Circles)* in which the participant clicks on target circles of varying diameter and location.

We collected a total of 22,540 timing measurements (11,040 for homogenous targets and 11,500 for heterogenous targets) from 46 human participants in the in-lab (controlled) experiment. After outlier detection and cleaning, the dataset contains 16,170 valid timing measurements (8,250 for homogenous targets and 7,920 for heterogenous targets). Data

[7] opinion.berkeley.edu

[8] Available at http://cone.berkeley.edu/ from 2008 to 2011

[9] http://tele-actor.net/fitts-dataset/

clean up procedure is explained in details later in this dissertation.

The controlled experiment is accompanied by an equivalent crowdsourced and uncontrolled experiment. To conduct the uncontrolled study the same applet was made available online at http://www.tele-actor.net/fitts/. Online visitors indicate the type of their pointing device that they use but we cannot verify that. The online applet presents visitors with 24 homogenous targets and 25 heterogenous targets and collects 49 timing measurements. After data cleaning, the online study dataset includes 78,410 valid timing measurements (39,360 for the homogeneous targets and 39,050 for the heterogenous targets).

We show that the data collected from the crowdsourced experiment accurately follows the results from the in-lab experiments. For Homogeneous Targets the Square-Root model ($T = a + b\sqrt{\frac{A}{W}}$) results in a smaller ERMS error than the two other control models, LOG ($T = a+b\log\frac{2A}{W}$) and LOG' ($T = a+b\log\frac{A}{W}+1$) for $A/W < 10$. Similarly for Heterogeneous Targets the Square-Root model results in a significantly smaller ERMS error when compared to the LOG model for $A/W < 10$. The LOG model resulted in significantly smaller ERMS error in the $A/W > 15$. In the Heterogeneous Targets the LOG' model consistently resulted in a significantly smaller error for $0 < A/W \leq 24$. These sets of experiments showed that the crowdsourced and uncontrolled experiment was consistent with the controlled in-lab experiment. To the best of our knowledge this is the largest experiment for evaluating a Fitts' law model. The project demonstrate that in-the-wild experiments, when constructed properly, can be used to validate scientific findings.

Opinion Space is a system that directly elicits opinions from participants for idea generation. It uses both numerical and textual data and we look at methods to combine these two sets of data. Canonical Correlation Analysis, CCA, is used as a method to combine both the textual and numerical inputs from participants. CCA seeks to find linear transformation matrices that maximize the lower dimension correlation between the projection of numerical ratings (Xw_x) and textual comments onto the two dimensional space (Yw_y). In other words it seeks to solve the following problem $argmax_{w_x,w_y} corr(Xw_x, Yw_y)$ in which X and Y are representations of the numerical rating and textual comments of participants in high dimensions and Xw_x and Yw_y are their lower dimension representation. By using participants' numerical feedbacks on each others' comments, we then develop an evaluation framework to compare different dimensionality reduction methods. In our evaluation framework a dimensionality reduction is the most appropriate for Opinion Space when the value of $\gamma = -corr(r, D)$ has the largest value. In $\gamma = -corr(R, D)$, R is the set of r_{ij} values. r_{ij} is the rating that the participant i is giving to the textual opinion of participant j. Similarly D is the set that contains d_{ij} values. d_{ij} is the Euclidean distance between the location of participant i and j. In this dissertation we provide supporting argument as to why this evaluation framework is appropriate for Opinion Space. We have compared different variations of CCA and PCA dimensionality reductions on different datasets. Our results suggests that the γ values for CCA are at least %169 larger than the γ values of PCA making CCA a more appropriate dimensionality reduction model for Opinion Space.

A product review on an online retailer website is often accompanied with numerical

ratings for the product on different scales, a textual review and sometimes information on whether or not the review is helpful. Generalized Sentiment Analysis looks at the correlation between the textual comment and numerical rating and uses that to infer the numerical ratings on different scales from the textual comment. We provide the formulations for using CCA for solving such a problem. We compare our CCA model with Support Vector Machine, Linear Regression, and other traditional machine learning models and highlight the strengths and weaknesses of this model. We found that training the CCA formulation is significantly faster than SVM which is traditionally used in this context (the fastest training time for SVM in LibSVM was 1,126 seconds while CCA took only 33 seconds for training). We also noticed that the Mean Squared Error for CCA was smaller than other competing models (The MSE for CCA with tf-idf features was 1.69 while this value for SVM was 2.28). Linear regression was more sensitive to the featurization method. It resulted in larger MSE when used on multinomial ($MSE = 8.88$)and Bernoulli features ($MSE = 4.21$) but smaller MSE when tf-idf weights were used ($MSE = 1.47$).

One research question that arises in crowdsourcing is what incentives are needed to obtain timely responses. We model the completion time as a stochastic process and build a statistical method for predicting the expected time for task completion. We use a survival analysis model based on Cox proportional hazards regression. We present the results of our work, showing how time-independent variables of posted tasks (e.g., type of the task, reward, day posted, etc) affect completion time. We train on data collected from Mechanical Turk through Mturk tracker[10] and show that the The LR statistic is 8434 which represents a good model fit.

Contributions of this dissertation

Many of the projects that are mentioned in this dissertation are the results of years of collaboration among many people. Below are my contributions:

- **CONE** is a crowdsourcing platform for avian classification. It is designed as a game in which participants operate a telerobotic camera, and find and classify birds. The project was designed and implemented by Ken Goldberg and Bryce Lee. My contribution was to analyze the data and find seasonal migration patterns. During the project more than 29,000 photos were taken and 74 species were identified. We were able to identify eight new bird species that were not known to have a breeding population in the area. From these 8 species 3 have more than 1,000 photos in our dataset. Our dataset[11] has 3,659 photos of Green Jay (*Cyanocorax Yncas*), 1,710 photos of Bronzed Cowbird (*Molothrus Aeneus*) and 1,671 photos of Buff-bellied Hummingbird (*Amazilia Yucatanensis*). The results showed that crowdsourcing can be successfully

[10] mturk-tracker.com

[11] Openly available for research at http://cone.berkeley.edu/dataset

used for studying bird population and migrations. The results of these contributions were presented in two publications [Faridani et al., 2009, Rappole and Faridani., 2011].

- **Crowdsourcing user experiments**: This project started six years before I joined UC Berkeley. Previously Ken Goldberg had devised a new derivation for the Square-Root form of the Fitts' law. He had also ran an uncontrolled experiment with more than 1,500 participants to evaluate the model. I designed and conducted the equivalent in-lab experiment to the uncontrolled experiment. The main contribution in this dissertation is that we compare and study three models, the original Logarithmic Fitts' model, the Square-Root variant and the McKenzie model on different values of the index of difficulty[12] A/W. We show that the crowdsourced data are agreeing with the in-lab controlled data. For Homogenous targets we found that the SQR model more accurately predicts cursor movements when compared to both McKenzie model and Logarithmic model ($p < 0.05$). However for $A/W = 24$ we found that the McKenzie model provides a better fit $p = 1.18^{-2}$. For heterogenous targets we found that, when compared directly to the Logarithmic model, the Square-Root variant is more successful in predicting human movements in the conditions with small index of difficulty (approximately $A/W < 10$) and the Logarithmic model is more accurate in the conditions with larger index of difficulty (when A/W is above 10).

- Opinion Space demonstrates crowdsourcing of open-ended ideas and suggestions for innovation. The system was designed and implemented by Ken Goldberg, Tavi Nathanson, Ephrat Bitton, myself, David Wong, Sanjay Krishnan, and Kimiko Ryokai. I led a user study and we found that dwell times in the Opinion Space interface is significantly larger than the dwell times in the control list interface ($p = 2.2^{-16}$). Also participants found Opinion Space as a tool to find more useful comments and ideas ($p = 0.00361$). Also, interestingly compared to the two control interfaces (List and Grid interface) the participants in Opinion Space agreed more with the comments that they read in that interface ($p = 2.073^{-5}$). Finally we found that participants in Opinion Space, on average, rated the comments that they viewed higher than the ones that they viewed in the List or in the Grid interface ($p = 1.105^{-3}$)

- A major contributions of this dissertation is the use of Canonical Correlation Analysis (CCA) for Opinion Space. CCA allows us to combine both textual responses and numerical responses[13] that we collect in Opinion Space. Here we compare CCA with Principal Component Analysis and we show that the γ value for the CCA method was approximately %169 higher than that of PCA, pointing to the fact that CCA is a more appropriate dimensionality reduction for Opinion Space than PCA.

[12]The magnitude of the distance to the target A over the width of the target W in the Fitts' law model is known as the index of difficulty A/W

[13]This model is explained in detail in the upcoming chapters. CCA works by finding two linear projection matrices w_x and w_y such that the correlation coefficient of two correlated datasets (in this case the textual responses in the Opinion Space X and the numerical responses Y is maximized $argmax_{w_x,w_y} corr(Xw_x, Yw_y)$

- **Multi-aspect sentiment analysis**: This dissertation introduces multi-aspect sentiment analysis. In multi-aspect sentiment analysis, instead of only one overall rating for a given text, the method assigns different numerical values for different aspects of the product or service. I derived a CCA model for performing multi-aspect sentiment analysis and showed that it can outperform current models that are used for classical sentiment analysis. The Mean Squared Error of the predictions for our CCA model was 1.69 which was smaller than that of Naive Bayes ($MSE = 2.26$) and Support Vector Machines ($MSE = 2.04$). The model works based on the fact that the lower dimension representation of the textual comments ($E_{yi} = Y_i w_y$) is close to the lower dimension representation of the numerical ratings ($E_{xi} = X_i w_x$) according to the CCA formulation. Assuming $E_{xi} \approx E_{yi}$ allows us to calculate an expected multivariate numerical rating vector X_i for a given characterized text Y_i. Compared to the SVM, this model was significantly faster and took only 33 seconds for training on our dataset compared to 1,126 seconds for SVM.

- **Pricing and Completion Time Estimations for Crowdsourcing**: Before our work in this area there was no model for estimating the completion time for markets like Mechanical Turk[14]. Our dataset contains 4,113,951 individual tasks (known as HITs on Mechanical Turk) that are worth $344,260. There are 126,241 HIT groups from 7,651 requesters. Together with Bjoern Hartmann I built a model for pricing tasks on mechanical Turk and showed that there is a correlation between pricing and completion time. Our market model can instruct task requesters on these markets to post their tasks wisely. For example we show that tasks that are 20% less likely to be picked up by workers are the tasks that are posted during the week. The paper appeared in AAAI/HCOMP conference.

This dissertation resulted in 7 publications and 1 patent application.

[14]http://mturk.com

Chapter 2

CONE: Using Crowdsourcing for Bird Classification and Studying Avian Range Change

2.1 Introduction

Historical documentation of wildlife often requires observation of animal behaviors in their original habitat over an extended period of time. This exposes a scientist to dangers, is expensive and tiresome, and the remote and inhospitable location might be hard to reach especially during night. In this chapter we discuss the development of a telerobotic observatory system that allows amateur scientists (also known as citizen scientists) to collaborate with professional scientist to document avian migration patterns and enables us to remotely observe, document and analyze animal activities over the internet.

In this chapter we describe CONE[1]-Welder, installed at the Rob & Bessie Welder Wildlife Foundation in Sinton, Texas, an area known to have the highest bird diversity in the United States outside the tropics. The main component of CONE is a telerobotic camera that allows an amateur scientist to operate the camera from the comfort of her home. It enabled us and our collaborators at the Smithsonian Institute and the Texas A&M University to collect photographic and quantitative data on subtropical bird species, their diversity, and their migration patterns. The system was deployed on 12 May 2008 and over 700 users ("players") participated online. Players requested over 2.2 million camera frames and captured over 29,000 photographs. Within the photographs that are collected by citizen scientist in CONE, 74 unique species were discovered. More importantly citizen scientist discovered eight avian species without a known record of having a breeding population in the area. The CONE dataset documents the presents of birds of particular interest like the Green Jay (*Cyanocorax yncas*). The project is another example of collaborative discovery where game mechanics, crowdsourcing, and collaborative robotics help us identify and classify different bird species.

[1]Short for "Collaborative Observatories for Natural Environments"

This chapter describes the system architecture, the game interface that provides incentives for player participation, and data collected. CONE-Welder was available online at: http://cone.berkeley.edu/. Parts of this chapter have been formerly published by the author and colleagues in the following papers [Faridani et al., 2009, Rappole and Faridani., 2011].

2.2 CONE: Collaborative Observatories for Natural Environments

Figure 2.1: The welcome screen of CONE-Welder. The interactive interface is developed using Adobe Flash and allows participants to share a Panasonic robotic camera in Texas over the web. Participants can operate the camera, capture photos of different birds and classify the photos that they take as well as those taken by other participants. The screenshot in this figure includes a Great Kiskadee (*Pitangus sulfuratus*) and a Green Jay (*Cyanocorax yncas*), both of these are species of interest in this project.

The CONE-Welder project is the last installment in a series of field installations that is the result of collaboration between UC Berkeley, Texas A&M, the Smithsonian Institution, and the Rob & Bessie Welder Wildlife Foundation. CONE was installed within the Rob & Bessie Welder Wildlife Refuge, 12 km NE of Sinton, Texas (28E6'51.1" N, 97E25'2.2" W) as shown in Figure 2.2. Historically, this region has been chosen for avian behavior studies because it has the largest diversity of bird species in the United States outside the tropics.

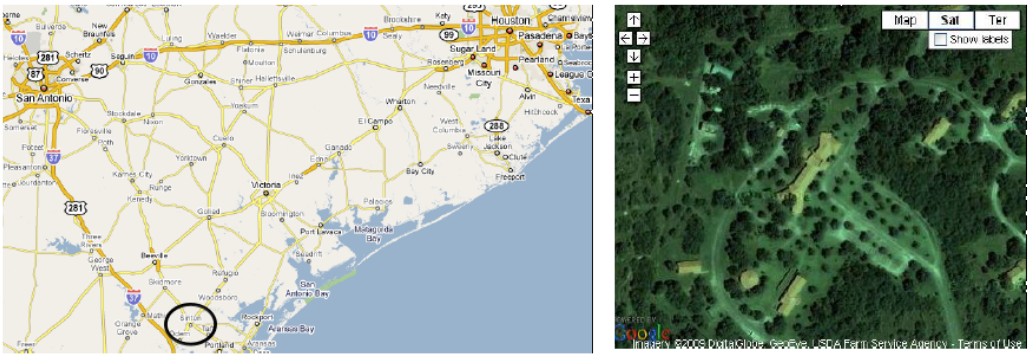


Figure 2.2: Aerial view and the location of the Welder Wildlife refuge on the map. The Welder Wildlife Refuge was established in 1954, 12 km northeast of Sinton, Texas.

Because of the long history of avian studies at Welder, there are detailed record of bird species in the region. This allows researchers to compare the current seasonal presence of certain bird species with the records that go back to the establishment of the Welder refuge in 1954 [Blacklock, 1984, Glasscock and Blankenship, 2007].

Documenting Bird Activities at Welder, Challenges and a Solution

Welder is located in a remote location. While human resources are scarce at the site, there is a significant amount of information that needs to be collected and analyzed to fully document the avian species present. Additionally, to document seasonal bird activities during the nighttime, scientists need to work during those periods. This raises a significant challenge in documenting avian activities in the area.

According to the report by U.S. Fish & Wildlife Service there were 48 million birdwatchers in the United States alone in 2006 [Carver, 2006]. Crowdsourcing the challenge above will enable us to use the large number of bird watchers in the US and around the world to help ornithologists study avian activities at Welder.

The objectives for the CONE-Welder initiative are as below:

1. CONE-Welder aimed to document daily and seasonal presence of subtropical birds that were not known to have previous breeding activities as far north as the Welder Refuge (Fig 2.1). Through this process the project aimed to study the effect of climate change on the migration and breeding of these birds [Rappole and Faridani., 2011].

2. The second goal of the project was to record the presence of the birds that were already banded and color-marked by the staff at the Welder-Refuge. The process of capturing, marking and taking blood samples from the birds is presented in [Rappole et al., 2007].

3. Enabling scientists to study the activities of birds and other animals at night.

The main objective for the study was to answer the question about avian range change and habitat shift that might be due to global effects like climate change. For this project the collaborators at Welder Wildlife Refuge built and maintained the infrastructure for mounting the robotic camera, feeding stations to attract birds, and night lights to illuminate areas of interest. To record the nesting and the breeding of avian species, Welder refuge undertook a two year project to search for nests and to locate and document breeding places of the new subtropical birds. After color-marking each bird a blood sample was acquired to study weather or not a parent of the bird was previously marked by scientists [Rappole and Faridani., 2011]. Over the period of time that the project was running, the staff at Welder Wildlife Refuge volunteered to maintain the feeding stations and they also helped with calibrating the camera and maintaining the hardware components of the system. The system ran on a low-bandwidth residential internet connection that was shared with other staff at the location. The throughput of the system was about 4 frames per second.

Despite all the limitations, the CONE-Welder project was able to engage and involve citizen scientist around the globe, including amateur bird watchers and students. Collaboratively, these participants photographed and collected information about the avian activities in the region. This project builds on past installations that have developed new models for collaborative observation, drawing on computational geometry, stochastic modeling, and optimization [Dahl, 2007a].

CONE-Welder introduces several new features:

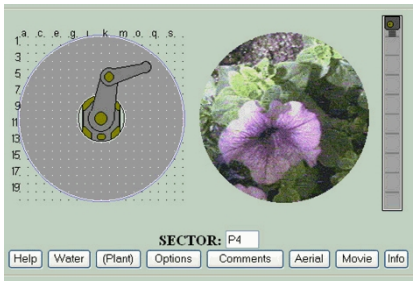
- The remote environment of the Welder Refuge provides extreme bird diversity
- Professionally designed feeding stations
- Night lights installed near the feeding areas for continuous observation and avian study through the night
- Interactive and cross platform interface
- Interactive image classification that enables citizen scientist to highlight different areas of the image that contain birds, determine species of birds in each segment of the image, and write a comment about the image. Through this process the system will mark the species with enough information as true classification and continue marking the ones with no confidence as the ones that need more information.
- Gamification through badges and scores

Previous Work and Related Literature

Telerobotics

The first web-based telerobotic projects appeared shortly after the invention of the World Wide Web. Goldberg and his colleagues developed the Telegarden (1995-2004) through which

(a) Telegarden robot



(b) The web-based interface of the telegarden

an online participant could plant and water a seed over the internet and through a web-based interface [Goldberg et al., 1995] . Telegarden was active from 1995 to 2004 attracting over 9,000 participants making it arguably the largest public telerobotic project in history [Marín et al., 2005]. The web-based interface and a front view of the telegarden robot is shown in Figure 2.2. A survey of literature of early telerobotic and internet-based robotic projects is available in [Goldberg and Siegwart, 2002]. Later in "Tele-Actors" the robot was replaced by a human actor allowing online participants to interact with the environment and participate in events [Goldberg et al., 2003]. Goldberg and colleagues also introduced multi-user control of robots by using a spacial voting system instead of a traditional queue of tasks. More recently telerobotics has been used in many different fields including telerobotic surgery [Arora et al., 2011, Haidegger et al., 2011], handling of explosives [Erickson et al., 2012], and even multi-user robotic camera for video conferencing by Kimber, Liu, Foote et al [Kimber et al., 2002, Liu et al., 2002]. For recent examples see [Kim et al., 2002, Schiff et al., 2009, Song et al., 2008b]. For more discussion of collaboration algorithms on shared robots see [Xu, 2012, Corredor and Sofrony, 2011]

The frame selection problem

To allow multiple participants to operate the telerobotic camera, CONE uses a frame selection algorithm. The "frame selection problem" for a shared networked telerobotic camera was introduced by Song et al., in [Song et al., 2006]. In the frame selection problem it is assumed that the control of only one single robotic camera is being shared among many participants. In practice these participants often connect to the camera through a web-based interface. In the frame selection problem n participants simultaneously submit a frame request to the camera and the algorithm has to output a number of optimal frames that maximizes the satisfaction among all participants [Song, 2009a]. The initial algorithm based on grouping and sorting of virtual corners had time complexity $O(n^2m)$ for n users and m zoom levels. Har-Peled et al. improved this to an $O(mn^{3/2}log^3n)$ algorithm. Song et al. provide a new distributed algorithm for solving the frame selection problem [Song et al., 2003]. See [Song, 2009b] for recent results on this problem.

Robotic Avian Observation

The Bird-Watching Learning System (BWL) was designed and implemented by Chen and colleagues to enable birdwatchers to capture and classify images of the birds by using PDAs over WLAN networks [Chen et al., 2003]. BWL was used at three elementary schools in Taiwan. One of the properties of BWL is that the birdwatcher has to be at the same place as the bird to be able to take a snapshot of the bird. Chen et al. conclude that children who used the BWL system could significantly improve their learning skills through BWL.

CONE-Welder builds on the infrastructure that was developed in previous iterations. Often in remote places the bandwidth of the internet is not large enough to stream a quality video to all the users. Dahl solves this problem by introducing an intermediate *relay server* [Dahl, 2007b]. In this setting the relay server will log into the camera as the sole user of the system. The camera would provide the relay server with separate frames in the JPG format and the relay server that is placed in an area with a higher internet bandwidth produces a stream of video and provides it to other users. This will prevent the camera from facing a high demand and potentially coming to a halt.

Gamification

Deterding *et al.*, define gamification as the process of using game element in non-game contexts [Deterding et al., 2011b, Deterding et al., 2011a].

The crowdsourced avian classification model that is used in CONE-Welder is inspired by is defined von Ahn's PeekaBoom [Von Ahn et al., 2006] and Google's Image Labeler [Weber et al., 2008]. In Google Image Labeler an image is presented to two participants not necessarily online at the same time. Each participant provides a number of labels for the image and they will both receive points if they provide the same label. Since the actual image is the only piece of information that is shared between the two participants it is in their best interest to provide correct information to the system. This will allow the system to improve its labeling through this crowdsourcing procedure.

Collaborative Camera Control

There are three significant components in CONE-Welder:

1. The Panasonic robotic camera installed in the Welder Wildlife Refuge and connected to the internet through a low bandwidth residential internet

2. The network server that serves as a relay server, calculates candidate frames that are being sent to the camera and serves the web-pages

3. The client interface that is shown in participants' browsers and can communicate with the camera through the intermediate network server

Camera

All of the CONE projects are built around a single, auto-focus and networked robotic camera. In the CONE-Welder project we have used the Panasonic KX-HCM280 Network Camera (Fig 2.3). The camera provides 350 degrees of pan range and 210 degrees for the tilt range. With 21x optical zoom (42x digital zoom) KX-HCM280 is capable of focusing on small objects such as hummingbirds and even insects from the distance. The maximum resolution for this camera is 640 by 480 pixels and maximum frame-rate of 30 FPS. In the CONE-Welder project, due to the limited Internet bandwidth at the Welder Wildlife Refuge site the video size was reduced to 320 by 240 pixels. This reduction in frame size increases the frame throughput. Due to the low bandwidth at the site, even with this frame size reduction, we could only receive 2-4 frames per second from the camera.

Figure 2.3: Panasonic KX-HCM280 Network Camera

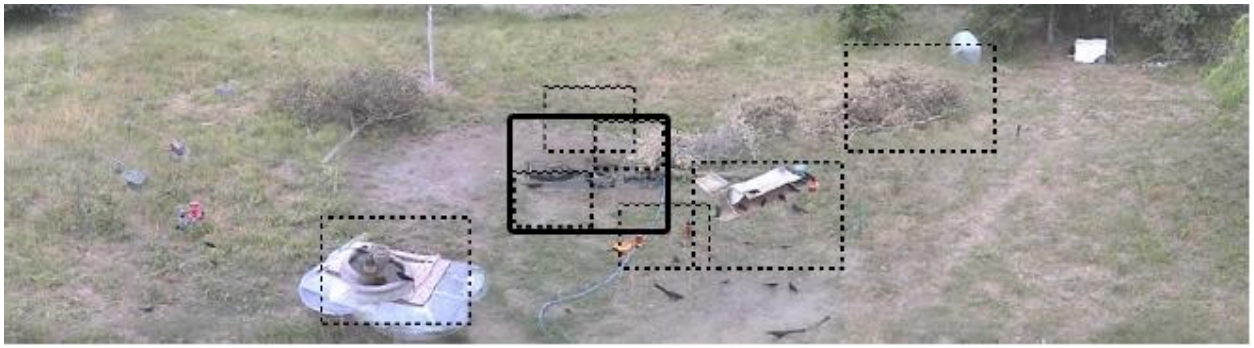


Figure 2.4: To select a candidate frame from multiple requests CONE uses a frame selection algorithm on the server. Adopted from [Dahl, 2007a] this frame selection model comes up with a candidate frame that maximizes user satisfaction. Dotted rectangles show the frames that are requested by different participants simultaneously r_i. The frame with the solid border is the final frame selected by the camera f.

Frame Selection

There is only one camera in CONE and all participants share this one viewpoint. The intuition behind the frame selection algorithm is as follows: As shown in Figure 2.4 each participant requests a frame on the panorama from the server r_i. Let's denote the set of n frames that are requested simultaneously by participants as R. Then $R = \{r_1...r_n\}$. An optimal single frame f is then selected by the algorithm (Fig. 2.4). Dahl suggests minimizing the user dissatisfaction as the mathematical model for frame selection [Dahl, 2007a]. In the process of selecting a single optimal frame from the set of requested frames R may result in a frame f that has no intersection with a requested frame r_j. In this case r_j is marked as an unsatisfied frame and will be inserted into the set of subsequent requested frames. To make sure that unsatisfied frames get processed faster, the priority weight of r_j is increased. In other words the requested frame that has been waiting more will be weighted more heavily. This guarantees that starvation will not happen and no frame is left unfulfilled.

Song et al., provides a single period algorithm that minimizes user satisfaction over one period of time [Song et al., 2008a]. It is memory-less and does not consider user dissatisfaction over multiple periods. This inspired adaptation of the time-dissatisfaction in CONE [Dahl, 2007a]. The algorithm not only minimizes the mean dissatisfaction it also shrinks the variance of user dissatisfaction across all requests in R. Meaning that the worst dissatisfaction will not be significantly larger than the average satisfaction. For more details on the frame selection algorithm please see author's former publication [Faridani et al., 2009].

2.3 User interface

The web-based interface was designed by Lee and Goldberg and is explained in [Lee, 2008]. As explained by Lee the main goal was to design the interface like a "bird lookout". Figure 2.1 demonstrates the main screen of CONE. The panoramic image on top is a static image that gives the participant a feel of possible search locations. The participant can see the location of feeding stations, water resources, trees and other spots that might be of interest. For frame selection direct manipulation [Shneiderman, 1993] was used. A participant can request a frame from the camera by directly drawing a rectangular bounding box around different objects on the panorama. The system will automatically translate this to an r_i request with proper coordinate values. The server automatically calucates the candidate frame f and requests it from the camera. The camera then moves to the location of f and stays there. Any participant can take a snapshot of the content of f at that time and this photo will be saved to the gallery as an unclassified photo.

A participant can then select a bounding box around the bird or any other interesting object in the photo. Figure 2.5 is an example of this zone selection and classification. A participant can highlight a bounding box around the bird and select the species from a drop down menu. CONE uses a voting schema to determine the confidence in an answer. If the number of votes for a species exceeds a threshold (here 3) the system will mark the majority

vote as the "true classification". If there are other suggestions for the species, the system will automatically raise the threshold, and until the 3 majority vote is met, the species will remain unclassified.

The system automatically generates analytics and reports that can help identify migration patterns. Figure 2.6 is a an example snapshot of the online classification report for Green Jays. The Green Jay was not known to have a breeding population in the area about 30 years ago, and CONE validated its presence in the Welder area.

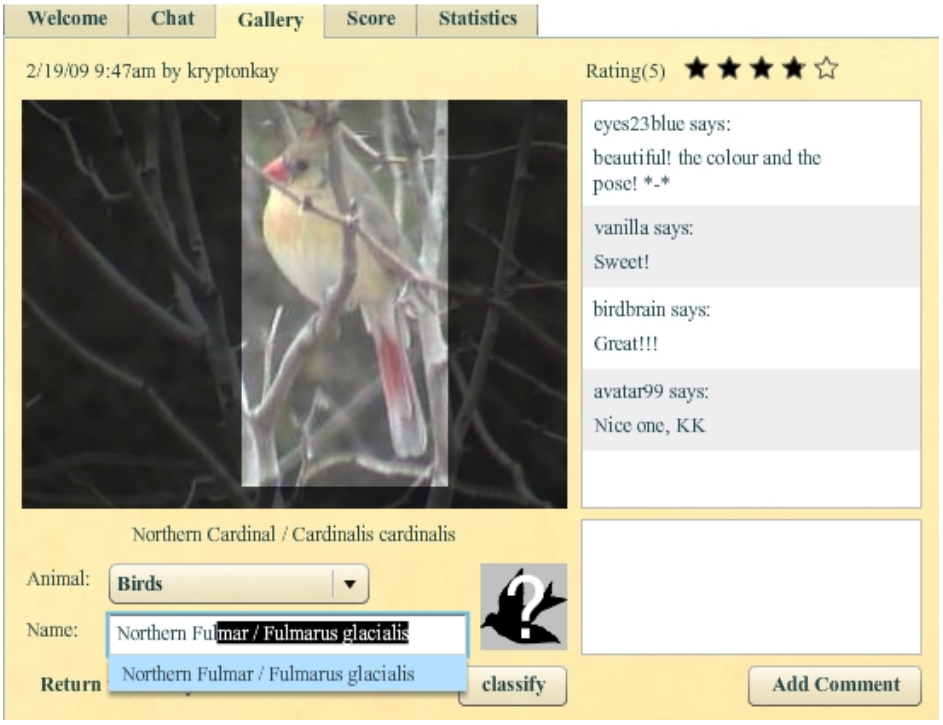
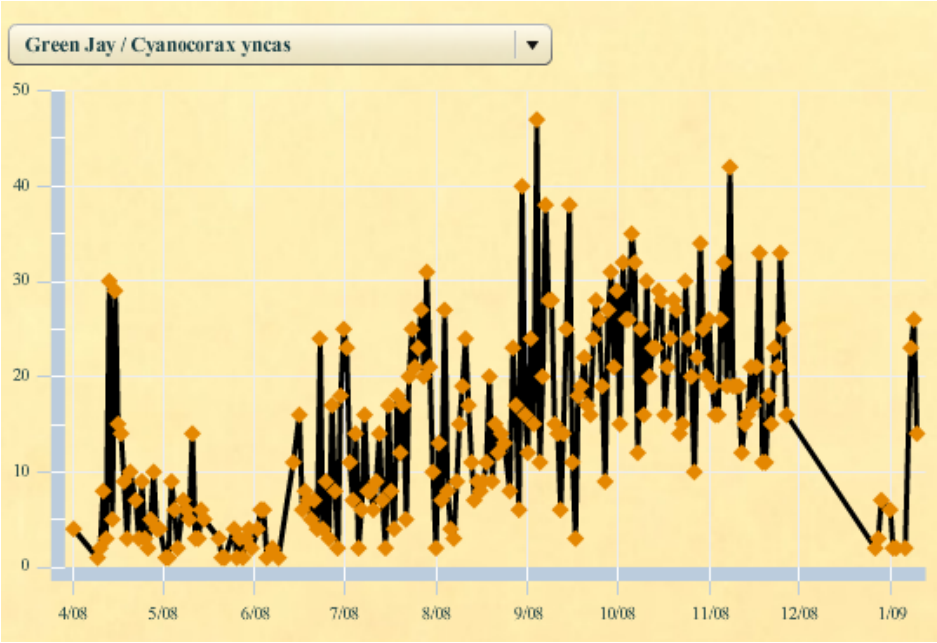


Figure 2.5: The box drawn around the bird is an example of Zone Classification. On this image, the photographer and at least three other participants have classified the bird in the Zone as a Northern Cardinal. Users can also rate each photo by assigning stars to each picture (top right). The chat area on the right allows citizen scientist not only to educate each other but it also allows them to engage with the community and support each other's efforts.

2.4 Gamification: incentivising participation through game elements

CONE generates valuable data for field biologists. To increase user engagement and encourage participation CONE uses different game elements. Users can collect badges and points



Graph of Green Jay sightings over time.

Figure 2.6: Avian data visualizations on the CONE-Welder site. The number of photos with a Green Jay is graphed from 4/2008 to 1/2009. Note that the system was still in private beta during the month of 4/2008. The Green Jay was not known to have a population in the area in previous documents [Blacklock, 1984].

for taking and classifying photos. Leading players are shown on the leaderboard. To engage new participants we also show the daily leaders on the front page of the interface as shown in Figure 2.1. Online badges are an inexpensive yet very efficient means to encourage participants to work towards a goal, engage with the community and keep coming back [Antin and Churchill, 2011]. To increase positive feedback CONE awards a *Daily Parrot Award* to the most useful comment of the day. *Cumulative Primary Classification Award* is awarded to the first participant who correctly classifies a bird. There are also awards that are related to human perception. A photographer may receive a *daily "Eagle Eye" award* if his photo had received the highest aggregated "star" rating as shown in (Fig. 2.5). There are also time-based awards that are designed to encourage participation through a 24-hour period. *Night Owl* is awarded to the last person who takes a photo during the night. Seven different online badges can be received for participation.

As of April 2009 there were over 97,000 awards with total value of over 125,000 points. A diagram of the total daily values of awards is shown in (Fig. 2.7).

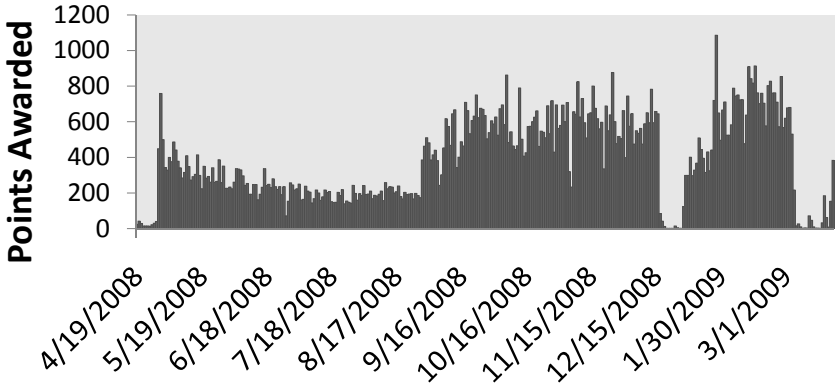


Figure 2.7: Daily value of game points as of 6 April 2009. There are two maintenance periods in the diagram during which no points were allocated

2.5 Data

Table 2.1 summarizes system usage. These data are summarized in Figure 2.8 where the number of log-ins to the system is modeled as a non-homogenous Poisson process.

Over 460 users have logged in to the system from April 18, 2008 to April 6, 2009. Of these, 256 users have contributed to classification and zoning. By the end of 2010 the system had more than 800 registered participants. CONE has a highly dedicated community of active users. The 30 most active users account for 120,838 score points, 96.4% of the total. A histogram of the number of snapshots is shown in Fig. 2.9.

By Oct 2010 the number of requests had increased significantly as shown in Fig. 2.2.

CONE participants often log-into the system many times during the day. Figure 2.10 shows the number of logins during each hour of the day. Figure 2.11 shows the percentage of photos that are taken in each day of the week. It shows that the system had a balanced

Table 2.1: Summary of statistics as of April 6,2009

Case	Amount
Frames requested by users	2,294,535
Frames selected by the system	2,014,196
Species	723
Subselections	33,110
Comments	15,609
Ratings	15,362
Awards distributed	97,326
Total value of awards	125,375

usage throughout the week.

Image Classification

Image classifications are useful to researchers to help document new species and track previously seen species. Participants defined zones, each a classification opportunity, on 93 percent of all photographs. Among these zoned photographs, 73 percent had at least one zone with a consensus classification. Furthermore, consensus classifications were established with an average of 4.5 votes. Users have identified a total of 74 unique species until April 2009 and 83 unique species until Oct 2010, shown in Fig 2.13. These results confirm that Welder Refuge is extraordinarily diverse, and also confirm the the presence of eight species whose range was not known to extend to the Welder refuge 30 years ago [Rappole et al., 2007].

The project produced a collection of bird images with more than 40,000 individual photos. This collection can serve as a training set for bird detection algorithms. The data set is available at (http://cone.berkeley.edu/frontdesk/gallery/).

Avian Range Change Data

Rappole and Glasscock have found fifteen Subtropical or Balconian (central Texas) species that now appear frequently in Welder during breeding seasons. These species were not there as breeders 30 years ago [Oberholser, 1974, Blacklock, 1984, Rappole and Blacklock, 1985]. With CONE we were able to document the presence of eight of these species through photo evidences (Table 2.3). Photographs of a newly-fledged Green Jay and a juvenile Bronzed Cowbird being fed by a Northern Cardinal confirm breeding by those species. A juvenile Eastern Bluebird was photographed in July 2008. In addition, we obtained photos of color-banded Green Jays from every month of the year, demonstrating year-round residency for this species at Welder. Figure 2.14 shows two examples of the photos of the Green Jay and the Eastern Bluebird.

Table 2.2: Summary of camera statistics for October 8,2010

Case	Amount
Frames requested by users	3,752,613
Frames selected by the system	3,272,585
Photos taken	44,585
Number of players	842
Unique species	83

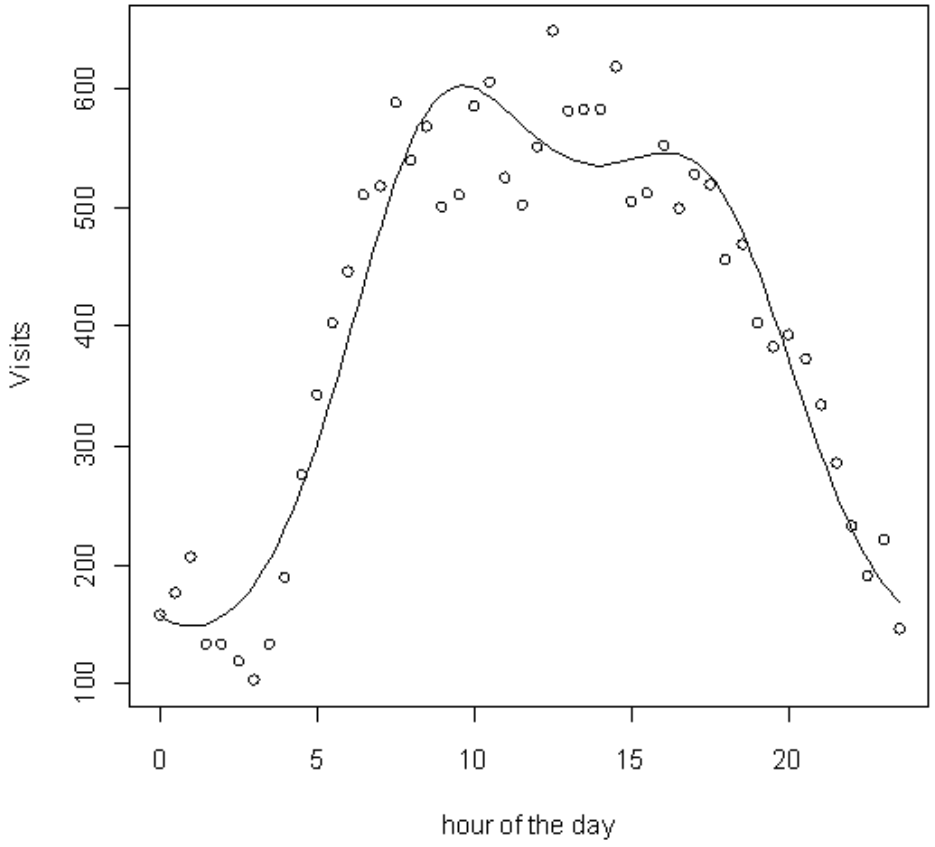


Figure 2.8: Poisson regression on the number of visits per hour of the day. Midnight is represented as time $t = 0$. We later used this for estimating the rate $\lambda(t)$ to be used in simulation process for building the autonomous agent for CONE.

2.6 An autonomous agent for CONE

One of the aspects of CONE was to educate the public. During the project we realized that some people log on to the website not to look for birds or classify them but simply to enjoy the view of interesting birds. The video coming from the camera is always interesting when somebody is operating the camera but when everyone is simply watching the feed the video appears to be uninteresting. We wanted to provide these participants with an interesting video feed even during the periods that nobody was operating the camera. Furthermore, this video feed could be an interesting screen saver for TV sets. To be able to still operate the camera during the times that participants are not moving the camera the author wrote an autonomous agent. Note that the goal of the autonomous agent was loosely defined as "providing an interesting video feed to passive viewers". In this section we provide a framework for automated search agent that can look for birds in their natural habitat using

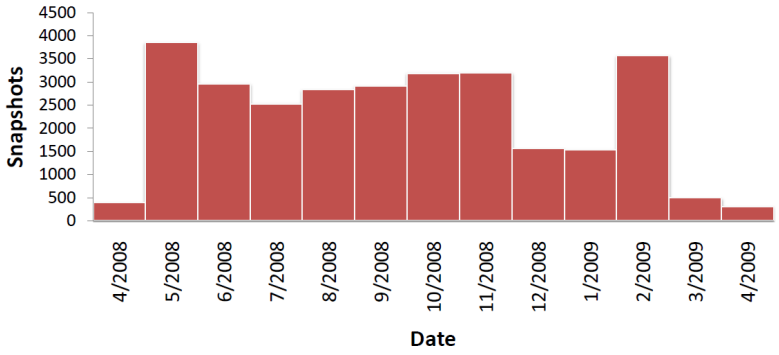


Figure 2.9: Histogram of the number of snapshots taken by users.

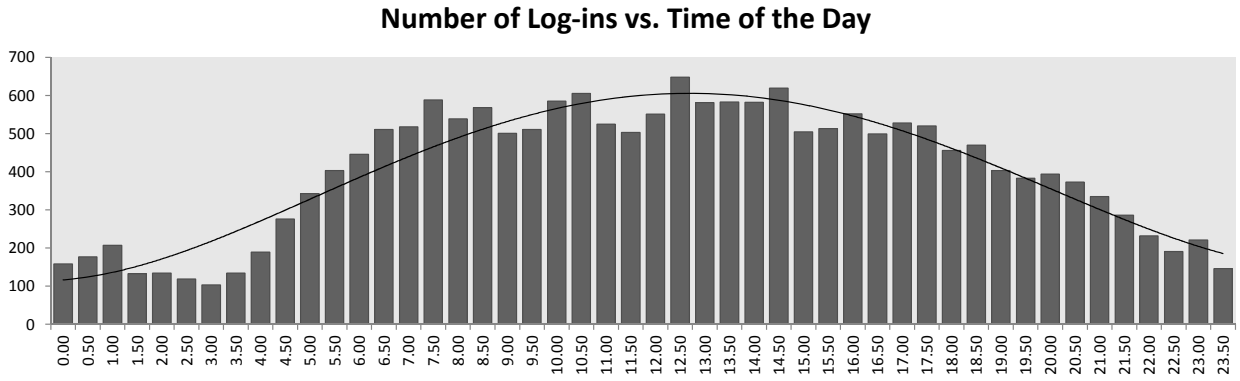


Figure 2.10: Number of user log-ins for each hour of the day

the cone infrastructure.

An open-loop autonomous agent for the one-sided search problem

CONE uses a robotic camera with a low end onboard cpu. Because of this running a sophisticated image processing technique is not practical on the camera. In the susequent sections we provide arguments for why a closed loop search fails in this setting and argue that an open-loop search can be beneficial. Three heuristic algorithms have been devised for this application and were run in a simulated environment. Finally we deployed one on the actual system.

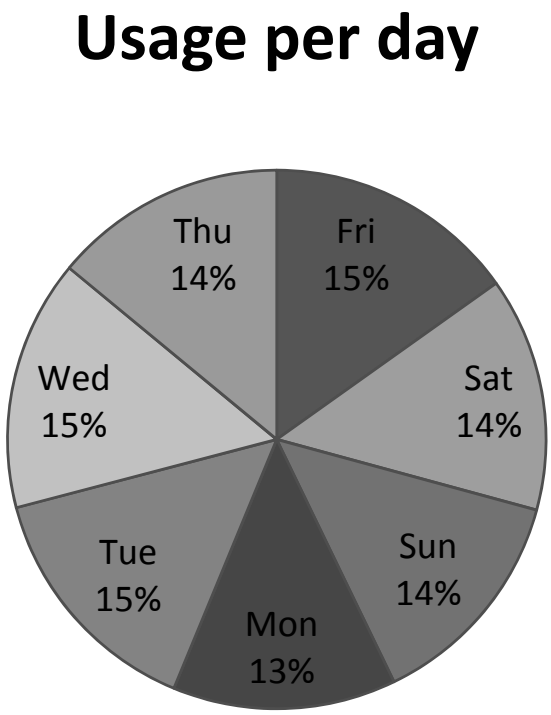


Figure 2.11: Percentage of photos taken in each day of the week

Previous work

Our problem is a case of a one-sided search problem in which the searcher's action does not affect the action of the target [Benkoski et al., 1991]. Schmidt and Goldberg developed an autonomous agent for motion detection [Schmidt, 2007] for CONE Welder's predecessor, CONE Sutro Forest. They implemented a Gaussian-mixture background subtraction model, effective in eliminating high wind effects on the images taken from the system. In their implementation the motion detection algorithm runs continuously in the background and makes a callback to the camera controller once the motion is detected in any subregion. Motion detection algorithms for outdoor natural environments are still ongoing research topics [Elgammal et al., 2000], [Wren et al., 1997],[Karmann and von Brandt, 1990] and [Stauffer and Grimson, 1999]. Previously Lee and Schiff used Adaboost on bird classification. They have used a learning set of bird images and defined a 4-by-4 grid on each image and trained the AdaBoost on the histogram of each cell. They report a 40% success rate in classifying the birds. Nadimpalli et. al. [Nadimpalli et al., 2006] have also worked on visual bird recognition. Bitton and Goldberg [Bitton and Goldberg, 2008] have developed an efficient algorithm for search and rescue. In their framework, the target is found by utilizing

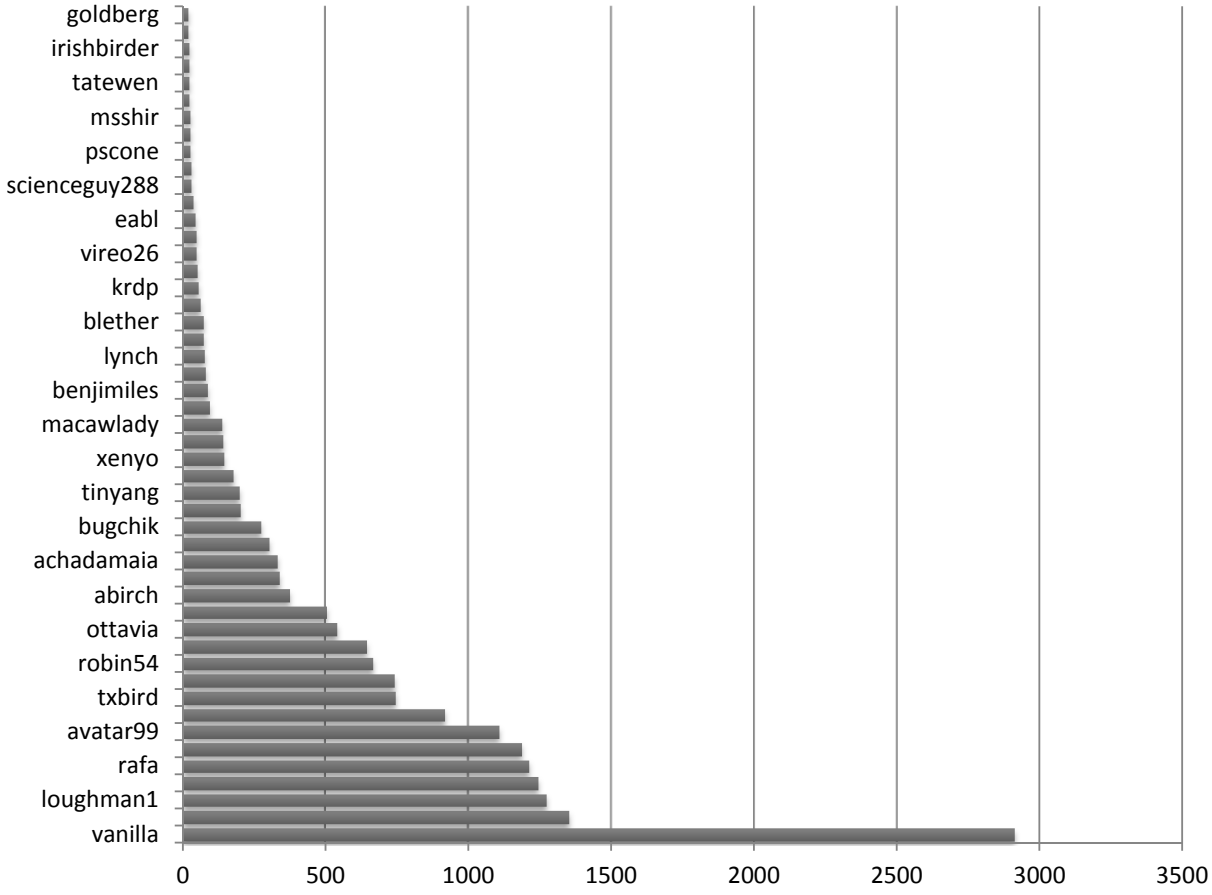


Figure 2.12: Histogram of the number of log-ins for each user from 8/22/2008 to 12/29/2008, it included 19348 sessions. Each user had to have at least 20 logins within that period to be included in the diagram. There were 49 participants who were qualified to be counted as active participants during that period.

a Bayesian model, and by maximizing the information gain. In a similar line of work Chung et al. formulate the optimal search and identification problem as a mixed integer linear programming model [Timothy H. Chung, Moshe Kress, and Johannes O. Royset, 2009]. They maximize the expected number of targets detected in a search process. They use GAMS/CPLEX to solve the MIP model and report 10 minutes of solution time for each instance of the optimization model for an area of 9km×11km. More work on optimization frameworks for search and rescue can be found in [Koopman, 1979], [Stone, 1978], [Washburn, 1981], [Benkoski et al., 1991].

Unfortunately, very low Internet bandwidth at the site (4 fps during low traffic periods

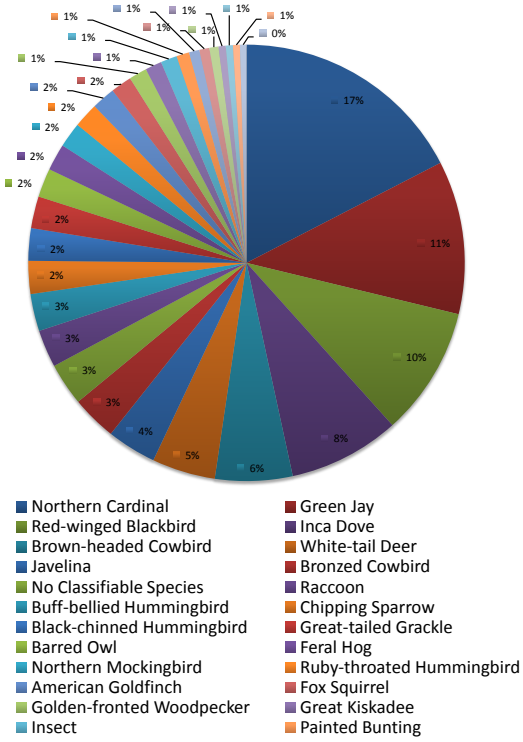


Figure 2.13: Species Classification Totals: April 28, 2008 - April 6, 2009

and 2-3 fps during regular hours) did not allow us to use frame image differencing algorithms effectively. There are often many changes in the scene during the subsequent times that a frame was computed that make the process of a closed loop search through frame differencing ineffective. We conclude that there are three reasons for the closed-loop search to be not feasible in CONE setting:

- The low power cpu on the camera was not fast enough for image segmentation.
- Successful image segmentation algorithms often fail on birds, especially in the low quality photos that come from CONE.
- Low bandwidth at Welder and communication lag does not allow any image processing to be done on the server, and thus a fast response to bird movement is not attainable.

Figure 2.6 shows an image segmentation method based on [Li et al., 2011b] that we performed on one of the photos obtained from CONE. After 680 iterations (about 5 minutes on a laptop with an Intel Corei7, 2.8GHz and 4GB of RAM) the method converged to a band around the bird. This long processing time disqualified this method for our real-time application.

(a) Green Jay (b) Eastern Bluebird

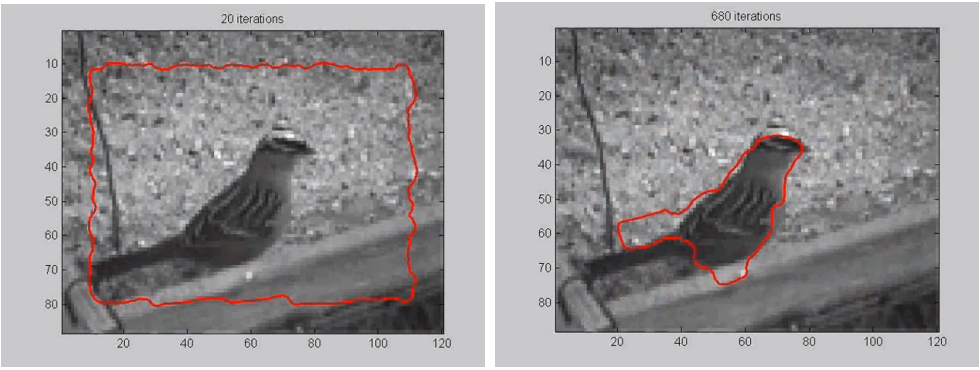
Figure 2.14: Photos of a Green Jay (a) and an Eastern Bluebird that are now appearing in Welder. Both of the species did not have a known breeding population 30 years ago.

Table 2.3: Subtropical or Balconian (central Texas) species that now occur at Welder during the breeding period that were not known to be there as breeders 30 years ago

Species	Photos
Green Jay (Cyanocorax yncas)	3659
Bronzed Cowbird (Molothrus aeneus)	1710
Buff-bellied Hummingbird (Amazilia yucatanensis)	1671
Black-chinned Hummingbird (Archilochus alexandri)	768
Great Kiskadee (Pitangus sulphuratus)	516
Eastern Bluebird (Sialia sialis)	144
Audubon's Oriole (Icterus graduacauda)	28
Couch's Kingbird (Tyrannus couchii)	12

Sensorless robotics has been previously discussed in the context of part feeding and mechanical manipulation by Mason and Erdmann [Mason, 1985] and [Erdmann and Mason, 1988]. They study real-time mechanical part feeding methods in cases in which using a sensor feedback is not feasible. In this work we study the cases in which an open-loop (sensorless) search algorithm is needed. Removing image processing modules from a vision system can reduce the complexity of a system when the closed-loop search is not feasible because of system resources and real-time constraints. For example a Marslander module may not be able to perform real time image processing in place because of its limited processing power and has to perform an open-loop search based on its belief tables (that can be collected by a number of satellites from Mars) and send all of the stored images for processing to the control station on Earth for future processing. Nadimpalli et. al. [Nadimpalli et al., 2006] discuss an autonomous robot for protecting crops from birds. Security cameras are another domain that open-loop search algorithms can be beneficial [Thompson, 1999]. Yu et. al. provide a number of conditions under which an open loop plan provides good performance [Yu et al., , Geyer, 2008]. They use the results to run a search and rescue operation with three robots

Figure 2.15: The CONE viewer uses an autonomous agent to provide hands free birdwatching experience for those who do not want to spend time looking for the birds.



(a) Original photo after 20 iterations of the segmentation algorithm

(b) The segmentation algorithm converged after 680 iterations and 5 minutes

in an office space. In POMDP algorithms, based on the prior belief, the algorithm takes actions in such a way that maximizes the expected rewards [Yu et al.,]. Papadimitriou and Tsitsiklis show that the complexity of the planning problem in a POMDP is PSPACE-complete [Papadimitriou and Tsitsiklis, 1987]. The best algorithm for n states over h horizons takes $2^{poly(n,h)}$ time to solve the problem.

A similar problem in the scientific literature is the *tag* problem [Rosencrantz et al., 2003], [Pineau et al.,]. In a tagging game the pursuer should find the target and tag it. The pursuer is penalized by 1 point for each extra action it takes and will be finally rewarded with 100 points once it finds the target and the game ends.

Therefore, we started exploring alternative algorithms for the autonomous agent. These algorithms are based on probabilistic models and do not need sensor feedback so they can work in environments with low Internet bandwidth where frame differencing algorithms are not suitable. These constraints, in addition to the very low Internet bandwidth at the Welder site, persuaded us to develop heuristic algorithms that can be implemented for real-time use.

The three algorithms designed for this purpose were as follows:

1. Algorithm I: A level set method
2. Algorithm II: Weighted sampling
3. Algorithm III: Simulation-based method

We developed three different methods for the autonomous agent. All the algorithms run in polynomial time. In the "Level set method" algorithm, areas with higher likelihood of finding a bird are scanned with higher zoom levels. In the "Purely statistical" approach, we harness the strength of the "Law of Large Numbers" and search the whole scene while expecting the maximum reward in the long run. Lastly, in the "Simulation-based method," we look at the top participants in the system and simulate and replicate their strategy and behavior on the system.

Heuristic Algorithms for the Autonomous Agent

When no human is operating the camera, the autonomous agent operates the camera by sending frame requests to the web server.

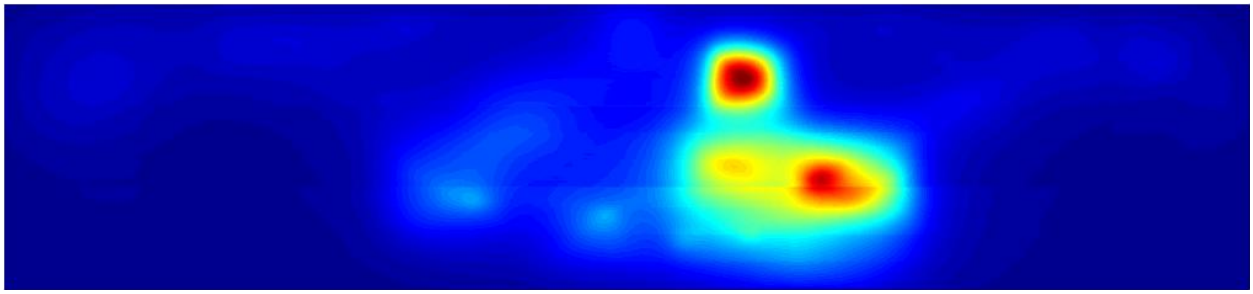
Algorithm I: A level set method

The first heuristic algorithm that was implemented for the system was a level set algorithm based on an initial "Density Map". Figure 2.16 shows the weighted "Density Map" generated by superimposing and normalizing the coordinates of all the photos from the database that contained a bird of interest. Birds are weighted based on their scarceness and importance, and this weighing is also considered in the generated Heat-map.

Our level set method is introduced in the Algorithm1. The algorithm is then implemented in MATLAB and compiled by "MATLAB Builder for Java" as a Java library to be used in the CONE framework.

An implementation of this algorithm is shown in Figure 2.17. A 3D representation of the probability density function (pdf) is sliced with a plane at each threshold step and the resulting level set is scanned with frames of certain zoom level. Zoom level is also associated

(a) Panoramic view of the CONE



(b) Density-map generated by superimposing all successful frame requests

Figure 2.16: Density-map generated for the CONE system based on all previous frame requests

with the average probability over the region and as the average probability becomes smaller camera zooms out. In the implementation we associate the *zoomlevel* with the *threshold*. The higher the *threshold* the higher the *zoomlevel*.

One of the advantages of the level set method is its ability to scan the area smoothly so that the camera will not jump from one location to another location. Although since the original heat-map will not change greatly after each run of the algorithm, the requested frames in the second run of the algorithm are very similar to the first set of frames.

Algorithm II: Weighted sampling

Algorithm I based on the level set method tends to generate similar frame requests in consecutive executions. As a consequence if we run the algorithm for a day many of the frame locations will be similar which makes it an uninteresting experience for the audience of the live feed. In order to solve this problem and generate more interesting frames, we augment **algorithm I** with more information. In this model we assume the probability of a bird (or any other interesting animal) being in the certain location in the scene is correlated the time of the day and light intensity (although seasonal information is very important we ignore this information simply because CONE Welder has not been running long enough to generate the

Algorithm 1: Algorithm I: Scanning regions with higher likelihood of birds with a higher zoom level

Input: D_{map}: Generated density-map
Output: F Sequence of suggested frames to be requested from the camera
$MAXTHRESHOLD = .95$;
$MINTHRESHOLD = .05$;
$STEP = 0.05$;
$MAXZOOM = 10$ (maximum possible zoom for the camera);
repeat
  $threshold = MAXTHRESHOLD$;
  $Z_{zoomlevel} = MAXZOOM$;
  $P = D_{map}$;
  while $Z_{zoomlevel} \geq 1$ **do**
    while $\exists p_i \in Ps.tp_i \leq threshold$ **do**
      f = coordinates of a frame that places p_i in the center and has zoom level $Z_{zoomlevel}$;
      request f from camera;
      Add f to F;
      Set $P_f = 0$;
    $Z_{zoomlevel} = Z_{zoomlevel} - 1$;
until *until terminated*;

required data). Therefore, we develop the heat map for each individual bird at each hour of the day. Figure 2.18 illustrates the density-map generated for the White-tail deer for 1pm.

Similar to the level set method, we sort the animals based on their scarceness and assign a weight to each species. Figure 2.19 shows the weighed reward associated with each species for five different species.

Our algorithm then starts choosing frames by employing a weighted sampling method. An species is selected from the reward table by executing a weighted sampling on that table. Then based on the current time the density-map for that specific species is loaded into the memory. Lastly, by taking a 2D sample from that heat-map, the algorithm selects a point from the heat-map as the center of the requested frame. The size of the frame and the zoom level are then chosen by looking at the probability of that specific point and assigning higher zoom levels to higher probabilities.

Based on the "Law of Large Numbers" as the number of requested frames by the algorithm approaches infinity, the probability density function generated by the algorithm for each species at each time will approach the original probability density function for each species.

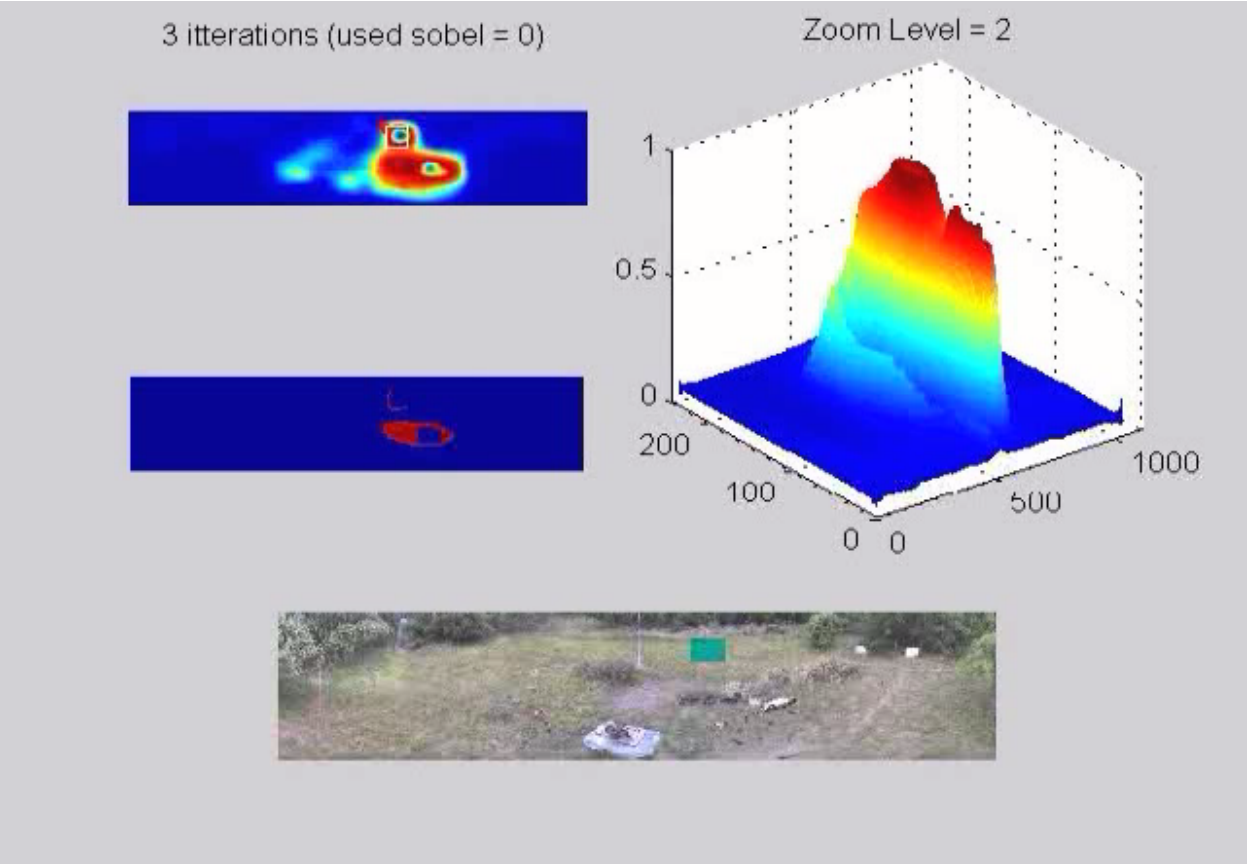


Figure 2.17: The simulation of Algorithm I. The algorithm is running in a simulated environment in Matlab.

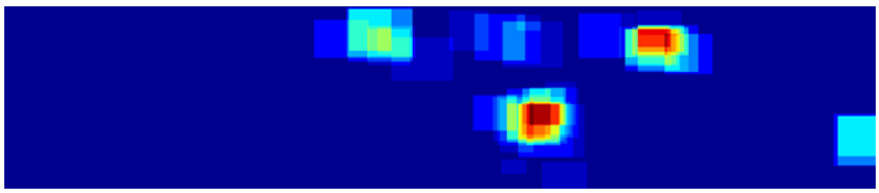


Figure 2.18: Density-map generated for the White-tail deer at 1pm

Algorithm III: Simulation-based method based on pseudo-intelligence

To harness the of the crowed in our algorithms we started to study the strategy of the top birders in the system. We realized that each birder follows a certain strategy. For example algorithm 3 is the strategy of the best birder (user id "vanilla") in the system.

Although a Markov Chain can be a proper tool to model this behavior it involves discretizing the scene into finitely many number of states. Since the grid generation can become

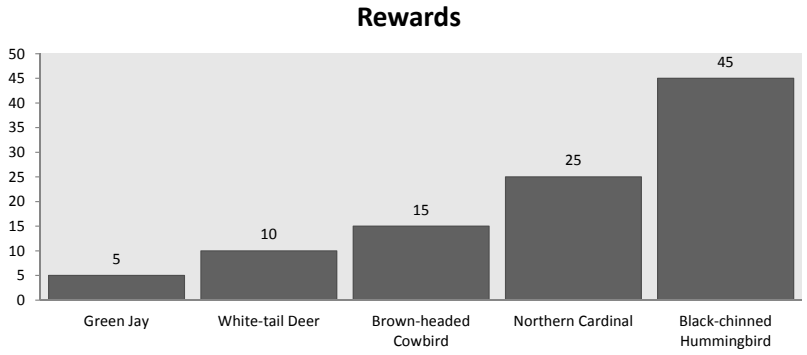


Figure 2.19: Weighted rewards associated with each species

Algorithm 2: Weighted Sampling

Input:

- *time* = time of the day
- Reward table
- Density-maps for all species for each hour

Output: F Sequence of suggested frames to be requested from the camera
while *not terminated* **do**
    species = a weighted sample from reward table;
    load the heat-map of *species* at *time* ;
    center= take a 2D sample of the loaded head-map;
    z =choose the zoom level based on $P(center)$;
    request $f(z, center)$;

very complex we incorporated a gridless alternative by introducing the concept of "Pseudo - intelligence". Algorithm III was one of the models that we wrote and implemented quickly as a first iteration of the autonomous agent. After the deployment on the server this algorithm received a favorable feedback in the chatroom. In this model we assume the probability of finding a bird in a specific frame is a function of the light intensity (time of the day), the season, and the strategy of the player. The algorithm simulates the arrivals of users (described later), and for each user it assumes that the strategy matches one of the five top birders' strategies. These simulated users then request frames and all the requested frames go through our frame selection algorithm and a final candidate frame is requested from the system. This algorithm was implemented on CONE Welder and was well received by users[2].

[2]From their qualitative feedback on the chat room

Algorithm 3: Strategy of the top birder (user id "vanilla")

while *not bored* **do**
- Check the feeders on the mid-right of the panorama;
- Pan at a distance to locate birds ;
- ***If*** found a bird ***Then*** Zero-in for good close-ups;
- Check the water fountain;
- ***If*** found a bird ***Then*** Zero-in for good close-ups;
- Check the tree behind the pond and use the same zoom strategy as the fountain;
- ***If*** No other user in the queue ***Then*** Search the whole edge of the playing field at a moderate zoom;
- ***If*** No animal is found ***Then*** Check the pond quite minutely for smaller animals,frogs,snakes,...;

We modeled website users as a *Nonhomogeneous Poisson Point Process* in which the process of users coming to the system is distributed independently from each other based on a Poisson distribution with variable rate $\lambda(t)$. To find the rate $\lambda(t)$ a Poisson distribution is fit to the historical data using a generalized linear model. Figure 2.8 a Poisson regression done using the Generalized Linear Model Package in R. The histogram for the number of requests of the top birder, "vanilla" is shown in Figure 2.20 .

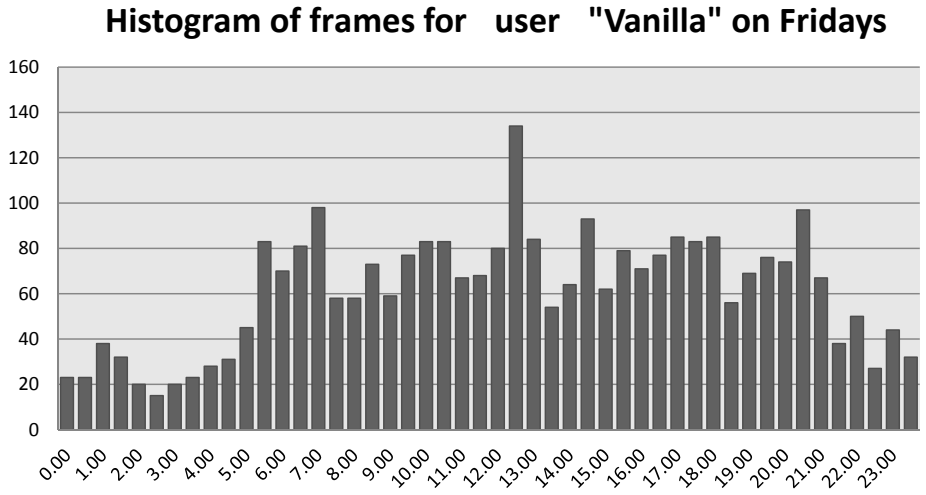


Figure 2.20: Number of requests to the system by the top birder "vanilla".

In the last algorithms we simulate the users coming to CONE system with the rate calculated in Figure 2.8 and we assume that each one of them behaves similarly to one of the top birders in the system. Then the algorithms assume that a portion of these users

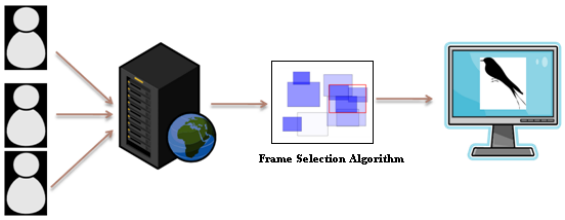


Figure 2.21: Requesting frames using simulated users

submit frame requests to the system. These frame requests are chosen from the historical information in the system at that hour of the day. All the requests are then fed to the frame selection algorithm designed by Song and Goldberg [Song, 2009a] and the final request is then calculated by the frame selection algorithm.

Implementation and evaluation of Algorithm III and qualitative evaluation of results by users

We developed a MATLAB wrapper around CONE Java codebase and we can submit frame requests from MATLAB. The CONE database stores all the information about the camera movement (we only need x,y,z and timestamp), each user can take a picture of what the camera sees and the userID, time taken and the picture name is stored in a table called "Observations". This information is enough to generate the data for Algorithm III. We programmed the algorithm such that it will not take more than 10% of the camera time during the evaluation period and it logged every 10 minutes to operate the camera for one minute.

The performance of the algorithm is perception dependant. Former metrics to define the the objective of the algorithm as a "reward maximizing" model have been shown to be reduced to a *3SAT* problem and not applicable in CONE [Bitton, 2012]. To see how well our algorithm performs we interviewed some of the frequent participants of the system about the performance of the robot and we found that participants are in general happy with the performance of the robot. In one case we were told that they thought that the agent was another new user. In another case they thought that the robot had some good strategies for night and that it searched the well lit areas.

Autonomous agent's snapshots

The goal for the autonomous agent was to make the search more interesting and control the camera when people are just interested in watching the live video. We ran the autonomous agent on CONE Welder side by side with other users and then interviewed our users. The autonomous robot was well received by the CONE community. The following are taken some of the chat history of the CONE Welder. The robot's user ID is "conetester".

(a) (b)

(c) (d)

(e) (f)

Figure 2.22: Results from the autonomous agent

- I think the conetester is a fantastic idea
- Do you think that conetester is being trained to replace us?
- Does conetester have a night strategy? answer: it seems it has a night strategy the robot stayed in illuminated areas

- This is better than most TV programming nowadays

Since the goal of the robot is to provide an entertaining experience for the users on the system it is difficult to measure its performance quantitatively. Although there are some unknown issues with this robot. Since there is no image processing used in this system, some of the snapshots taken by the robot are not focused properly, do not contain the whole animal, or the animal is moving fast and was not recorded properly. Also an animal might be in the picture but since the robot does not get the feedback, it may move to another location in the scene.

2.7 Conclusion and Future Work

Installed at Welder Wildlife Foundation in Texas, CONE-Welder was remarkably robust. It remained online 24 hours a day during its life time, although the camera was offline for maintenance for two periods during the life of the system. Through novel ideas that were implemented in the system such as the game mechanics, the system also provided photo evidence for more than 70 unique species that live in the area. The system also attracted about 30 active participants who participated almost daily. These participants also confirmed the presence of 8 new species whose breeding habitat did not include Welder Refuge. A large corpus of photos that were collected by CONE is available for researchers in various areas such as image processing and ornithology.

We hypothesize that the presence of the new species is due to climate change. This hypothesis is further studied by this author and his colleagues in [Rappole and Faridani., 2011]. We agree that comparing the new crowdsourced results with the traditional bird observation results that were collected thirty years ago may not be an unbiased comparison. We do not claim that the findings from CONE support or reject global warming and that is outside the scope of this chapter. Moreover, we looked at the data on the number of days that each one of the birds of interest was observed in the area [Rappole and Faridani., 2011]. The Buff-Bellied Hummingbirds and Green Jay were seen in 315 and 386 days of the 456 days period. These numbers are significantly higher than the observations for the other species.

CONE is an example of a successful crowdsourced scientific project. During the project more than 45,000 images were taken and labeled freely. Assuming that each image costs 30 cents. The project saved researchers $13,500 and countless hours of their time. And there are many photos in the database that document avian behaviors at night which were not previously available.

(a) (b)

(c) (d)

(e) (f)

Figure 2.23: Some of the photos that were taken by the autonomous agent were out of focus, were not centered, or they violated the privacy of maintenance staff.

Chapter 3

Completion Time Estimation for Crowdsourcing

Online micro task marketplaces like Mechanical Turk[1] provide an online workforce on demand. Mechanical Turk enables people and organizations to crowdsource their tasks that are often hard to do using machine learning or artificial intelligence. Examples of these tasks are writing an essay, solving a CAPTCHA[2], and verifying a physical business address. One research problem in this case is "How much money should a task requester offer for each individual task for people to complete the task". This chapter answers this question.

In this chapter we first provide a classical machine learning approach to estimate completion time through linear regression, and Support Vector Regression. We show that because of the long tail behavior in the market these classical models fail to properly estimate the completion time. Finally, we build a predictive model by using a survival analysis model called "Cox proportional hazards regression" and show how time-independent predictors like the type of the task, the time it is posted, its monetary reward, and even the keywords that are used can influence the completion time.

The model is tested on a dataset that contains 165,368 types of tasks, with the total of 6,701,406 individual subtasks known as HITs[3]. These tasks were posted by 9,436 requesters and were collected by Mechanical Turk Tracker[4] over the period of 15 months. A shorter version of this chapter was originally published in CSDM 2011 [Wang et al., 2011b].

3.1 Introduction

In this dissertation we show a number of scenarios in which crowdsourcing can be useful. CONE was one example for crowdsourcing the process of studying bird migration. Opinion

[1] https://www.mturk.com/

[2] Completely Automated Public Turing test to tell Computers and Humans Apart

[3] Short for Human Intelligence Tasks

[4] http://mturk-tracker.com/

Space is a tool to use crowdsourcing for idea generation. In both examples a dedicated system was built to enable us to crowdsource these tasks. To provide an accessible and scalable crowdsourcing workforce, Amazon started the Mechanical Turk marketplace for crowdsourcing in 2005. In Mechanical Turk a requester can post a task to the open market. Workers can then see the tasks and accept to work on each individual task, known on the market as HITs. After completing the task, the task requester approves or rejects the work. A worker will receive the monetary reward upon approval.

More recently researchers have started using this market as an alternative platform for running experiments [Paolacci et al., 2010], online businesses have used Mechanical Turk to enhance their platform capabilities, even in search and rescue scenarios [Keller, 2010]. Bernstein et. al deployed a Microsoft Word plug-in, Soylent, that allows writers to use the Mechanical Turk's workforce to enhance their writing [Bernstein et al., 2010]. Soylent allows a writer to automatically shorten text, expand text and even change active sentences to passive. Similarly Bigham et. al built, VizWiz, a phone application that allows blind people to find different objects, differentiate colors and read texts in real time [Bigham et al., 2010a].

For any requester who posts tasks on Mechanical Turk, it is important to have an estimate of the completion time beforehand. Mason and Watts study the effect of financial incentives on the quality of completed tasks [Mason and Watts, 2010]. They have shown that the quality of a completed task is not correlated with the monetary reward that is paid to the workers. However, they show that the quantity of the work done by workers increases as the monetary reward for the task increases. We later study what factors contribute to completion times. For example by using an LDA[5] topic model we show that tasks related to transcribing are getting done much faster than other tasks.

3.2 Data Set

All the analysis for this chapter is done on the data collected by MTurk Tracker software[6]. MTurk Tracker crawls the Mechanical Turk website every half hour and records information about the tasks that are still available on the market. The source code for MTurk tracker is available to the public[7]. Mechanical Turk shows all the tasks that are on the market in a linear list 3.1. In addition to the information about each task, MTurk Tracker records the location in the page that each task is posted. For more information about this crawler see the work by Ipeirotis [Ipeirotis, 2010a].

The dataset used in this work was collected using MTurk Tracker from January 2009 through April 2010. There were 6,701,406 individual HITs for 165,368 different tasks. The dataset contains $529,259 worth of crowdsourced work from 9,436 requesters. The completion time for the task (known as HITgroup in Mechanical Turk) is the time before the task

[5]Latent Dirichlet Allocation

[6]http://mturk-tracker.com/

[7]A copy of the source code is available at https://github.com/faridani/Mturk-Tracker, more up to date versions are available at https://github.com/10clouds/Mturk-Tracker

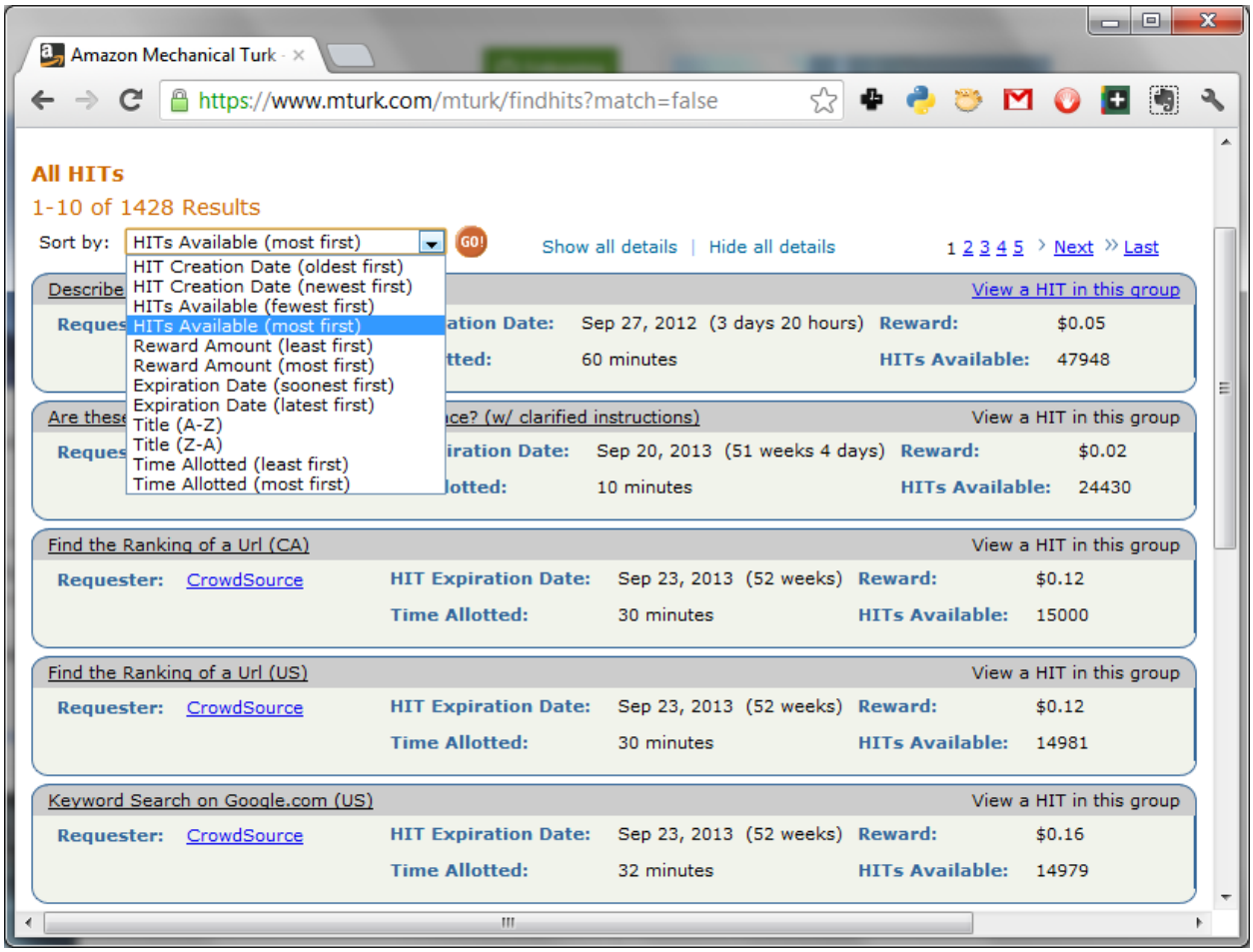


Figure 3.1: The tasks that are posted on the market are sorted based on recency. Mechanical Turk sorts the available tasks according to different factor. This causes some of the tasks to be less visible to workers.

disappears from the market. The distribution of the completion times in Mechanical Turk is heavy-tail, meaning that there are tasks that stay on the market for a long time. These tasks are called "starved" tasks. For derivations and justifications for the long tail behavior of the market see our paper [Wang et al., 2011b]. In cases that this long tail behavior exists, using the sample mean is never a good estimate of the completion time.

In this work, completion time for each HIT group is calculated by finding the number of hours between the first time that HIT group was observed on the market until the last time. Figure 3.2 is the histogram for the frequency of HIT groups with different number of HITs. Bumps at point $x = [1, 10, 20, 30, 40, 50, 100]$ show that requesters tend to post HITs with these round numbers. Figure 3.4 shows the number of HIT groups posted on different hours of the day. Figure 3.3 shows the number of HIT groups posted in each day of the week.

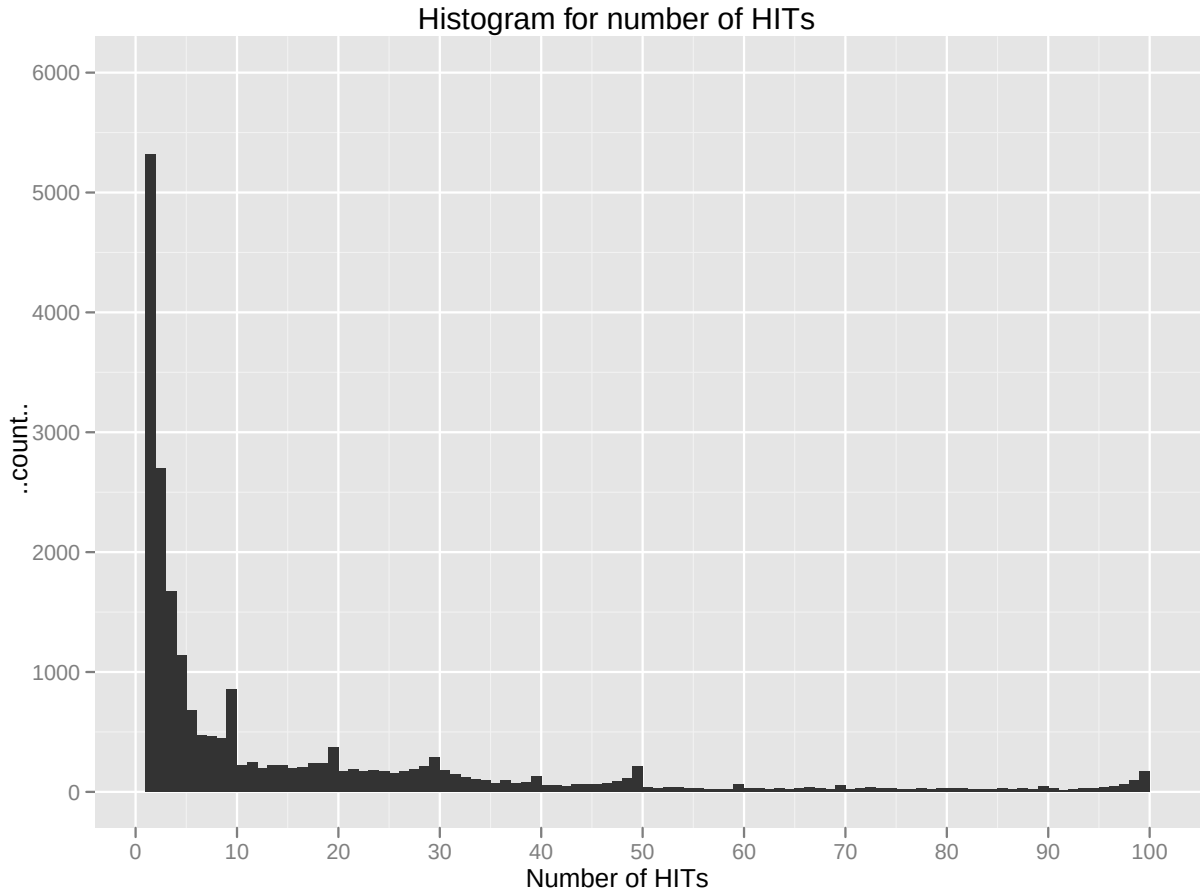


Figure 3.2: Histogram for the number of HITs for individual HIT groups. Point X=0 is excluded from the graph as its value, 100,723, is higher than the rest of the graph.

3.3 Using traditional machine learning algorithms to estimate the completion time

As a preliminary step in this project we used Weka data mining software [Goebel and Gruenwald, 1999] to predict the completion times for the tasks in the dataset. We have split the dataset into a training set with 66% of the HIT groups and the rest are used as the test set. We were not able to obtain a reliable and consistent result from Weka. For example we realized that the *Mean Absolute Error (MAE)* changes when we use another set of recorded data. Also the order of most successful methods changed when we switched our validation algorithm from Test set/ Training set to cross-validation. Therefore, we were not able to highlight one reliable algorithm for successfully predicting the completion times. One explanation for this problem with off-the-shelf and general purpose machine learning models

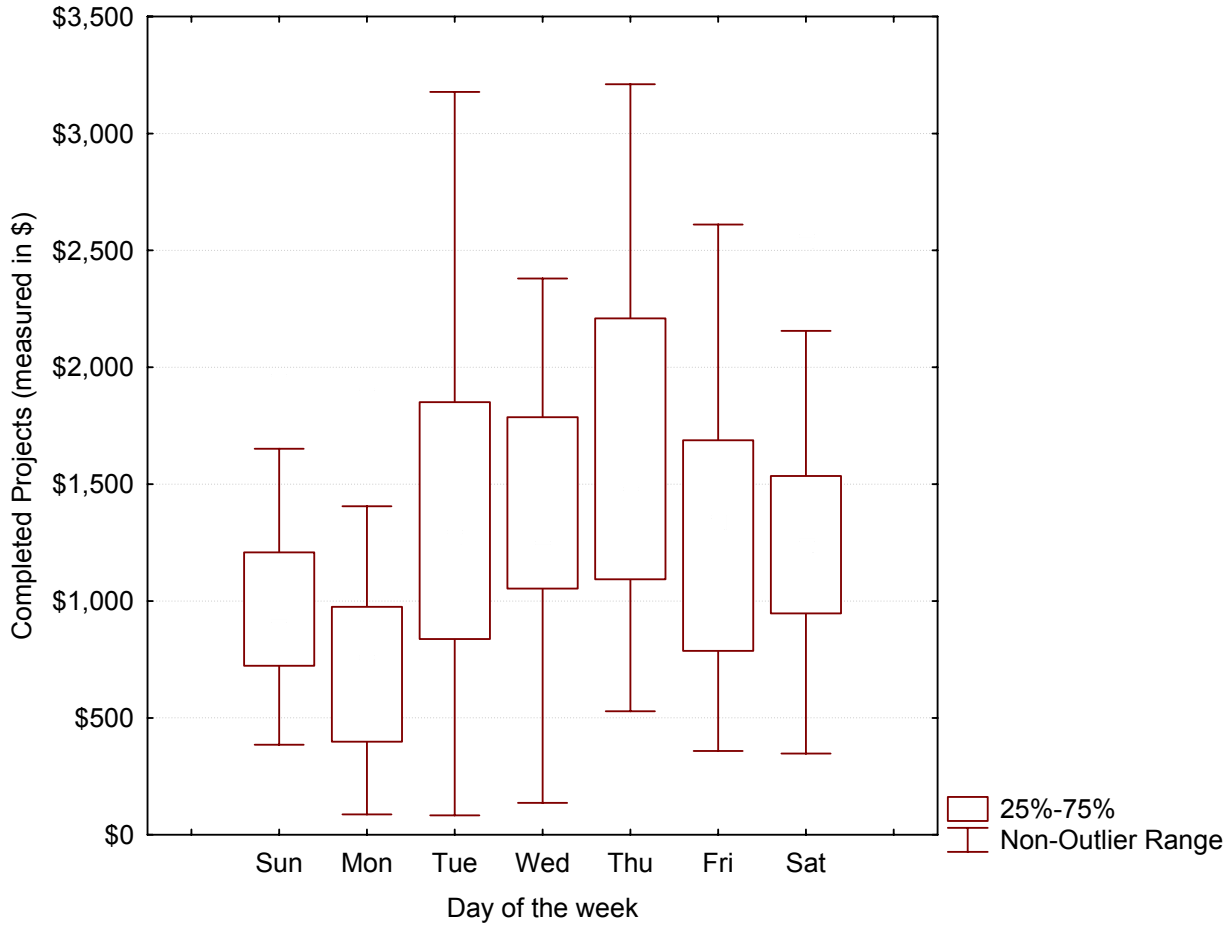


Figure 3.3: Number of HIT groups posted by requesters in different days of the week

might be that many traditional machine learning and regression models assume Gaussian distribution for either predictors or the latent variable. As a result of the long tail distribution many off-the-shelf machine learning models fail to predict the expected completion time properly.

Weka results also do not provide any insight into how the market behaves. As a result we use a survival analysis algorithm based on Cox proportional hazard to predict the completion times for tasks in our dataset. We then provide a theoretical model to explain Turkers' behavior in the market.

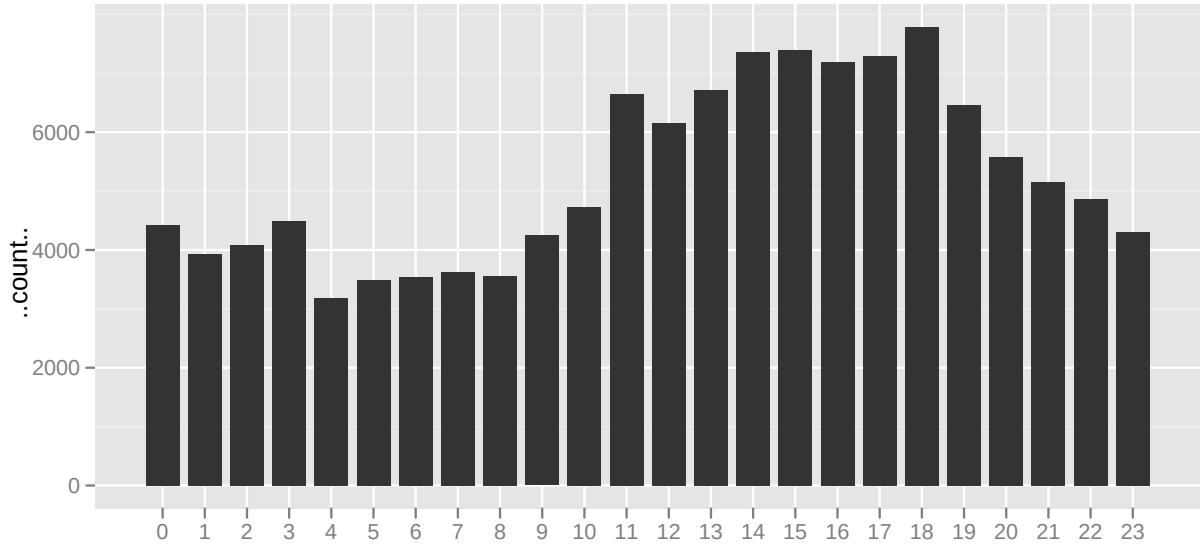


Figure 3.4: Number of HIT groups posted by requesters in different hours of the day

3.4 A statistical model for the long-tail expected completion time

We built a predictive model for the completion times using a survival analysis method called "Cox Proportional Hazard" model. Since completion times do not follow a normal distribution, standard linear model are not appropriate tools for estimating them. Used often in biology, epidemiology, and maintenance scheduling, survival analysis looks at the lifespan of one entity [Dalgaard, 2008, Kleinbaum and Klein, 2005, Fine et al., 2003]. In this section we look at the lifespan of a crowdsourced task in Mechanical Turk.

Predictors in the model

We use three types of time-independent variables as predictors. An example of time-independent variables would be the number of subtasks (HITS) in a task. These predictors can be classified into three categories:

- **Characteristics of the requesters:** historical information about the activities of the requester such as (the number of days since first activity on the market, total amount of money paid as rewards, total number of HITS, average value of prior jobs posted on the market, and the average completion time for prior HITs.
- **Characteristics of the market:** We use a number of characteristics of the market at the time that the task was posted. The day of the week, time of the day, and the

size of the market (number of competing jobs on the market)

- **Characteristics of the job:** The reward amount in U.S. dollars, number of subtasks (HITs), keywords, and also the main topic that was presented in the task and was extracted by using Latent Dirichlet Allocation (LDA) [Blei et al., 2003b].

Using LDA topic models as a predictor

The keywords of the tasks were analyzed by using the LDA model [Blei et al., 2003b]. LDA assumes that a document is a mixture of topics. Each task in this chapter is assigned only one topic (the topic with the highest probability). The number of topics were manually selected to be seven topics based on the types of tasks that were present in the market.

Censoring in the market

A task is called "Censored" when it is taken off the market without completion. This phenomenon happens in Mechanical Turk as requesters cancel the tasks that are starving[8]. The fundamental assumption of the survival analysis models are that the completion times are independent from censoring times. We will provide more details of survival analysis and Cox proportional hazards model in the next chapter. A Cox proportional hazards model was fit to the data scraped from Mechanical Turk and is shown in Figure 3.5. As we see, approximately 75% of the tasks are completed within two days of the start.

Stratified model fitting

One of the traditional ways of comparing results for different task characteristics in this model is to use stratified survival analysis. This analysis is done for variables like reward, number of subtasks (HITs), day of the week that the task is posted, and the topic[9].

The analysis showed in Fig. 3.7 illustrates that the survival rate varies significantly with varying the HIT characteristics like reward and number of HITs in a HITgroup. While we see that the rewards and number of HITs have a large and significant effect on the completion, we do not see a significant effect when we look at the effect of the day of the week on the completion. We conjecture that this is because we look at the tasks that take too long to complete. In the next chapter we look at another model that validates these results again. For a survey of techniques for tasks with short completion time see [Bigham et al., 2010a].

[8]Tasks that are not completed but are known to not attract workers. Requesters often cancel these tasks and repost them.

[9]Based on the LDA results

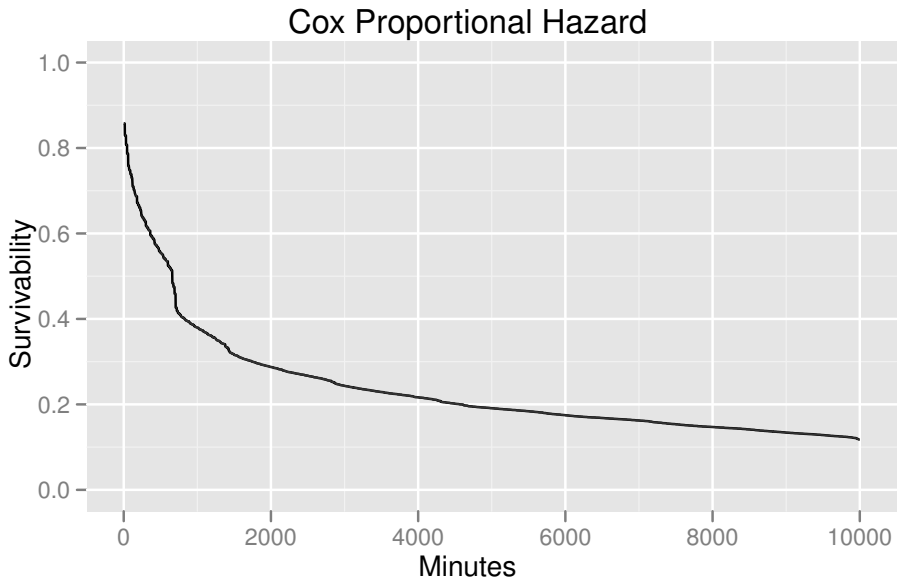


Figure 3.5: Cox proportional hazards regression model was fitted to the data scraped from Amazon Mechanical Turk

3.5 Prediction

The framework that is presented in chapter enables us to isolate and study the effect of one variable on the task completion time. In this section a couple of scenarios are presented and studied. The following are the tasks that are considered here:

- A task with 3000 HITs that is from "Topic6"
- A task with 30 HITs and still from "Topic6"
- A task with 30 HITs and from "Topic1"

For this experiment we have fixed all the other variables at the following levels:

- The tasks are posted on Monday
- The time that the task is posted is fixed at 6pm
- A 20 cents HIT is used

Figure 3.6 demonstrated the prediction results. As we see transcribing tasks are finished more quickly. CastingWords, the company that is posting these tasks on Mechanical Turk, is a long-term active user of the market. In Mechanical Turk a worker first accepts a job and then finishes it. We unfortunately do not have any data on how long each HIT is being

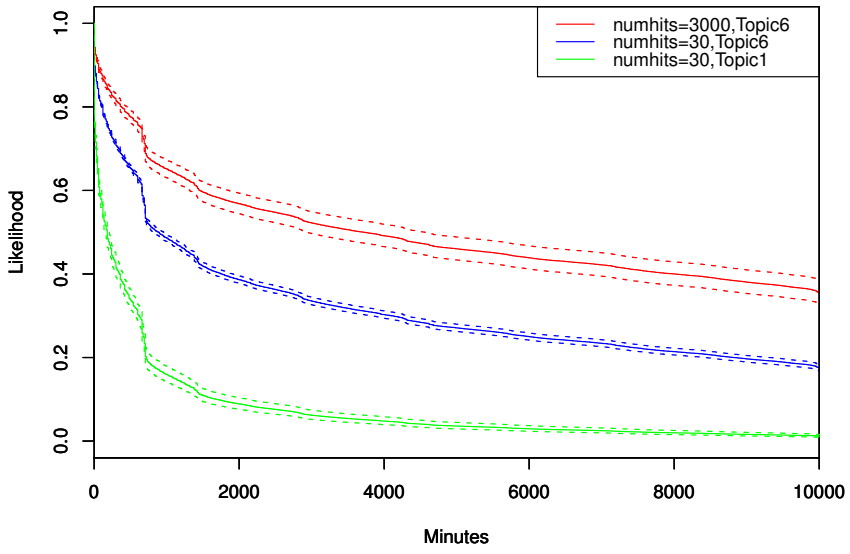


Figure 3.6: The probability that the tasks are alive on the market for three different tasks are shown over time. Error bounds are also shown around each survival curve.

worked on. It is suggested in [Kochhar et al., 2010] that because of different difficulty levels for the tasks, these durations follow a log-normal distribution.

3.6 Evaluation of the Model

The dataset is divided into two equally large training and test sets. The Cox model parameters were trained on the training set and tested the learned parameters to calculate the completion times of the HITs in the test set. Because of the non-Gaussian completion times, using typical error measures like ERMS or MSE is not ideal. We use a likelihood ratio test (LR) to measure the goodness of fit. Cox model parameters $\hat{\beta}_{(train)}$ that are trained on the training set are then used to estimate the Cox log partial likelihood $l^{(test)}\hat{\beta}_{(train)}$ for the HITs in the test set. For the null hypothesis the likelihood $l^{(test)}(0)$ is computed which means we predict a constant value for task completion time . The LR statistic is 8434 and larger than $\chi^2_{25} = 37.65$ which demonstrates a good model fit.

3.7 Conclusion and Next Steps

We showed the long tail behavior of completion times in the Mechanical Turk market. Because of this a simple averaging cannot give an accurate estimate of the completion time.

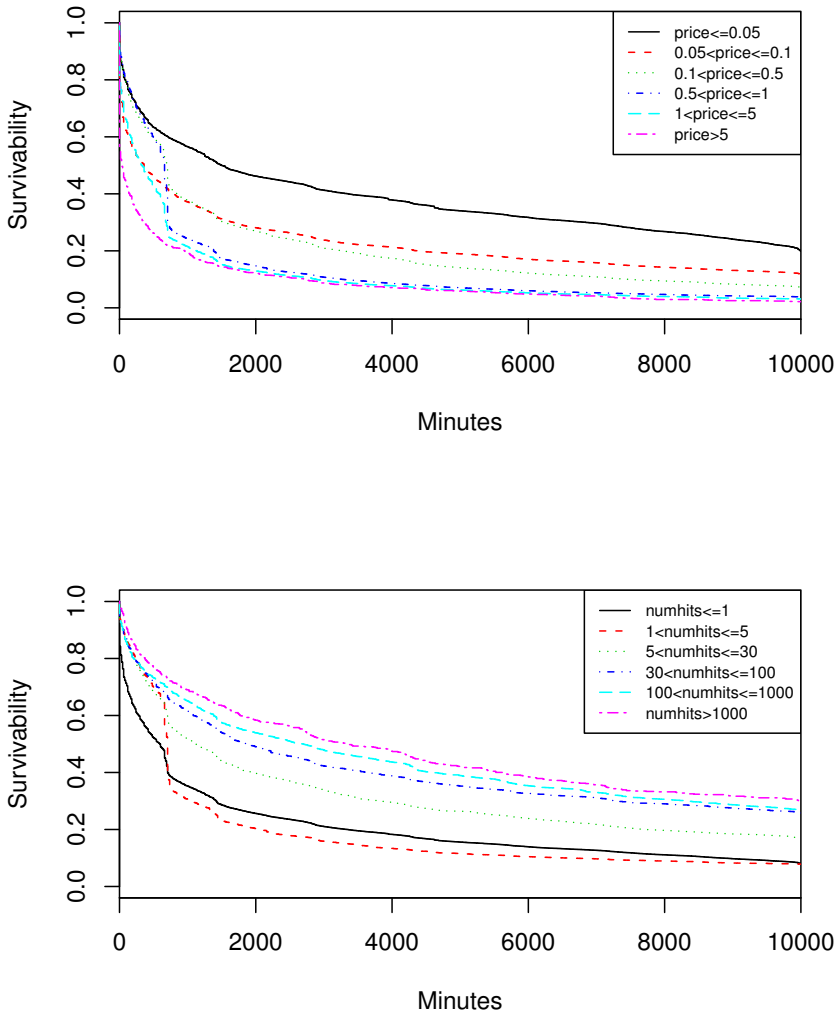


Figure 3.7: Stratified analysis for reward value and number of subtasks (HITs)

The Cox survival model that is proposed in this chapter is only one way of estimating this value. In the next chapter we approach this problem from another view and show that we can use the framework for both pricing and completion times.

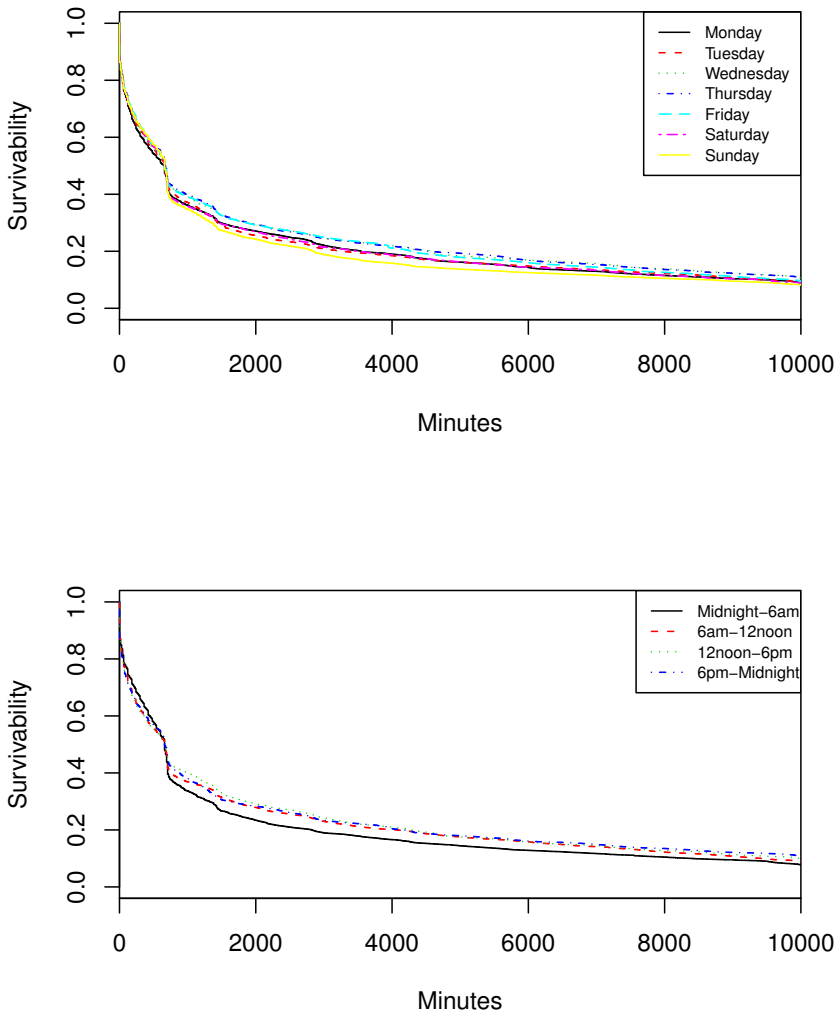


Figure 3.8: Stratified analysis for day of the week that the task was posted to the market and time of the day that the task is posted to the market.

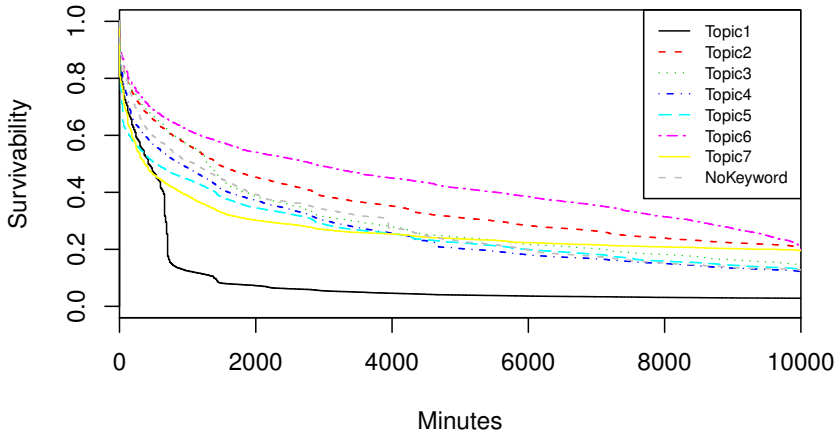


Figure 3.9: Stratified analysis for reward value, number of subtasks (HITs), day of the week that the task was posted to the market, time of the day that the task is posted to the market, and HIT topic based on the LDA model.

Chapter 4

Pricing Crowdsourced Tasks for Finishing on Time

Many practitioners currently use rules of thumb to price tasks on online labor markets. Incorrect pricing leads to task starvation or inefficient use of capital. Formal pricing policies can address these challenges. In this chapter we argue that a pricing policy can be based on the trade-off between price and desired completion time. We show how this duality can lead to a better pricing policy for tasks in online labor markets. This chapter makes three contributions. First, we devise an algorithm for job pricing using a survival analysis model that is discussed in the previous chapter. We then show that worker arrivals can be modeled as a *non-homogeneous Poisson Process (NHPP)*. Finally, using NHPP for worker arrivals and discrete choice models, we present an abstract mathematical model that captures the dynamics of the market when full market information is presented to the task requester. This model can be used to predict completion times and pricing policies for both public and private crowds.

4.1 Introduction

One of the most important challenges for task requesters on crowdsourcing markets like *Amazon Mechanical Turk (AMT)* is to properly price and schedule their tasks (or "HITs," which stands for "Human Intelligence Tasks"). Improper pricing or scheduling often results in task starvation and loss of capital on these markets. For example, it is believed that workers have an expected hourly wage in mind and they tend to not accept *underpriced* tasks that need more time per unit reward than what they have in mind. Tasks that are not accepted stay in the system. (These tasks are often called *"starved HITs"*.) Starved HITs may be canceled or reposted by the requester resulting in expenditure of more time and money than planned for the task. Overpriced tasks are also undesirable since requesters can invest excess capital in quality assurance for the data that they have collected. By using a survival analysis model we devise an algorithm for determining the optimal reward for a crowdsourced task.

Even though we focus just on reward setting on this chapter, our approach is generalizable and practitioners can use it to optimize the other task characteristics, such as task length, bonus, and even keyword selection for their tasks.

Survival analysis can yield the expected task completion time and optimal reward for a candidate task by using a model that is trained on the historical data of the market. However, survival analysis provides no insight into how the market works, how workers arrive to the system and how they decide to perform a task. In the second half of the chapter we focus on the worker/task dynamics that characterize individual workers. For this second section, we assume that requesters are exposed to complete information about the market — they can access snapshots of the tasks posted on the market and get information about task completion by individual workers. Private crowds are examples of these type of markets where requesters often have access to historical data about arrivals and task choices for the workers. By looking at quantitative data from Mechanical Turk we show that worker arrivals can be modeled with a non-homogeneous Poisson Process (NHPP).

Building a proper model for worker behavior also requires a descriptive model of how workers decide to take on and finish a task. Workers often select their tasks from a desirable task pool. Our observation shows that workers often have preferences for the types of tasks they like to accept. We use this concept to develop a *discrete choice based model* for a better pricing policy and scheduling for crowdsourced tasks. In cases where complete, or even partial, information of the market is available, a requester can optimize her task attributes to increase the likelihood of workers accepting the task. Discrete choice models can provide a framework to optimize the attributes of a task and therefore increase its desirability to the user. One convenient aspect of discrete choice models is that this change in desirability can be captured, quantified, and used for attribute optimization.

4.2 Terminology and Definitions

Before continuing we define some of the terminology used in this chapter. We define ***workers*** as individuals who accept tasks on a crowdsourcing market. A crowdsourcing ***market*** is the place, usually an online website, where workers find and perform ***tasks*** often for a financial ***reward***. In the literature, workers are occasionally called ***Turkers***, a description of workers who perform tasks on ***Amazon Mechanical Turk (AMT)***. Crowdsourced tasks are posted to the market by individuals or companies. In this work, the entity that posts tasks to the market is called a ***requester***. A task may be composed of atomic subtasks, ***HITs (Human Intelligence Tasks)***. HITs are completed by workers. Some of the HITs are never picked by workers and they stay on the market until canceled by the requester or the market administrator. We call these ***starved HITs***. The ***optimal reward*** is the minimum amount of money that the requester can pay for each HIT and still have the task ***completed*** by his desired completion time.

4.3 Data Set

We have been monitoring the AMT marketplace and taking snapshots of the market since January 2009. For a description of the process and of the dataset, please see [Ipeirotis, 2010b]. For the purpose of this chapter, we used a smaller dataset, containing 126,241 HIT groups from 7,651 requesters. Our dataset contains 4,113,951 individual HITs that are worth $344,260. We use *Latent Dirichlet Allocation (LDA)* and requesters' selected keywords to capture the type of the work [Blei et al., 2003a]. The reputation of the requester is accounted for by using their historical number of posted HITs, amount of rewards that they have spent in the market, the number of different HIT groups that they have used and the first time that the requester has posted to the market. Market condition is also captured by counting the number of competing HIT groups and competing rewards that were available when each HIT was posted.

4.4 A Brief Introduction to Survival Analysis

Survival analysis, frequently used in epidemiology and biostatistics, is a general term for statistical techniques used to estimate the time until a particular ***event*** occurs. Time can be represented in any units (hours, minutes or years). What constitutes an ***event*** depends on context. For instance, in epidemiology an event usually refers to the *death* of the individual. In the context of maintenance scheduling, an *event* can be referring to a machine breakdown. A ***survival function***, $S(t)$, is the probability that the survival time is longer than t. The survival function $S(t)$ is often defined through a hazard function $h(t) = -\frac{S'(t)}{S(t)}$, with $S'(t)$ being the first derivative of $S(t)$. The hazard function captures the rate of death at time t, across the population that survived until that point. A Cox proportional hazard (CoxPH) model is a ***semi-parametric*** model in which the hazard function for an individual with predictors $\mathbf{X}$ is defined as:

$$\log(h(t, \mathbf{X})) = \log(h_0(t)) + \sum_i \alpha_i \cdot X_i \tag{4.1}$$

where $h_0(t)$ is the "baseline hazard function" and can have an arbitrary form.

In the basic form of CoxPH, the predictor variables are assumed to be *time-independent*. Extensions of the Cox model use time-dependent predictors [Kleinbaum and Klein, 2005]. In our work, we used the CoxPH implementation available in R that considers time-dependent variables and multiple events per subject, by using the counting process formulation introduced by Andersen and Gill [Andersen and Gill, 1982].

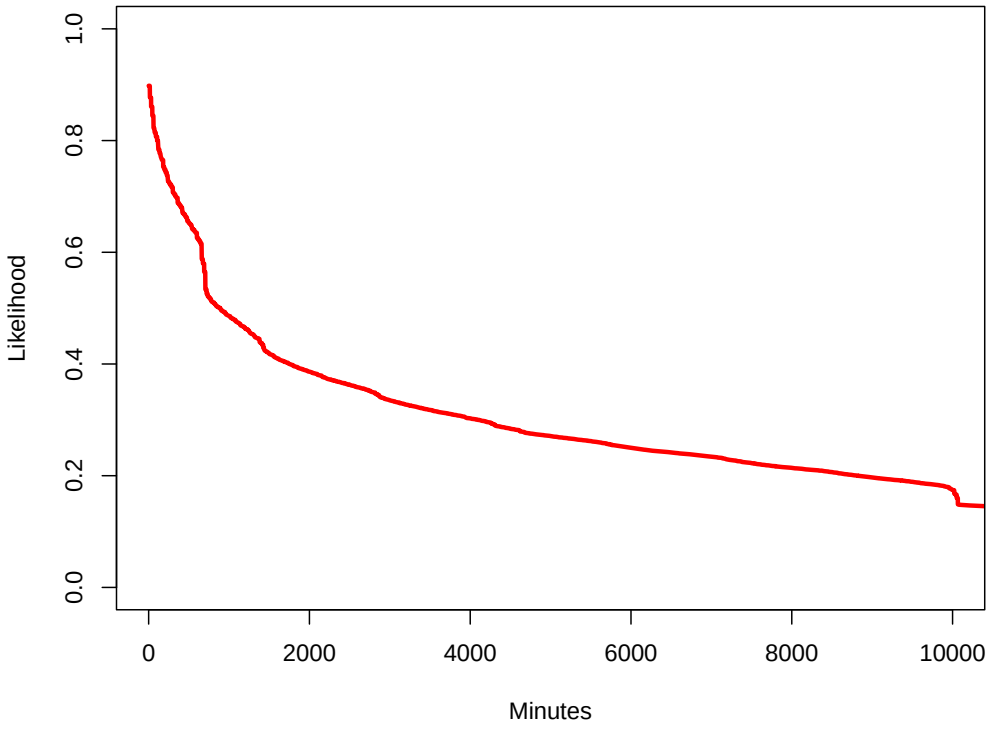


Figure 4.1: Survival curve for a crowdsourcing task with reward = \$0.25, with 30 HITS, from a requester that has posted \$1,341 worth of tasks and 7100 total tasks. The task was posted on a Monday with 917 other competing tasks on the market.

4.5 Pricing Policy Based on Survival Analysis

In Figure 4.2, we vary the reward from 1 cent to 1 dollar and calculate the expected completion time for the task described in Figure 4.1. A continuous curve is fitted to data points for better visualization. As we see, the graph is monotonically decreasing for increasing values of the reward. This behavior, in conjunction with the desired completion time, can be used to develop a procedure to determine the price for a task, in order to finish right before the desired completion time. Algorithm 4 shows an optimization algorithm for finding the price for a crowdsourcing task.

Algorithm 4 uses a bisection search on the results of a Cox proportional hazard model to find the appropriate reward for the desired completion time. The appropriate reward is defined as the minimum value of the reward that ensures the completion by the desired completion time t_{max}. In this example we have only used the most important attribute (reward) but this approach can be easily extended to a multivariate optimization model that includes other attributes of the task like number of HITs and even keywords.

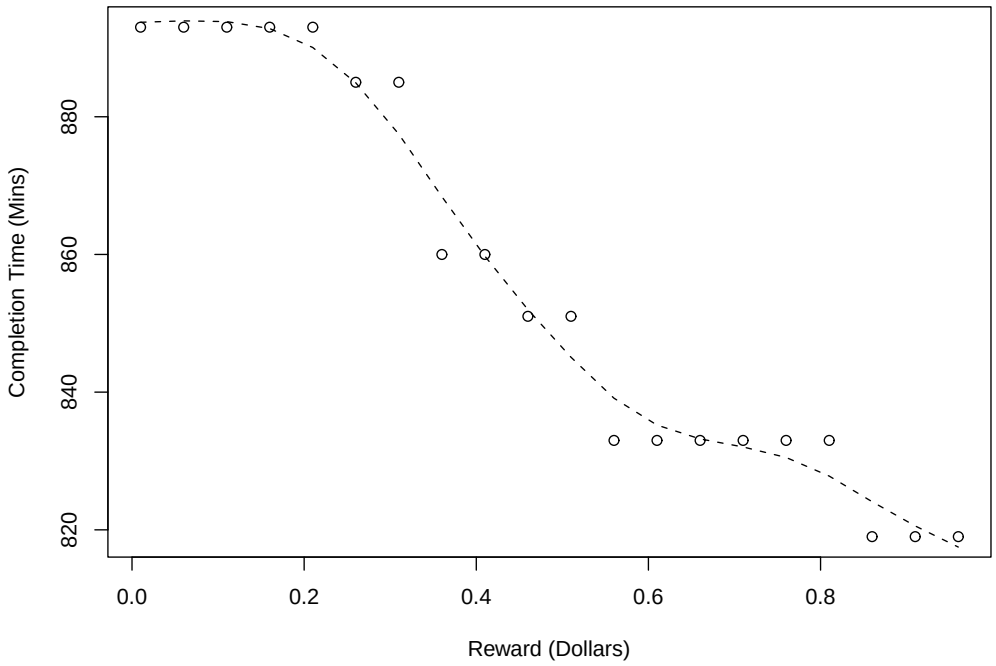


Figure 4.2: Expected completion times to the task presented in Figure 4.1 when the reward varies from \$0.01 to \$1.00. The curve is monotonically decreasing.

4.6 Towards a Better Theory Model for Market Behavior

In the previous section, by using a CoxPH model, we provided an algorithm for a pricing policy based on the attributes of the task. The procedure uses the trade-off between the reward and desired completion time to come up with the lowest reward that ensures a proper completion for the task. CoxPH is used as the module that provides the values of completion times to the algorithm. In this section, we further study the market dynamics and provide a model that can eventually replace the CoxPH model in our algorithm. We first focus on worker arrivals and show that worker arrivals to a crowdsourcing labor market follow a *Non-Homogeneous Poisson Process (NHPP)*. We then show that the likelihood of the task being selected by a worker can be maximized by using a discrete choice model such as the multinomial logit model (MNL). In order to find the completion times of a certain task, we can simulate the NHPP arrivals and simulate their choice behavior based on the MNL model. Providing a closed form equation for completion times is out of the scope of this chapter, but is a topic of interest for future research.

4.7 Stochastic Arrival Model

Figure 4.3 shows the amount of activity for workers based on the day of the week. The result indicates different levels of activity, and suggests that the assumption of time homogeneity is not justified. To alleviate the assumption of homogeneity, we consider a *non-homogeneous Poisson Process (NHPP)* as the arrival model. This means that workers arrive at the labor market according to a Poisson model with a varying rate $\lambda(t)$. Unlike the Poisson model, in a *NHPP* arrivals of two workers are not independent and they both depend on a latent variable, *time* t.

Traditionally used for counting data, *Poisson regression* is a subclass of *generalized linear* models where we fit a distribution from the exponential family to experimental data. Generalized Linear Models were introduced as a regression tool for the random variable of the exponential family of distributions [Nelder and Wedderburn, 1972]. This family includes the normal, Poisson, Gamma, inverse Gaussian and binomial distributions. Many statistical methods are a subclass of a generalized linear model. To formulate this problem we first use classical stochastic process arguments to show that worker arrivals to a labor market can be modeled as NHPP arrivals.

Worker arrivals to the labor market are NHPP

A Poisson distribution is used for modeling counting processes. We show that worker arrivals to an online labor market follow a NHPP process. Using empirical data, [Gunduz and Ozsu, 2003] showed that the number of visits to a web page can be modeled with a Poisson model. Faridani et. al. [Faridani et al., 2009] study their private crowd of volunteers and demonstrate an NHPP model for their worker arrivals.

Poisson Regression

Poisson regression is a subcategory of the generalized linear model in which a non-homogeneous Poisson distribution with a varying rate $\lambda(t)$ is fit to the data. The goal of regression is to find the value of the function $\lambda(t)$ for different values of t.

In Figure 4.3 we have used a Poisson regression to fit a regression line to more than 131,000 worker arrivals to Amazon Mechanical Turk. This number of workers only represents a portion of the workers on the market but we can argue that this is a *thinned* Poisson process. As a result the original arrival of workers to the market is also a Poisson model. (The superposition of Poisson processes is also Poisson.)

4.8 Choice Based Crowd Dynamics

In this section we look at the discrete choice model presented in Train[Train, 2003]. We also explain how an agent makes decisions in the DCM framework. The decision of an agent is influenced by both observed and unobserved factors. To be consistent with Train's notation

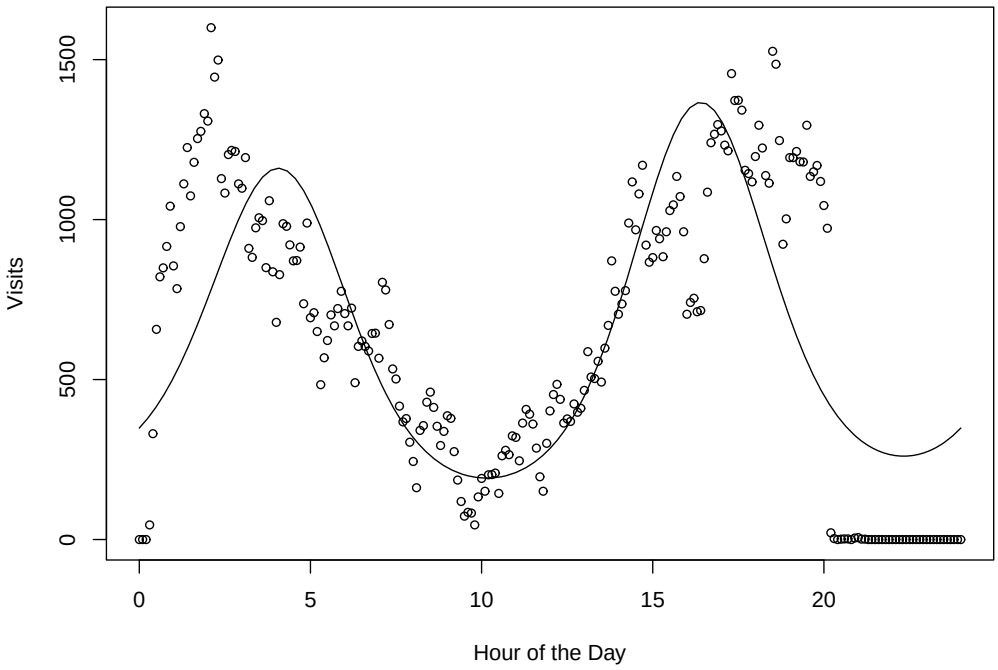


Figure 4.3: Fitting a NHPP to visitors' data retrieved from one of our tasks on Mechanical Turk. $\lambda(i)$ is found by using GLM.

we use x for observed factors (or attributes) and we use ϵ for unobserved factors that are unknown to us. These two sets of factors determine an agent's choice through a *behavioral process* function $y = h(x, \epsilon)$. Note that if x and ϵ are known, the value of h is completely known to us. Although in the DCM model the ϵ factor is not observed, and thus probabilistic approach is evident. In this case ϵ is considered to be a random variable with density $f(\epsilon)$.

Discrete choice models have been used extensively in economics and revenue management [Train, 2003, Vulcano et al., 2008, Vulcano et al., 2010]. A discrete choice model assumes that the worker has a choice of tasks to work on, and chooses the task that optimizes the worker's own "utility," whatever the definition of utility may be for each worker (e.g., it may be the hourly wage). The model adopts utility as a modeling concept, and explicitly assumes that utility is latent and we only observe the actions that are the result of optimizing this utility. In our case, the different choices for the workers are the different tasks on the labor market, with different attributes. Attributes include the reward for the task, time of the day that the task is posted, number of total HITs in that task, and other properties. These attributes make a task more desirable or less desirable for a worker. Of course, the worker's decision to accept a task is also influenced by the other tasks that are available on the market at that moment. For example, a worker may decide to not accept a transcription task for $1 if a survey task with a $1 reward is available on the market. However, the same $1 transcription task may be desirable if the only tasks available in the

market are other, comparable transcription tasks worth 50 cents are available. We may also have the case that a worker may decide not to accept any tasks and leave the market without completing anything.

This *dependent behavior* can be modeled with discrete choice models. One aspect of such models is that the likelihood of accepting a task can be updated as the attributes of available tasks on the market change. We assume that workers are utility maximizers and in order to capture the preference behavior of workers we use a *logit model*. In this work we assume that the crowd is homogeneous in terms of task preferences. An extension of this model, explicitly modeling the fact that there are groups of workers with different preferences and skills is the BLP model [Li et al., 2011a]. While we do not cover BLP-style models in this work, it is definitely a direction for interesting future research [Berry et al., 1995].

4.9 Choice Based Model for Task Selection for Crowd Workers

In the previous section we showed that workers arrive to the system according to a *NHPP* process. The question that we answer in this section is "How do workers select a task to work on." In our framework, as described above, we assume that workers are ***utility maximizers*** and work toward maximizing the value of their own utility. One of the advantages of this viewpoint is that it does not require information about individual decisions of workers (such information is not observable on platforms like AMT) but relies on just observing the aggregate output to infer the aggregate preferences of the workers. Aggregated market data can be used to estimate the weights of individual attributes in the aggregated utility. To analyze the choice model for workers, we define two utility values:

- *Utility of Task*: The utility that the worker will gain by selecting and finishing a task, which is typically the utility of money earned, but also can include aspects such as learning something, having fun, etc.
- *Utility of Time*: The utility that the worker will lose by spending time on a task (and thus not being able to accept and finish other tasks).

Intuitively, a worker works on a task only if the *utility of task* is larger than the *utility of time*. The assumption of a rational worker implies that workers select the task that maximizes the difference between these two values.

Assume that n tasks are available on the market and denote j as the index of task H_j. If we assume that a worker has T units of time at hand to spend on working on tasks, then we denote the utility of task for task j as $U_h(X_j)$ and the utility of time as $U_t(t_j)$ in which t_j is the amount of time that takes for the worker to finish task j. The *rational worker* assumption implies that workers maximize the value of $U_h(X_j) - U_t(t_j)$. In this formulation X_j is a multidimensional attribute vector for the task ($X = \langle x^1, ..., x^k \rangle$). For our analysis, we consider $U_h(X_j)$ to be a linear combination of weighted utilities from

observable task attributes, plus an unobservable stochastic term ξ to capture the utility of the unobserved characteristics of the task and is typically assumed to be independently and identically distributed according to a Gumbel distribution. In this case, the utility of tasks are formulated as:

$$U_h(X) = \sum_{k=1}^{K} \beta^k x^k + \xi \tag{4.2}$$

Note that β is positive for desirable attributes (e.g., number of HITs) and is negative for undesirable attributes (e.g., required time to finish the task). To make the formulation simpler we assume that the utility of time is a term with negative β in the utility of task (Equation 4.2). The main challenge for this formulation is to estimate the value of parameters β from recorded market information.

4.10 Homogeneous Workers Model (Logit Model)

We assume that the crowd is *homogeneous* meaning that the values of β are the same for all workers [Train, 2003, McFadden, 1972]. Workers arrive to the labor market according to a non-homogeneous Poisson Process and each of the workers selects a task from available tasks with a certain probability that is determined by a logit model. For example the worker w is then presented with a set C_w of tasks to work on. The worker can also decide not to accept any task, an option that we denote with C_0. In our setting, the probability that worker i decides to work on task j is:

$$P(choice_j^i) = \frac{e^{\beta X_j^i}}{\sum_{j \in C_w} e^{\beta X_j^i} + 1} \tag{4.3}$$

In Eq 4.3 the number one in the denominator is due to the zero utility for C_0 cases when worker decides to not accept any tasks ($e^{\beta \cdot 0} = 1$). For homogeneous workers Equation 4.3 is equal to the market share of the task j. McFadden shows that β values can be found by using a logistic regression [Li et al., 2011a].

Results

Figure 4.4 shows our preliminary results for the model. As expected, as we increase the number of HITs for a task it becomes more likely that it will be picked up by workers, resulting in increased demand for the product. Increasing the number of competing projects decreases the demand for the task. A potentially surprising outcome is that increasing the reward decreases the demand for the task. While this appears counter-intuitive, it is the direct result of the choice model for the market. High reward tasks usually mean more complex and more involved tasks and that decreases the utility of high reward tasks for the worker. Effectively, we observe the Simpson's paradox in our analysis. In the future, we

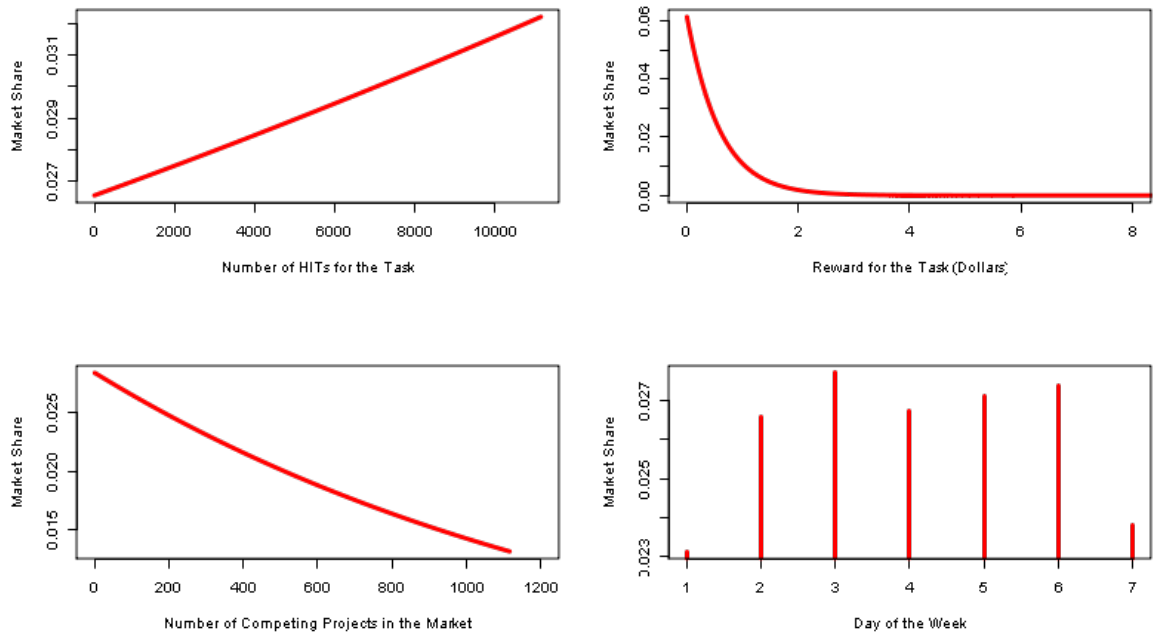


Figure 4.4: Training a logistic regression model on the market data. Plots show predictions for a typical task with a 50 cent reward that contains 100 HITs and is posted on a Monday on 9AM where there were 100 other competing projects on the market. Graphs depict results of experiments where we have varied each predictor and predicted the likelihood

are planning tease out these confounding factors, by incorporating a topic model that will capture the inherently difficulty of each available task.

Our choice model can now be used to price tasks. Instead of changing prices for survival time, we can change prices to adjust demand for the task. By simulating NHPP arrivals and simulating demand for the task over time, we can achieve the same effect and price the task for being completed on time.

4.11 Conclusion

Heavy tail distributions of completion times cause traditional machine learning algorithms in software packages like Weka to fail in predicting the numerical value of completion times for crowdsourced tasks. This heavy tail distribution is detailed in [Barabasi, 2005] and also studied for AMT in [Wang et al., 2011b]. Cox proportional hazard regression models and survival curves are typically used to model these heavy tail behaviors. There is a nonlinear relationship between the value of completion time and the predictors that are used to train

the model. We show that this value is monotonically decreasing for increasing reward values. This property is used to design an algorithm for finding the reward for a candidate task. This reward ensures the minimum payment for the task with a desired completion time.

Using the empirical data from Mechanical Turk and examples from private crowds like CONE [Faridani et al., 2009] we show that arrivals follow a NHPP model. This enables us to simulate arrival of the workers to the market. We then use discrete choice models and a multinomial logit model in particular to show how a requester can optimize her task by increasing the likelihood of the task being picked by workers.

We are interested in exploring the discrete choice model further and extending it to a closed form formulation that combines both the arrival model and the logit model to estimate the completion times.

Algorithm 4: Algorithm for calculating reward for the desired completion time. **R** is the reward and **CT** is completion time

Input: Dataset A that contains historical information about different tasks, their posting date, reward, requester,...
Attributes of the new task h
Desired completion time t_{max}
Maximum payable reward R_{max}
Precision value ϵ
Output: Reward amount
begin
$R_{min} \leftarrow 0$
$R_{max} \leftarrow R_{max}$
$R_{mid} \leftarrow (R_{min} + R_{max})/2$
$CT_{R_{min}} \leftarrow SurvFit(A, h_{R_{min}})$
$CT_{R_{max}} \leftarrow SurvFit(A, h_{R_{max}})$
$CT_{R_{mid}} \leftarrow SurvFit(A, h_{R_{mid}})$
while $|CT_{R_{min}} - CT_{R_{max}}| > \epsilon$ **do**
if $CT_{R_{mid}} \geq t_{max}$ **then**
$R_{max} \leftarrow R_{mid}$
$R_{mid} \leftarrow (R_{min} + R_{max})/2$
if $CT_{R_{mid}} < t_{max}$ **then**
$R_{min} \leftarrow R_{mid}$
$R_{mid} \leftarrow (R_{min} + R_{max})/2$
$CT_{R_{min}} \leftarrow SurvFit(A, h_{R_{min}})$
$CT_{R_{max}} \leftarrow SurvFit(A, h_{R_{max}})$
$CT_{R_{mid}} \leftarrow SurvFit(A, h_{R_{mid}})$
return R_{mid}

/* **Function *SurvFit(A, h)*** */
Input: Dataset A and new task h as defined above; **Output:** Expected completion time
begin
Completion Time = Find the completion time for h by fitting CoxPH to A (i.e., by using survfit(coxph(Surv(A),h)) in R language)
return Completion Time

Chapter 5

Crowdsourcing human subject experiments

5.1 Abstract

Fitts' Law is one of the few mathematical models that is used in designing user interfaces, mobile devices and even video games. It characterizes the expected time to reach a target by a logarithmic two-parameter relationship between the distance to the target (A) and the perceived width of the target (D). The A/D ratio is known as the "index of difficulty". In this chapter we compare three variants of the Fitts' Law. The Square-Root model (SQR), The modified Fitts' Law model by MacKenzie *et al.* (LOG') [MacKenzie and Buxton, 1992], and the classical Logarithmic Fitts' Law (LOG). The derivation of the Square-Root model that is provided in this chapter is by Ken Goldberg and is provided here for reference. This derivation is more intuitive, exact and makes fewer assumptions than the derivation presented in Meyer *et al.* [Meyer et al., 1988]. We use a linear regression to fit the unknown parameters for each model and compute the resulting root-mean-squared error[1] and variance. We perform two-sided paired Student t-tests comparing the within-subject models using the $p = 0.05$ level of significance.

We present data from two experimental user studies, one a controlled (in-lab) study and the second an uncontrolled (online) study. The controlled study collected $16,170$ valid timing measurements from 46 volunteers using the identical mouse types and settings. The uncontrolled (online) study collected $78,410$ valid timing measurements from an indeterminate number of volunteers who visited the website with a variety of mouse types and settings. Both studies include two conditions, a "homogeneous targets" condition where sequential targets are constant in distance and size, and a "heterogeneous targets" condition where sequential targets vary in distance and size. To the best of our knowledge, the dataset of $94,580$ timing measurements is the largest dataset to date for human reaching

[1] In this chapter we use ERMS as the principal error measure ($ERMS = \sqrt{E((y_i - f_i)^2)}$). ERMS and R^2 have been both used in the literature. In this case R^2 is calculated as $R^2 = 1 - \frac{\sum_i (y_i - f_i)^2}{\sum_i (y_i - \overline{y})^2}$

Figure 5.1: Using an applet, sequences of rectangular and circular targets are presented to users, where target distance A and width W can remain constant (homogeneous) or vary (heterogeneous) after every click.

motion. The experimental applet and dataset are openly available to other researchers at http://www.tele-actor.net/fitts/.

In this chapter we show that: (1) the data from the controlled and uncontrolled studies are remarkably consistent; (2) for homogeneous targets, the SQR model yields a significantly better fit than LOG or LOG', except with the most difficult targets (with higher index of difficulty) where the models are not significantly different; (3) for heterogenous targets, SQR yields a significantly better fit than LOG for easier targets and LOG yields a significantly better fit for more difficult targets, while the the LOG' model yields a significantly better fit than both LOG and SQR on more difficult targets.

This chapter is an example of crowdsourcing a user experience study. The fact that the two studies were consistent demonstrates that, when designed well, crowdsourced user studies can be viable tools for validating hypothesis.

5.2 Introduction

Variety of tasks from playing video games to operating an airplane require timely human reaching movements between targets. In this work we focus on computer interfaces where a pointing device (e.g., mouse, touch, etc) is used to reach target areas displayed on the scene. To facilitate better user interface designs, engineers and scientists seek accurate human motor models. One of the well-known models that demonstrates the tradeoff between speed and accuracy of human motion is provided by Paul Fitts in 1954 [Fitts, 1954]. He studied times for reaching "homogeneous targets" where the size and the distance between the initial starting point and the target are fixed. He measured the time T required to repetitively move a stylus between two parallel plates with the set width W and the amplitude A for 16 human volunteers.

Based on Shannon's information theory [Shannon, 2001], Fitts derived the classic Logarithmic model for calculating the expected time to reach a target with the index of difficulty of A/W. This model is now known as the "Fitts' Law".

This chapter contributes new experiments based on the new theory that we are presenting in our working paper [Goldberg et al., 2012]. In the paper Goldberg, Faridani and Alterovitz reconsider the square-root model and provide a succinct derivation based on optimal control theory. The derivation is intuitive, exact, makes fewer assumptions, and requires fewer steps than the derivation presented in Meyer *et al.* [Meyer et al., 1988]. Meyer *et al.* used the original Fitts' data and performed new experiments with only four human subjects performing wrist rotation movements to heterogenous targets.

We undertook two comprehensive user studies to gather data on humans performing computer screen cursor motions. We designed and implemented a java-based applet that can be easily downloaded from the web. It presents users with a sequence of visual targets to click on, records completion times, and sends the data back to our lab. Our first study is a standard controlled (in-lab) user study with volunteers using the applet with identical mouse types and settings. Our second study is an uncontrolled (web-based, "in the wild") study based on an indeterminate number of volunteers who visited the website (many visited more than once) and used a variety of mouse types and settings.

Uncontrolled (also known as "in the wild") studies on the web do not provide the consistency of controlled in-lab studies but can collect data from large numbers of diverse human participants and are gaining acceptance, especially when confirmed by controlled experiments [Bigham et al., 2010b, Bakshy et al., 2012].

Uncontrolled studies gather data in a variety of settings with perhaps greater ecological validity conditions than found in a laboratory. However, there are substantial methodological disadvantages with uncontrolled (web-based) studies. One cannot obtain detailed data about the users, some users may perform the experiment multiple times, and one has no control over the user environment nor the input and display devices.

Kittur et al [Kittur et al., 2008] consider how a "crowdsourcing" system such as Mechanical Turk can be used for user studies and find that the diversity and unknown nature of the user base can be "both a benefit and a drawback." They suggest that careful design of the tests to avoid gaming can yield results that are comparable with controlled studies.

In a survey, Andreasen et al [Andreasen et al., 2007] systematically compare controlled and uncontrolled (web-based) usability studies and find that synchronous studies (with a live remote human monitor) are more reliable than asynchronous studies (akin to our uncontrolled experiments) but that both enable collection of user data from a large number of participants. They note that *"it would be interesting to perform comparative studies of remote usability testing methods"* against controlled studies.

Uncontrolled experiments are gaining acceptance in the Computer Human Interaction community. Our uncontrolled study was motivated by our desire to learn how the results might vary between a controlled lab setting and online with many different experimental environments. We were surprised to find that data from the controlled and uncontrolled studies were remarkably consistent.

5.3 Related Work

Since the goal of this chapter is to compare the controlled and uncontrolled experiments we only provide a brief summary of the models and their history. For a more detailed overview of related work we ask interested readers to see our working paper [Goldberg et al., 2012] or read the author's Master's thesis[2].

Classic Fitts' Law (LOG)

In 1954, Fitts published his now-classic paper in which he hypothesized that the information capacity of the human motor system is specified by its ability to produce consistently one class of movement from among several alternative classes of movements [Fitts, 1954]. Using Shannon's definition of information as a guideline, Fitts provided Eq. 5.1 for movement time T as a two-parameter Logarithmic function of the index of difficulty:

$$T = a + b \log_2 \left(\frac{2A}{W} \right). \tag{5.1}$$

In this chapter we refer to this as the LOG model.

The MacKenzie Model (LOG')

Scott MacKenzie developed a variation on Fitts' model that accurately predicts data of the original Fitts' experiment [MacKenzie, 1992]. In MacKenzie's model the movement time is given by:

$$T = a + b \log_2 \left(\frac{A}{W} + 1 \right). \tag{5.2}$$

Applications of Fitts' Law

Plamondon and Alimi review a number of studies of speed/accuracy trade-off models and their applications [Plamondon and Alimi, 1997]. They categorize the experimental procedures used for the speed/accuracy trade-offs into two different categories: spatially constrained movements and temporally constrained movements. For the procedures in the first category, distance (A) and the width (W) are usually given and the time (T) is measured. In the temporal group, movement time is given and the accuracy of reaching the target is being measured. With this definition, Fitts' Law falls into the first category. They classify different studies on the Fitts' Logarithmic model based on different types of movements (tapping, pointing, dragging), limbs and muscles groups (foot, head, hand, etc), experimental conditions (underwater, in flight, etc), device (joystick, mouse, stylus, touchpad, etc), and participants (children, monkeys, adults of different ages, etc).

[2] Available at http://www.eecs.berkeley.edu/Pubs/Theses/Years/2012.html

Hoffmann and Hui study reaching movements of fingers, wrist, forearm and shoulder. They show for the cases where an operator can choose which limb to use to reach a target, the limb with the smallest mass moment of inertia is often used to minimize energy needed to reach the target [Hoffmann and Hui, 2010].

Alternative Models of Reaching Movements

Alternatively Plamondon modeled movement time as a power model with two parameters

$$T = K\left(\frac{2A}{W}\right)^{\alpha} \tag{5.3}$$

with parameters K and α.

Equation 5.3 defines a power model, an alternative two-parameter formulation based on a fitted log-normal approximation of the velocity profile. We welcome fellow researchers to apply such models to the dataset we provide.

The Square-Root Model (SQR)

Several researchers have considered a two-parameter square-root model:

$$T = a + b\sqrt{\frac{A}{W}}. \tag{5.4}$$

In this chapter we refer to this as the SQR model. Kvalseth and Meyer *et al.* noted that the SQR model behaves similarly to the logarithmic model in the standard range of index of difficulty [Kvålseth, 1980, Meyer et al., 1988].

Meyer *et al.* used the homogeneous target data from the original Fitts' paper [Fitts, 1954], and showed that the SQR model fits the original data better than the LOG model [Meyer et al., 1988]. Meyer *et al.* also performed experiments with 4 human subjects performing wrist rotation movements to heterogenous targets with similar results

Meyer *et al.* propose a complex derivation of the SQR model based on the assumption that reaching motion can be partitioned into two submovements, a primary ballistic submovement and a secondary corrective submovement, with near-zero velocity at the transition. The derivation is an approximation based on four strong assumptions: 1) there are two submovements with a stop between them, 2) submovement endpoints have Gaussian distributions around the center point of the target, and 3) the standard deviation of each Gaussian is linearly related to the average velocity during that submovement, and 4) there are strong numerical bounds on values of A and W for which the approximation holds.

Meyer *et al.* then derive the time T to reach the target as the sum of the average times for the primary submovement T_1 and for the corrective submovement T_2. They estimate T by minimizing its derivative with respect to the submovements and show that when $A/W > 4/z\sqrt{2\pi}$ the value of T can be approximated by the SQR function above where z is the

z-score such that 95% of the area under a standard Gaussian distribution $N(0,1)$ falls inside $(-z, z)$.

In addition to its complexity vis a vis Occam's Razor, there are several other drawbacks to this derivation [Rioulo and Guiard, 2012]. As Meyer et al. note, if the participant reaches the target in a single movement, the derivation collapses to a linear model which fits the data very poorly. The approximation requires numerical bounds on values of A and W. Furthermore, Guiard et al. note that for a fixed positive value of A/W Meyer's model approaches 1 as the number of submovements n approaches infinity [Guiard et al., 2001, Rioulo and Guiard, 2012]. Meyer et al. evaluated their model with one-dimensional movements using wrist rotation of a dial that can be rotated to different angular targets. In their experiments, 4 participants are presented with 12 target conditions with A/W values ranging from 2.49 to 15.57. This range of A/W does not violate the assumption made for their derivation.

There is a detailed derivation of the Meyer's model in author's thesis that is available for download on the Berkeley website [3]

5.4 A Succinct Derivation of the Square-Root (SQR) Model

It is well known in control theory that the optimal time for a system to reach a target is obtained by "bang-bang" control, where maximal positive acceleration is maintained for the first half of the trajectory and then switched to maximal negative acceleration for the second half [Macki and Strauss, 1982, Jagacinski and Flach, 2003].

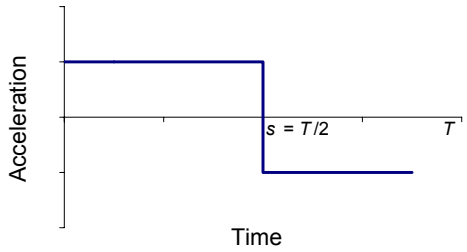
In this section we provide a new derivation for the SQR model that models acceleration as (1) piecewise constant as predicted by optimal control theory, and (2) proportional to target width: wider targets are perceived by humans as "easier" to reach and hence humans apply larger accelerations as they have a larger margin for error.

Given this model, we define the halfway point (the point reached at the switching time) for a human to reach a target at distance A as $x_{mid} = A/2$. Acceleration as a function of time for bang-bang control is shown in Figure 5.2(a), where the switching time between maximum acceleration and maximum deceleration is $s = T/2$.

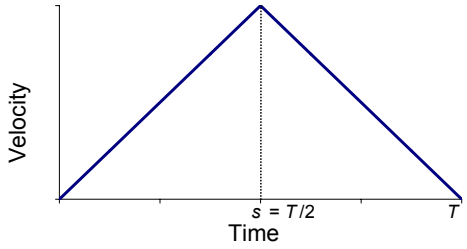
As shown in Figure 5.2, Acceleration has only two values: full forward or full reverse, hence the term "bang-bang". Velocity is initially zero and then ramps up linearly during the first phase and ramps down during the second. Velocity is thus $\dot{x}(t) = \ddot{x}t$ during the acceleration phase ($t \leq s$) and $\dot{x}(t) = \ddot{x}s - \ddot{x}(t-s)$ during the deceleration phase ($t > s$), where $\ddot{x}$ is the constant magnitude of acceleration.

We can integrate this linear velocity with respect to time to get a quadratic function for position $x(t)$. At the switching time s, the position by integration will be $x(s) = \frac{1}{2}\ddot{x}s^2$. By symmetry, position after time $T = 2s$ will be $x(T) = \ddot{x}s^2 = \frac{1}{4}\ddot{x}T^2$. For cursor motion, we

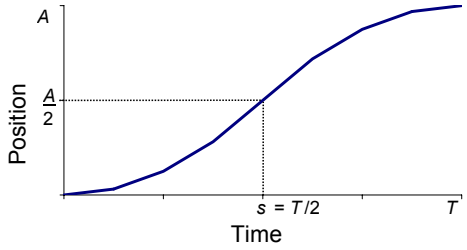
[3] http://www.eecs.berkeley.edu/Pubs/Theses/Years/2012.html



(a)



(b)



(c)

Figure 5.2: Acceleration vs. Time (a), Velocity vs. Time (b), and Position vs. Time (c) under symmetric optimal control. The "bang-bang" controller maintains the maximal positive acceleration in the first half of the motion and then switches to the maximal negative acceleration until the target is reached (a). The maximal velocity is reached in the middle of the path (b).

set the total distance traveled during movement time T as the amplitude $x(T) = A$. Hence, $A = \frac{1}{4}\ddot{x}T^2$ which implies

$$T = 2\sqrt{\frac{A}{\ddot{x}}}. \tag{5.5}$$

Now, from the second assumption, acceleration magnitude is proportional to the width of the target: $\ddot{x} = kW$ where k is a constant scalar and W is the target width. Substituting into equation 5.5, we get

$$T = 2\sqrt{\frac{A}{kW}}.$$

We now add an initial reaction time a and let $b = 2/\sqrt{k}$. The total movement time is then:

$$T = a + b\sqrt{\frac{A}{W}}. \tag{5.6}$$

This derivation is intuitive, exact, makes fewer assumptions, and requires fewer steps than the two-submovement derivation presented in Meyer *et al.* [Meyer et al., 1988].

For the derivation of this model for asymmetric acceleration please see our paper [Goldberg et al., 2012]

5.5 Experimental User Studies

We performed two experimental user studies, one a controlled (in-lab) study and the second an uncontrolled (web-based) study. Both studies include two conditions, a "homogeneous targets" condition where sequential targets are constant in distance and size, and a "heterogeneous targets" condition where sequential targets vary in distance and size. The experimental test and full dataset are available online at http://www.tele-actor.net/fitts/.

Experiment Conditions: The Java Applet

For both the controlled and uncontrolled studies, we implemented a Java applet that asks each subject to complete two experiments by using his or her cursor to click on a sequence of rectangular or circular targets as they are presented on the screen. The Java applet is available online at http://www.tele-actor.net/fitts/.

The applet records the time in milliseconds between when the target appears until the subject clicks on the target. A subject may click when the cursor is outside the target, but the timer increments until the target is successfully clicked upon. To allow precise measurement of movement times without lag from Internet communications, movement times are measured locally by the applet and sent to our central server after completion of the trials. We did not attempt to capture the complete motion trajectory as we were not confident that clients would have sufficient processing speed when running other processes to take reliable measurements.

Homogeneous Targets Experiment

This set of trials focuses on repetitive motions like the ones studied in the original Fitts papers. A sequence of 33 vertical rectangles are presented as illustrated in Figure 5.1(a). The first, second, and third set of the 11 rectangles have the same (homogenous) width

and amplitude and hence the same index of difficulty. In other words after the 11th, 22nd, and 33rd repetition, the width and amplitude (and index of difficulty) of the rectangles are changed. To allow subjects to "warm-up" and become familiar with each set, the system discards timing data from the first 3 timing measurements out each set of 11, so data from the latter 8 rectangles for each index of difficult is collected, producing 24 timing measurement.

Heterogeneous Targets Experiment

This set of trials focuses on changing targets as might be encountered in a game or computer human interface. A sequence of 25 circular targets are presented as illustrated in Figure 5.1(b). Each trial begins when the subject clicks inside a small "home" circle in the center of the window and ends when the user successfully clicks inside the target. Each of the circular targets varies in distance from the home circle and varies in diameter (and hence in index of difficulty).

The distance/amplitude and size/width of the targets (in pixels) are shown in Table 5.1. Note that the index of difficulty varies and is not strictly increasing or decreasing. Since the targets are measured in units of pixels, the distance and size of targets may appear different on computer systems with different display sizes and resolutions.

Two User Studies

User studies were conducted under UC Berkeley human subject certificate $IRB - 2009 - 09 - 283$.

Controlled User Study

For the controlled user study, we posted ads on campus and Facebook offering an Amazon.com gift certificate for participation. Forty-six (46) people responded, including 17 female (37%) and 29 male (63%) participants. From a questionnaire, we learned that the distribution of their ages is as shown in Figure 5.3. The average age was 24.7 (variance = 23.8). We also learned that participants play video games for an average of 1.5 hours per week (the population has a high variance of 10.01 hours, suggesting that the majority do not play video games during the week ;). Out of the 46 subjects, 4 were left-handed, but opted to use their right hand to operate the pointing device. Although all of the left-handed participants were given the chance to customize their environment, none of them changed their mouse settings to left-handed; prior studies have shown that this does not disadvantage left-handed users [Hoffmann et al., 1997].

Each subject performed the homogenous target and the heterogeneous target experiments 10 times. Subjects were given breaks between experiments to reduce fatigue. The experiments were performed under supervision of lab assistants who encouraged subjects to repeat a trial if the subject became distracted.

Trial	Homogeneous Targets		Heterogeneous Targets	
	A	W	A	W
1	370	50	67	20
2	370	50	184	38
3	370	50	280	14
4	370	50	230	29
5	370	50	144	55
6	370	50	249	29
7	370	50	255	14
8	370	50	96	50
9	240	10	225	19
10	240	10	263	12
11	240	10	259	25
12	240	10	229	20
13	240	10	215	31
14	240	10	198	83
15	240	10	301	16
16	240	10	194	66
17	180	70	260	12
18	180	70	296	14
19	180	70	180	44
20	180	70	278	11
21	180	70	283	37
22	180	70	40	32
23	180	70	233	10
24	180	70	191	50
25	-	-	179	18

Table 5.1: Target distance/amplitude (A) and size/width (W), in display pixels, for the 24 recorded Fixed Rectangles (Fixed Rectangles) trials and 25 Variable Circles trials.

For this controlled experiment, we collected a total of $22,540$ timing measurements ($11,040$ for homogenous targets and $11,500$ for heterogenous targets). We cleaned this raw dataset by keeping only timing measurements for cases where the subject successfully clicks on all presented targets within a "reasonable" time period (within 3 std dev of the global mean time). Our goal is to remove most cases where subjects were distracted or decided not to complete the experiment. After cleaning, the dataset contains $16,170$ valid timing measurements ($8,250$ for homogenous targets and $7,920$ for heterogenous targets).

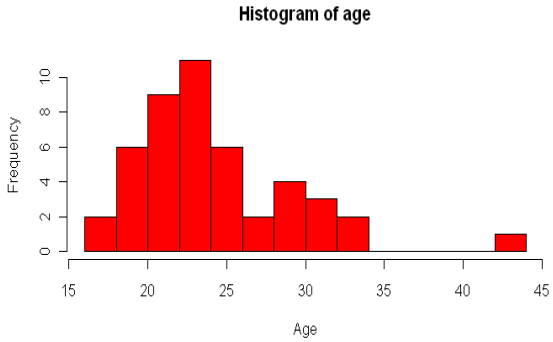


Figure 5.3: Age distribution for participants for the controlled study

Uncontrolled User Study

To conduct the uncontrolled study, we made the same applet available online and advertised by emails and postings on user groups. The experimental applet and datasets are available online at http://www.tele-actor.net/fitts/.

To comply with our Human Subjects approval, each online participant is asked to click an online consent box before starting the applet. An entry is created in the server database each time the consent box is clicked. We do not record IP addresses and cannot determine if the same person runs the experiment multiple times so we do not know the number of unique participants. We ask online visitors to indicate the type of mouse device they use (trackpad, mouse, trackball, etc), but cannot verify these responses.

The online applet presents visitors with 24 homogenous targets and 25 heterogenous targets and thus collects up to 49 timing measurements. Unlike the controlled experiment, online visitors were not asked to repeat each experiment 10 times. (The online applet includes a third experiment with variable-sized rectangular targets; we discovered a timing error in that experiment so we do not use data from that experiment.)

We cannot determine how many visits are repeats (by the same person). We collected timing data from $2,689$ visits to the homogeneous target experiment and $2,811$ visits to the heterogenous target experiment. As we did in the controlled study, we cleaned the raw dataset by keeping only timing measurements for cases where the subject successfully clicks on all presented targets within a "reasonable" time period (within 3 std dev of the global mean time). Our goal is to remove most cases where subjects were distracted or decided not to complete the experiment.

After cleaning, the online study dataset includes $78,410$ valid timing measurements ($39,360$ for the homogeneous targets and $39,050$ for the heterogenous targets).

	LOG Model		SQR Model		Hypothesis Testing	
A/W	μ_{ERMS}	σ_{EMRS}	μ_{ERMS}	σ_{EMRS}	p-value	Best Fit
2.57	224.16	147.90	120.22	64.04	7.68E-28	**SQR**
7.40	421.80	291.36	237.92	132.98	5.26E-23	**SQR**
24.00	704.86	489.78	553.09	329.48	3.74E-06	**SQR**

Table 5.2: Homogeneous Targets: Controlled Study: Prediction Error and Pairwise Fit between LOG and SQR models. SQR yields a significantly better fit than LOG.

	LOG' Model		SQR Model		Hypothesis Testing	
A/W	μ_{ERMS}	σ_{EMRS}	μ_{ERMS}	σ_{EMRS}	p-value	Best Fit
2.57	147.36	87.26	120.22	64.04	6.32E-06	**SQR**
7.40	299.46	191.16	237.92	132.98	2.00E-06	**SQR**
24.00	549.20	358.28	553.09	329.48	8.84E-01	LOG'

Table 5.3: Homogeneous Targets: Controlled Study: Prediction Error and Pairwise Fit between LOG' and SQR models. SQR yields a significantly better fit than LOG' except for the most difficult targets, where the two models are not significantly different.

Experimental Results

Using the data we collected, we compare three two-parameter models that relate motion duration to the index of difficulty: LOG (the classic logarithmic function), SQR (square-root), and LOG' (logarithmic plus 1.0 proposed by [MacKenzie and Buxton, 1992]).

We use regression to fit the unknown a, b parameters for each subject and model and compute the resulting root-mean-squared (RMS) error and variance. We perform two-sided paired Student t-tests comparing the within-subject models using the $p = 0.05$ level of significance. As noted by R. A. Fisher in his classic text, Statistical Methods for Research Workers: "The value for which $p = 0.05$, or 1 in 20, is 1.96 or nearly 2; it is convenient to take this point as a limit in judging whether a deviation ought to be considered significant or not. Deviations exceeding twice the standard deviation are thus formally regarded as significant. Using this criterion we should be led to follow up a false indication only once in 22 trials, even if the statistics were the only guide available. Small effects will still escape notice if the data are insufficiently numerous to bring them out, but no lowering of the standard of significance would meet this difficulty."

Homogeneous Targets

Data from the controlled study are presented in Tables 5.2 and 5.3 Data from the uncontrolled study are presented in Tables 5.4 and 5.5. The last column indicates the model that fits better and is in bold face if the difference is statistically significant beyond the $p < .05$ level.

	LOG Model		SQR Model		Hypothesis Testing	
A/W	μ_{ERMS}	σ_{EMRS}	μ_{ERMS}	σ_{EMRS}	p-value	Best Fit
2.57	257.63	166.89	143.87	82.75	1.80E-120	**SQR**
7.40	484.39	322.55	296.63	177.29	8.84E-88	**SQR**
24.00	814.39	545.63	686.34	423.14	7.68E-14	**SQR**

Table 5.4: Homogeneous Targets: Uncontrolled Study: Prediction Error and Pairwise Fit between LOG and SQR models. As in the Controlled study, SQR yields a significantly better fit than LOG.

	LOG' Model		SQR Model		Hypothesis Testing	
A/W	μ_{ERMS}	σ_{EMRS}	μ_{ERMS}	σ_{EMRS}	p-value	Best Fit
2.57	173.60	102.60	143.87	82.75	1.15E-19	**SQR**
7.40	351.91	218.18	296.63	177.29	2.33E-15	**SQR**
24.00	649.56	412.45	686.34	423.14	1.18E-02	**LOG'**

Table 5.5: Homogeneous Targets: Uncontrolled Study: Prediction Error and Pairwise Fit between LOG' and SQR models. As in the Controlled study, SQR yields a significantly better fit than LOG' except for the most difficult targets, where the two models are not significantly different.

The results from the controlled and uncontrolled experiments are remarkably consistent. For homogeneous targets, the SQR model yields significantly better fit than LOG or LOG', except for the most difficult targets where the models are not significantly different.

Heterogeneous Targets Experiments

Data from the studies are presented first using four sets of plots and then in four numerical tables (Tables VI through IX). The plots show RMS Error and std. deviation for increasing values of index of difficulty for pairs (two models) at a time. The first two plots compare the LOG and SQR in the Controlled and Uncontrolled Experiments respectively. The third and fourth plots compare the LOG' and SQR in the Controlled and Uncontrolled Experiments respectively. In the tables, the last column indicates the model that fits better and is in bold face if the difference is statistically significant beyond the $p < .05$ level.

In both controlled and uncontrolled studies with heterogeneous targets, SQR yields a significantly better fit than LOG for easier targets and LOG yields a significantly better fit for harder targets. For heterogenous targets, the LOG' model yields significantly better fit than LOG or SQR, except for easier targets where the models are not significantly different.

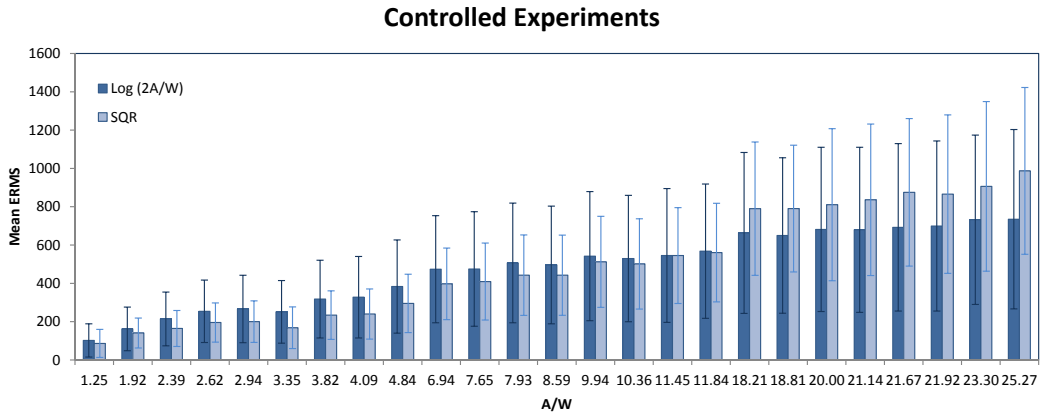


Figure 5.4: Heterogeneous Targets: Controlled user Study: LOG vs SQR models. See Tables VI through IX for numerical details.

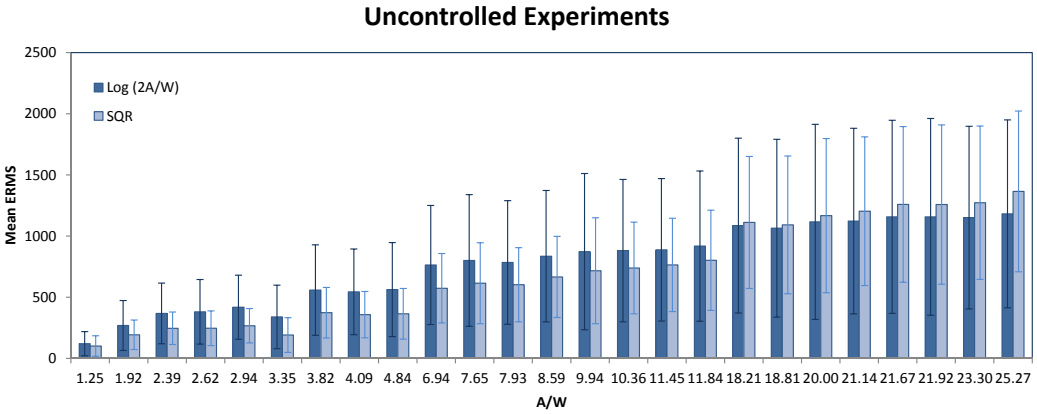


Figure 5.5: Heterogeneous Targets: Uncontrolled user Study: LOG vs SQR models.

Discussion and Conclusion

We explore three two-parameter models that relate human motion duration to the "index of difficulty" of the targets: LOG (the classic logarithmic function), SQR (square-root), and LOG' (logarithmic plus 1.0 proposed by [MacKenzie and Buxton, 1992]). The latter two have been proposed as superior models.

We describe new experiments. We present data from two experimental user studies, one a controlled (in-lab) study and the second an uncontrolled (online) study. The controlled study collected 16, 170 valid timing measurements from 46 volunteers using the identical mouse and settings. The uncontrolled (online) study collected 78, 410 valid timing measurements from an indeterminate number of volunteers who visited the website with with a variety of mouse types and settings. Both studies include two conditions, a "homogeneous targets" condition

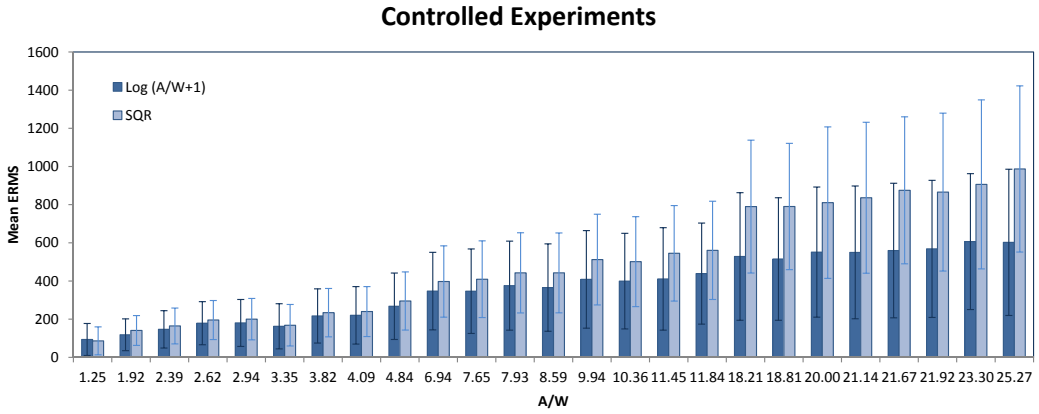


Figure 5.6: Heterogeneous Targets: Controlled user Study: LOG' vs SQR models.

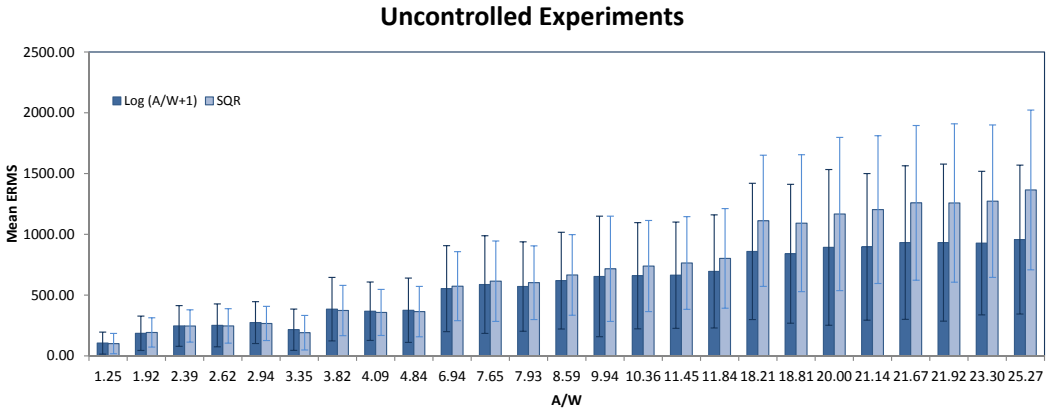


Figure 5.7: Heterogeneous Targets: Uncontrolled user Study: LOG' vs SQR models.

where sequential targets are constant in distance and size, and a "heterogeneous targets" condition where sequential targets vary in distance and size.

We use regression to fit the unknown parameters for each model and compute the resulting root-mean-squared error and variance. We perform two-sided paired Student t-tests comparing the within-subject models using the $p = 0.05$ level of significance.

We find that (1) the data from the controlled and uncontrolled studies are remarkably consistent. Tables VIII and IX exhibit some inconsistency for easier targets but these values are not statistically significant. Although uncontrolled experiments do not provide the consistency of controlled in-lab studies, they are gaining popularity as they can collect data from large numbers of diverse human participants. A few earlier studies have also shown consistent results from controlled and uncontrolled experiments [Bigham et al., 2010b, Bakshy et al., 2012].

We find that (2) for homogeneous targets, the SQR model yields a significantly better fit

	LOG Model		SQR Model		Hypothesis Testing	
A/W	μ_{ERMS}	σ_{EMRS}	μ_{ERMS}	σ_{EMRS}	p-value	Best Fit
1.25	102.09	86.82	86.26	73.34	1.02E-02	**SQR**
1.92	162.18	113.97	140.68	78.22	4.42E-03	**SQR**
2.39	214.24	140.20	164.37	93.82	1.38E-07	**SQR**
2.62	253.96	162.95	195.47	102.42	6.79E-08	**SQR**
2.94	266.23	176.04	199.92	108.42	1.28E-08	**SQR**
3.35	250.99	163.39	168.21	108.85	1.00E-13	**SQR**
3.82	317.67	203.14	234.14	126.64	4.47E-10	**SQR**
4.09	327.71	212.77	239.75	130.97	4.62E-10	**SQR**
4.84	383.26	243.54	295.41	152.44	4.26E-08	**SQR**
6.94	473.81	279.80	397.26	186.73	4.90E-05	**SQR**
7.65	474.89	299.14	409.27	201.03	1.09E-03	**SQR**
7.93	506.52	312.39	442.76	209.92	2.41E-03	**SQR**
8.59	495.86	307.24	442.47	209.20	8.85E-03	**SQR**
9.94	541.84	337.11	512.23	237.91	2.04E-01	SQR
10.36	529.48	329.97	501.45	235.70	2.07E-01	SQR
11.45	545.50	349.61	545.01	250.04	9.92E-01	SQR
11.84	567.83	350.57	560.45	257.54	7.35E-01	SQR
18.21	663.13	419.94	789.96	348.02	2.55E-05	**LOG**
18.81	649.96	405.78	790.24	330.94	1.86E-06	**LOG**
20.00	681.42	428.81	810.65	397.08	6.07E-05	**LOG**
21.14	679.44	431.03	836.16	395.38	1.41E-06	**LOG**
21.67	692.45	437.13	875.12	385.34	1.74E-08	**LOG**
21.92	699.30	444.25	865.82	413.59	1.08E-06	**LOG**
23.30	732.01	442.11	906.26	442.91	7.10E-07	**LOG**
25.27	734.77	468.18	987.07	435.41	1.83E-12	**LOG**

Table 5.6: Heterogeneous Targets: Controlled user study: LOG vs SQR models. SQR yields a significantly better fit than LOG for easier targets and LOG yields a significantly better fit for harder targets.

than LOG or LOG', except with the most difficult targets (with higher index of difficulty) where the models are not significantly different. That SQR is superior is surprising in these cases since Fitts' original experiments were with homogenous targets but is consistent with more recent experiments.

We find that (3) for heterogenous targets, SQR yields a significantly better fit than LOG for easier targets and LOG yields a significantly better fit for more difficult targets. The results are inconclusive for targets in the middle range of index of difficulty, while the the LOG' model yields a significantly better fit than both LOG and SQR on more difficult

	LOG Model		SQR Model		Hypothesis Testing	
A/W	μ_{ERMS}	σ_{EMRS}	μ_{ERMS}	σ_{EMRS}	p-value	Best Fit
1.25	119.57	98.92	100.97	83.80	1.56E-08	**SQR**
1.92	268.83	204.39	192.50	120.64	6.22E-36	**SQR**
2.39	367.02	248.20	245.90	133.16	3.02E-61	**SQR**
2.62	380.05	264.30	246.15	141.72	1.32E-65	**SQR**
2.94	417.87	261.97	266.70	140.79	4.45E-83	**SQR**
3.35	338.92	259.77	190.52	141.98	4.51E-81	**SQR**
3.82	558.07	369.86	373.06	206.70	4.52E-63	**SQR**
4.09	543.80	350.37	357.48	189.44	1.87E-71	**SQR**
4.84	562.20	384.60	364.24	207.13	2.01E-67	**SQR**
6.94	763.44	486.99	573.27	283.94	3.13E-39	**SQR**
7.65	800.26	538.79	614.45	330.99	1.97E-30	**SQR**
7.93	784.19	505.72	602.04	303.43	2.40E-33	**SQR**
8.59	835.37	537.14	665.74	331.57	8.00E-26	**SQR**
9.94	872.83	638.81	716.32	433.10	1.62E-15	**SQR**
10.36	881.19	582.18	738.77	374.93	6.56E-16	**SQR**
11.45	887.13	582.48	764.25	381.57	3.85E-12	**SQR**
11.84	917.90	614.22	801.91	409.63	6.14E-10	**SQR**
18.21	1085.70	715.08	1111.37	539.99	2.58E-01	LOG
18.81	1064.30	727.25	1091.28	563.31	2.46E-01	LOG
20.00	1116.00	797.88	1167.20	630.27	4.66E-02	**LOG**
21.14	1122.52	758.94	1203.31	607.69	1.03E-03	**LOG**
21.67	1157.07	790.10	1258.60	636.56	7.83E-05	**LOG**
21.92	1156.89	804.50	1258.05	651.53	1.15E-04	**LOG**
23.30	1151.02	746.97	1272.71	627.45	8.67E-07	**LOG**
25.27	1181.75	768.04	1364.91	657.32	1.00E-12	**LOG**

Table 5.7: Heterogeneous Targets: Uncontrolled user study: LOG vs SQR models. As in the Controlled study, SQR yields a significantly better fit than LOG for easier targets and LOG yields a significantly better fit for harder targets.

targets. This suggests that there might be an underlying difference in human motor processes for targets of different levels of difficulty and more work remains to be done.

Our applet records the time in milliseconds between when the target appears until the subject clicks on the target. We did not attempt to capture the complete motion trajectory as we were not confident that clients would have sufficient processing speed when running other processes to take reliable measurements, but this is an interesting avenue for future study.

To the best of our knowledge, the dataset of $94,580$ timing measurements is the largest

	LOG' Model		SQR Model		Hypothesis Testing	
A/W	μ_{ERMS}	σ_{EMRS}	μ_{ERMS}	σ_{EMRS}	p-value	Best Fit
1.25	93.39	83.74	86.26	73.34	2.33E-01	SQR
1.92	118.17	83.18	140.68	78.22	4.36E-04	**LOG'**
2.39	146.71	97.94	164.37	93.82	1.76E-02	**LOG'**
2.62	178.96	112.99	195.47	102.42	4.69E-02	**LOG'**
2.94	180.12	123.00	199.92	108.42	2.71E-02	**LOG'**
3.35	162.56	118.13	168.21	108.85	5.20E-01	LOG'
3.82	216.83	142.10	234.14	126.64	1.06E-01	LOG'
4.09	220.04	150.55	239.75	130.97	6.95E-02	LOG'
4.84	267.82	174.09	295.41	152.44	3.36E-02	**LOG'**
6.94	347.05	203.18	397.26	186.73	9.34E-04	**LOG'**
7.65	346.52	221.30	409.27	201.03	1.54E-04	**LOG'**
7.93	375.59	233.22	442.76	209.92	1.11E-04	**LOG'**
8.59	365.48	228.88	442.47	209.20	9.72E-06	**LOG'**
9.94	408.58	255.77	512.23	237.91	9.08E-08	**LOG'**
10.36	399.28	250.50	501.45	235.70	1.18E-07	**LOG'**
11.45	410.84	268.34	545.01	250.04	5.82E-11	**LOG'**
11.84	438.75	264.69	560.45	257.54	5.19E-09	**LOG'**
18.21	528.38	334.25	789.96	348.02	2.00E-21	**LOG'**
18.81	515.00	321.04	790.24	330.94	6.00E-25	**LOG'**
20.00	551.39	341.18	810.65	397.08	2.53E-18	**LOG'**
21.14	549.85	347.95	836.16	395.38	1.83E-21	**LOG'**
21.67	559.49	352.79	875.12	385.34	8.17E-26	**LOG'**
21.92	568.38	359.30	865.82	413.59	3.90E-21	**LOG'**
23.30	606.58	356.09	906.26	442.91	3.89E-20	**LOG'**
25.27	602.49	383.22	987.07	435.41	2.59E-30	**LOG'**

Table 5.8: Heterogeneous Targets: Controlled user study: LOG' vs SQR models. The LOG' model yields a significantly better fit than SQR on harder targets (with higher index of difficulty).

dataset to date for human reaching motion. The experimental applet and dataset are openly available online at http://www.tele-actor.net/fitts/. The data may also contain patterns such as variations between subjects with overall faster response times and those that have slower response times. We encourage others to use this data with other metrics or to evaluate models beyond the three we study in this chapter.

	LOG' Model		SQR Model		Hypothesis Testing	
A/W	μ_{ERMS}	σ_{EMRS}	μ_{ERMS}	σ_{EMRS}	p-value	Best Fit
1.25	104.93	90.87	100.97	83.80	2.06E-01	SQR
1.92	185.80	141.69	192.50	120.64	1.55E-01	LOG'
2.39	245.99	167.69	245.90	133.16	9.87E-01	SQR
2.62	250.80	176.23	246.15	141.72	4.17E-01	SQR
2.94	273.89	172.17	266.70	140.79	2.02E-01	SQR
3.35	215.11	169.70	190.52	141.98	1.16E-05	**SQR**
3.82	384.23	260.92	373.06	206.70	1.85E-01	SQR
4.09	367.28	240.42	357.48	189.44	2.06E-01	SQR
4.84	375.31	264.38	364.24	207.13	1.93E-01	SQR
6.94	552.96	353.79	573.27	283.94	7.69E-02	LOG'
7.65	586.59	401.56	614.45	330.99	3.45E-02	**LOG'**
7.93	570.40	367.80	602.04	303.43	8.76E-03	**LOG'**
8.59	618.94	397.74	665.74	331.57	3.60E-04	**LOG'**
9.94	653.28	495.88	716.32	433.10	1.57E-04	**LOG'**
10.36	659.37	436.75	738.77	374.93	5.39E-08	**LOG'**
11.45	663.57	437.44	764.25	381.57	8.62E-12	**LOG'**
11.84	695.07	465.35	801.91	409.63	1.16E-11	**LOG'**
18.21	859.01	560.57	1111.37	539.99	1.13E-36	**LOG'**
18.81	839.83	571.25	1091.28	563.31	1.95E-34	**LOG'**
20.00	892.45	640.79	1167.20	630.27	7.12E-33	**LOG'**
21.14	896.95	603.33	1203.31	607.69	4.82E-44	**LOG'**
21.67	932.24	631.92	1258.60	636.56	1.89E-45	**LOG'**
21.92	931.79	646.30	1258.05	651.53	1.58E-43	**LOG'**
23.30	927.85	590.91	1272.71	627.45	3.09E-54	**LOG'**
25.27	956.81	612.56	1364.91	657.32	1.24E-68	**LOG'**

Table 5.9: Heterogeneous Targets: Uncontrolled user study: LOG' vs SQR models. As in the Controlled study, the LOG' model yields a significantly better fit than SQR on harder targets.

Chapter 6

Opinion Space: Effectiveness of dimensionality reduction for crowdsourced idea generation

Websites often collect textual comments from users. They often show and share these comments with other users of the website by using a linear list. Youtube[1], twitter [2] and Wikipedia [3] are among many websites that show users' textual comments in linear lists. Linear lists are one of the most common interfaces for collecting and showing textual opinions. One of the limitations of this interface is that it is not scalable and favors some comments over others based on the sorting model that is used (Figures 6.1, 6.2, 6.3). Limitations of linear lists in public discourse systems are well studied in the literature [Rourke and Kanuka, 2007, Thomas, 2002].

In this section we are examining crowdsourcing of the idea generation. To level the playing field and giving each idea an equal chance of being seen we examine dimensionality reduction and visualization as a viable options to make idea generation scalable. In this section we discuss Opinion Space, our project for crowdsourcing ideation. We also evaluate Principal Component Analysis as the algorithm to produce the visualization. In later chapters we extend this model to Canonical Correlation Analysis and highlight the pros and cons of that model over the current algorithm.

Opinion Space is a self-organizing system to collect, visualize and present textual ideas from the crowd around various topics. We have compared the new interface in an in-lab controlled experiment with two control interfaces – the traditional linear list and a grid interface that uses elements from Opinion Space but does not use the dimensionality reduction method. We have shown that the new interface is more engaging and participants were agreeing more with the comments that were visited through this interface and also found more insightful comments through the Opinion Space interface.

[1] http://youtube.com
[2] https://twitter.com/
[3] http://en.wikipedia.org

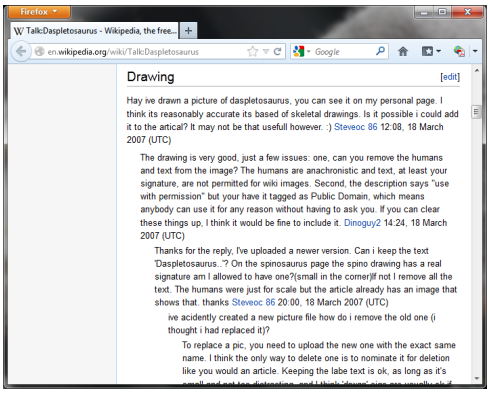


Figure 6.1: The "Talk" page for a wikipedia article about Daspletosaurus (meaning "frightful lizard". Comments are presented in a linear list sorted by topics.

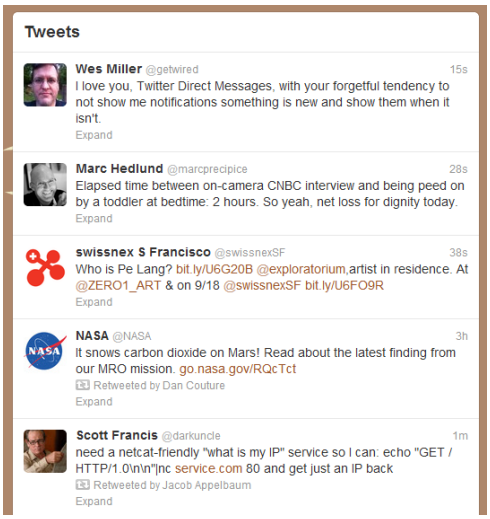


Figure 6.2: Tweets in twitter homepage of a user are sorted based on recency.

An earlier version of this chapter was originally published in [Faridani et al., 2010]. The author was repressible for building the control interfaces in Adobe Flex, running the human subject experiments and performing hypothesis testing on the data.

6.1 Introduction

Providing feedback, partaking in the policy making process, and evaluating other ideas is an essential component in participatory democracy. In a participatory culture in general people are free to provide content and evaluate the content that is provided by others. In many cases these contents are in the form of text. For example blog posts, news items, tweets are

Figure 6.3: Many of the comments on Mark Cuban's facebook page are not visible since the website only shows the latest comments.

different forms of online textual contents. More participants in online environment increases the likelihood of more valuable ideas being inserted to the collective knowledge of the crowd. Although as Kittur, Chi and Suh argue this growth increases the information entropy in the space hence making it harder to find useful information for participants [Kittur et al., 2009]. This inspires new research for building scalable self-organized systems.

As James Surowiecki [Surowiecki, 2005] argues there are four criteria for "Crowd Wisdom" to emerge:

1. *Diversity of Opinions*: It expands the set of possible solutions
2. *Independence:* Surowiecki defines it as the relative freedom from the influence of others and not just isolation
3. *Decentralization:* people should be able to specialize and draw from local knowledge
4. *Aggregation:* a mechanism to turn individual judgments to collective intelligence

Diversity of Opinions is one of the most challenging criteria in online systems. The challenge is recruiting a critical mass of participants with enough diversity in opinions. Without diversity a participatory culture will not thrive and will eventually become an echo chamber for ideas where only similar ideas are repeated and challenging ideas fade into the background [Sunstein, 2001]. For a crowdbased idea generating system not only we want to have as many participants as possible but we also want enough diversity to avoid "cyberpolarization" [Sunstein, 2007]. Building a system that facilitates large-scale conversations falls under the topic of "Discourse Architecture" a multidisciplinary topic that is related to Computer-Supported

Cooperative Work (CSCW), Computer-Human Interaction (CHI), and Computer-Mediated Communication (CMC) [Sack, 2005].

Opinion Space[4] is an online tool for engaging the crowd in generating ideas in different areas. So far Opinion Space has been used for political discussions around policy making, collecting ideas for advancing educational technologies, and in the ideation phase for improving the quality of American vehicles. However, this technology can be used in various areas.

Opinion Space borrows from different research ideas in the HCI and Computer Science:

- Social incentives and game mechanics
- Deliberative polling
- Dimensionality reduction and machine learning
- Visualization
- Crowdsourcing

Brief introduction to Opinion Space

In the first iteration of Opinion Space the system solicits numerical ratings from each participant on a number of controversial statements. A participant will use a slider to state her agreement with each statement from a strongly disagree on a continuous scale to strongly agree. These statements are designed such that they have minimum correlation with each other and can differentiate and segment differen types of participant. A participant rate each statement based on how much she agrees or disagrees with that statement. Each participant also provides an answer to a qualitative question like "How can an American vehicle manufacturer improve the quality of its cars?". The system then takes the numerical ratings and runs a dimensionality reduction on these numbers from many users[5]. Each participant then sees a two-dimensional visualization of all the ideas, the visualization is designed such that each idea has an equal likelihood of beefing seen by others. This will effectively place all the participants onto a level playing field.

Opinion Space 1.0 is shown in Figure 6.4. In this Figure each participant is represented as a dot in the visualization. The dimensionality reduction model is designed such that it places participants with similar opinions proximal and the ones with different opinions far apart. One of the goals of Opinion Space is to move beyond the simple right/conservative and left/liberal polarity. Opinion Space, as we will see in later chapters, can highlight clusters of participants in the system. Similar to CONE-Welder there are also gamification elements in the system to increase engagement and participation. Each participant can earn a point by

[4]available at http://opinion.berkeley.edu

[5]Later in this dissertation we extend this model and include textual comments in the dimensionality reduction as well

reading another participant's comment. To make the system self-organizing in addition to the game mechanics a simple spacial reputation system is used for ranking the comments. The reputation system was presented in the dissertation of the former member of the lab [Bitton, 2012]. It works like this if a person agrees with a comment that is far from his comment in the Euclidean space the author of the comment will receive more reputation points than the case that the two comments are close to each other in this space. The first version was made available to the public on March 28, 2009. Opinion Space 1.0 attracted 21,563 unique visitors of which 4,721 registered and provided their email address. In this uncontrolled experiment each registered user on average rated 14.2 comments.

The uncontrolled experiment of Opinion Space 1.0 was well received by the audience but in order to validate our hypothesis that the visualization is effective we followed up with an in-lab experiment with five hypothesis. Each hypothesis was formed to evaluate one aspect of Opinion Space in terms engagement, leading towards finding diverse opinions, etc.

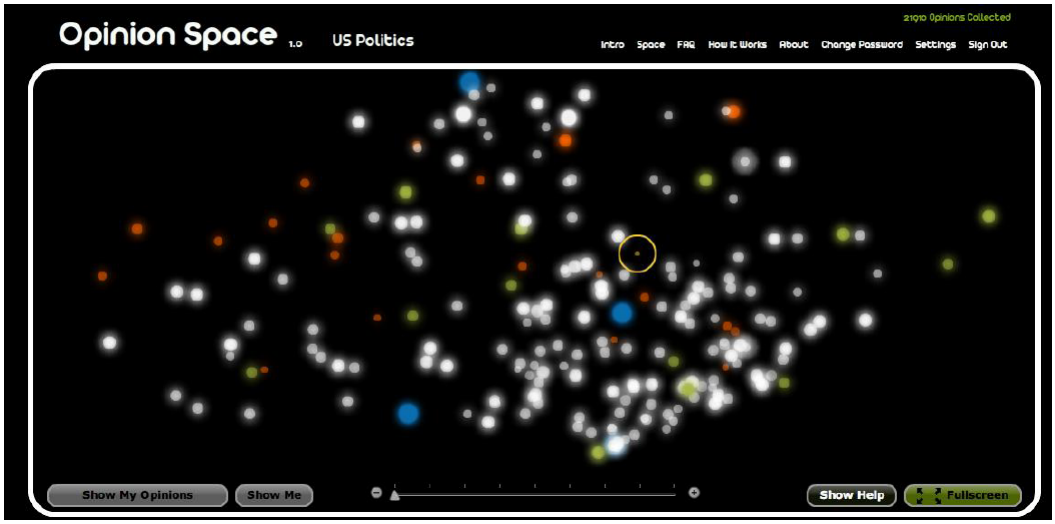


Figure 6.4: The visualization of the Opinion Space 1.0 two dimensional map. Each participant is presented as a dot in this map and a participant can click on other dots to read other views. Proximity of the dot to each participant's own dot represents the similarity between their ratings to the initial propositions. Position of the current participant is shown by a dot with a halo. Comments that have received positive ratings are colored in green and red dots show participants/comments that have received negative overall ratings. The size of the dot shows the magnitude of the overall rating (meaning that the most agreeable comments should be a large green dot).

After reviewing related research in this chapter we present the experiment design and results of hypothesis testing.

6.2 Related Work

Online Political Discussions, Deliberative Polling and Opinion Mining

Bernisky argues that public opinion polling is one of the most effective means of including the feedback from the public in the participatory democracy however he emphasizes that this process is not unbiased [Berinsky, 1999]. The term "Deliberative Polling" was first proposed by Fishkin in 1991 [Fishkin and Luskin, 2005]. In deliberative polling participants are first polled on a set of issues or ideas, they are then allowed to discuss and deliberate among themselves and at the end the pollster conducts another round of polling. Fishkin argues that the results from these polls are a significant signal of how the understanding of the public changes as they become more informed about the original issue. Luskin *et al.,* demonstrate a successful use of deliberative polling around the issue of rising crime in Britain [LUSKIN et al., 2002]. In a most recent line of work Kriplean *et al.,* present ConsiderIT, a crowdbased website for introducing citizens with pros and cons of ballot initiatives [Kriplean et al., 2012]. Each participant in ConsiderIT curates a list of pros and cons from a list of available items for each ballot initiative. They argue that this personal deliberating tool helps citizens make a more informed decision about each ballot initiative. Similar to ConsiderIT, Opinion Space is a personal deliberating tool, where participants can inform each other asynchronously by providing their textual idea and by reading each other's ideas.

The process of deliberative polling can be active or passive. We define active deliberative polling as the cases that a participant is solicited for an idea and voluntarily provides her idea to the system. Systems like Opinion Space and ConsiderIT are examples of active online deliberative polling systems. In later sections we introduce one of our experiments, Social Space, which is an example of a passive online deliberative polling system. In the passive deliberative polling different ideas around an issue is collected from public online contents like blogs and tweets through Opinion Mining. Pang and Lee [Pang and Lee, 2008] present a detailed overview of different techniques for opinion mining and sentiment analysis. Sentiment Analysis and Opinion Mining is often used for predicting the results and understanding the polarity of political and consumer opinions [Pang and Lee, 2008]. In later chapters we provide a novel sentiment analysis tool based on Canonical Correlation Analysis and highlight its strengths over traditional tools.

Visualizing Social Networks

Visualization of social networks has been an active area of research in recent years. For example LinkedIN INmaps demonstrates the professional network of each person based on connectivity of each note to others.

Social networks analysis and the study of social networks is still an active research area [Carrington et al., 2005]. Freeman [Freeman, 2000] reviews different methods of visualizations of social networks. Viegas and Donath [Viégas and Donath, 2004] visualize social networks and patterns that are formed by email messages. Morningside Analytics (http://morningside-analytics.com/) uses text analysis to provide visual representations of

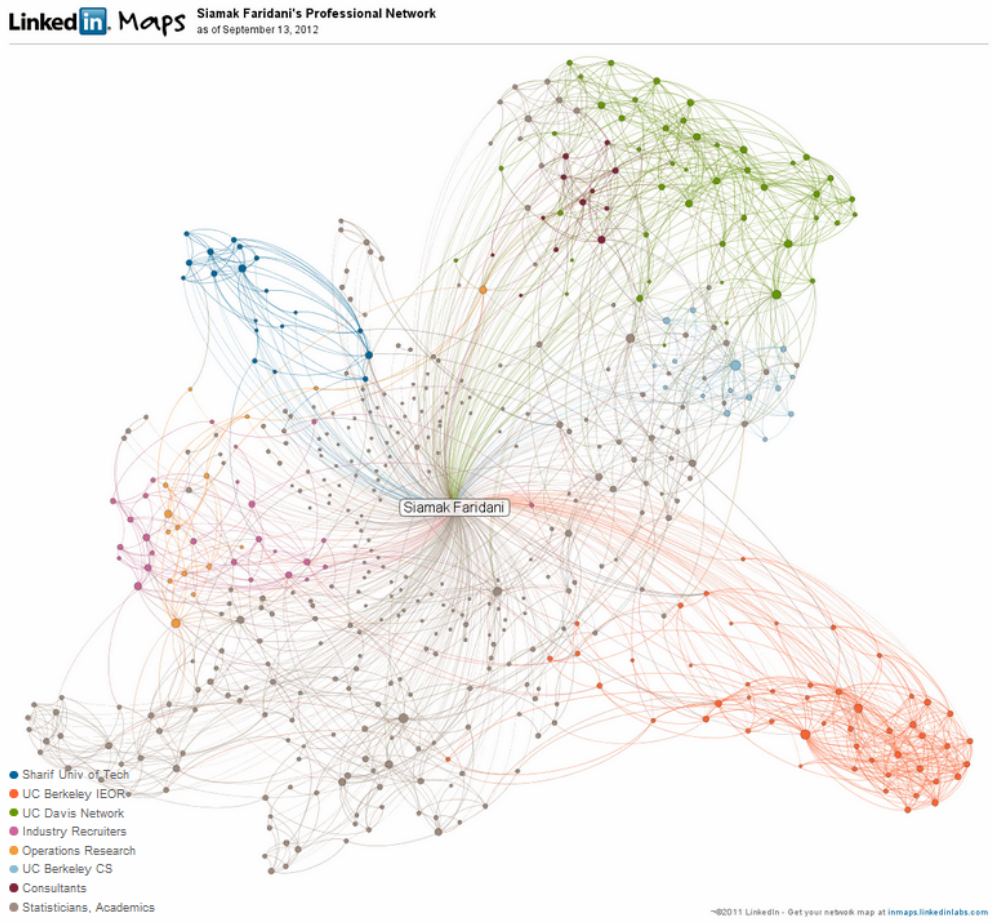


Figure 6.5: A LinkedIN inmap allows a user of the website to visualize his or her professional network based on interconnections between her friends. It also allows the user to color code each segment of their network based on their mutual workplace.

social network data. Other examples are presented in Sack et al. [Sack, 2000]. Similar to Opinion Space, SocialAction visualizes different social network measures [Perer and Shneiderman, 2006]. Heer et al., built Vizster, a visualization interface for social networks like Orkut and Friendster [Heer and Boyd, 2005]. In Vizster users explicitly input their mutual friendship to visualize their social network.

Increasing Participation in Online Communities

Some of the underlying factors that encourage user participation in online forums is discussed by Bishop [Bishop, 2007]. Ludford et al. [Ludford et al., 2004] highlight that diversity of users in online forums increase participation. This participation is increased when users are informed about how unique they are in the community. Opinion Space uses both Bishops

and Ludfords findings to encourage more positive participation by visualizing the diversity of Opinions. For more studies about participation in online forums see Brandtzaeg and Heim [Brandtzæg and Heim, 2009].

6.3 Opinion Space

Opinion Space was designed by a group of researchers and graduate students at Berkeley. This section is an overview of the project and goes into how the system works. We built different versions of Opinion Space and each varies in the way it works. In this section we only go through the first iteration of Opinion Space known as Opinion Space 1.0.

User Registration and Entering Initial Opinions

When a participant enters the system[6] she is first presented with five initial propositions. These initial propositions are often important controversial that allows the system to efficiently segment different types of participants. Figure 6.6 demonstrates an example of these propositions for a participant. Each participant enters her opinion on a numerical scale using a horizontal "slider" below each proposition (Fig. 6.6). Positioning slider on the most left position means that the participant is strongly disagreeing with the proposition. Similarly the most right position means that the participant is strongly agreeing with the proposition. The system then stores each position as a numerical value in the database.

Opinion Space was initially inspired by the 2008 US presidential election as a result most of the propositions were focused on US domestic politics. The original propositions were as below:

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

These issues were thought to be a predictor for a participant's position on the main question. In the first version of Opinion Space we were only able to use these numerical values to generate the visualization and we were not able to utilize the textual responses to the main question yet[7]. Below are some of the questions that were used in Opinion Space

[6]Different versions of Opinion Space are accessible at http://opinion.berkeley.edu/

[7]In later chapters we provide a solution to this problem

as main topics. Note that in each iteration of Opinion Space only one discussion question was used.

- Every woman is entitled to have as many children as she wants, even having octuplets via in vitro fertilization.
- Professional journalists provide regular reports on issues like these. How might your political views change if newspapers were no longer available?
- 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?
- 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?

The last question about the legalization of Marijuana was the question that we used for all the experiments in this chapter. After a participant provides her numerical agreement rating on each proposition, she enters her textual response to this question. She is later able to change both her numerical and textual responses if she wants to.

6.4 Opinion Space Map

After a participant enters her responses. The interactive map of Opinion Space will be generated by the server 6.4. In this interactive map a participant can click on each dot and see other participants' responses. The yellow point with the halo indicates the location of the active participant. The dimensionality reduction model is designed such that it places similar participants closer to the active participants. In later chapters we will evaluate the accuracy of the dimensionality reduction. A participant uses two other horizontal sliders to enter how much she agrees with this new comment and how much she respects the opinion of this person. Our in-lab experiments confirmed that participants could easily differentiate between the meaning of these two sliders.

In Opinion Space greener points are the ones that are marked as agreeable by the active participants. Similarly, red points are the people who were rated negatively by the active participant. The size of each point in the map indicates the weighted average user rating associated with that comment. The larger and brighter the comment means that the comment is more agreeable to a diverse group of users rather than the ones that share the same belief. The spatial model for overall ratings is described in [Bitton, 2009].

Is it possible to give different participants of Opinion Space the opportunity to share a common view. A participation can post a link to her Opinion on social networks like

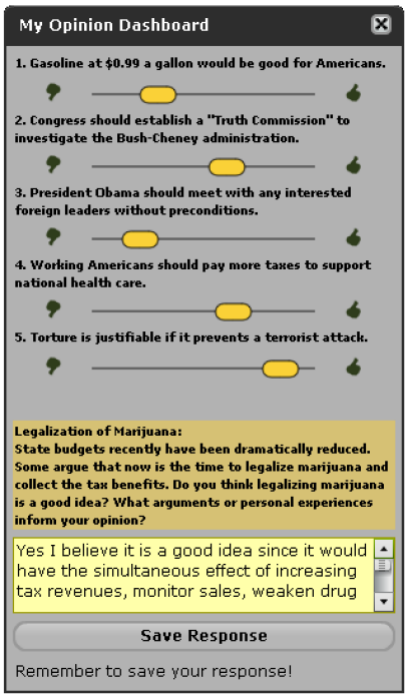


Figure 6.6: Five propositions and one main question is presented to each participant. A participant then enters her numerical ratings using the horizontal sliders and provides her textual comment in the textbox.

Twitter[8] and Facebook[9]. While this is similar to the CommonSpace project by Willett *et al.* [Willett et al., 2011], In Willett *et al.* analysts can explore different visualizations of the data and annotate them with their comments. Since each person sees a different view, they can share a snapshot of their view and the next analyst can then click on the shared snapshot to go to that specific visualization. In opinion space everyone sees the same map and the interactive map is not customized for the user. As a result a participant can share a link to her comment in the space on an online social networking website. When another participant clicks on that link, since the visualization is the same, it is guaranteed that this new participant will see the same view as original commenter has seen. Although having different type of views for different people might be advantageous in Opinion Space. For example diversity averse people may like a dimensionality reduction method that preserves local structures (like t-SNE [van der Maaten and Hinton, 2011]) while diversity seeking participants may like visualizations that preserve global structures.

[8]http://www.twitter.com
[9]https://facebook.com/

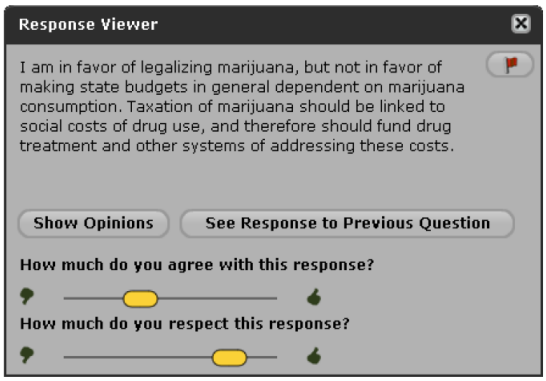


Figure 6.7: Users read each comment and rate it based on how much they agree with the comment and how much they respect it.

Dimensionality Reduction Model for Opinion Space 1.0

It is well established that human mind interprets two-dimensional visualizations much better than three-dimensional visualizations [Sebrechts et al., 1999]. In the process of analyzing high dimensional data and to be able to visualize the data in a two-dimensional map we use a dimensionality reduction algorithm. Most dimensionality reduction algorithms assume that the data is sparse, and lie on some lower dimensional manifold or subspace [Chen et al., 2001, Eckart and Young, 1936, Belkin and Niyogi, 2003, Tenenbaum et al., 2000]. It was possible to use different dimensionality reduction techniques in Opinion Space. For example Factor Analysis (FA) is one candidate for this purpose [Johnson and Wichern, 2002]. Primarily developed by psychometrics researchers, Factor Analysis seeks to find the covariance relationships among many observable variables in terms of few latent variables[10] called *factors* [Johnson and Wichern, 2002]. The first iteration of Opinion Space used a classic multivariate statistical analysis tool known as Principal Component Analysis (PCA) to reduce the dimension of proposition ratings from five dimensions to two dimensions. In Opinion Space 1.0 the textual comments were not used to develop the interactive map. This assumption is relaxed in later chapters by using another dimensionality reduction model.

In its simplest form a PCA can be explained as a low rank approximation method. As explained by Candes et al. [Candes et al., 2009], if we assume that the datapoints are stacked in a column matrix X. The low-rank approximation of the matrix X can be shown as:

$$X = L + N$$

In which L is a low-rank matrix and N is a small perturbation matrix. In Opinion Space we solve a rank-2 approximation of the model by solving the model below. This is the least-square interpretation of PCA:

[10]In statistics and machine learning a latent variable is an unobservable variable that is often inferred

$$\begin{aligned} minimize &= ||X - L|| \\ s.t. &= rank(L) \leq 2 \end{aligned}$$

It is easy to show that this problem will is equal to a variance maximization technique meaning that it this will find the projection in which the variance of variables after projection is maximized [Izenman, 2008, Jolliffe and MyiLibrary, 2002]. In other words if the data is a random vector $\mathbf{X} = (X_1, ..., X_r)^T$. A linear lower dimension projection $\boldsymbol{\xi} = (\xi_i, ..., \xi_t)^T, (t \leq r)$ is constructed by using coefficients b:

$$\xi_j = \boldsymbol{b}_j^T \mathbf{X} = b_{j1}X_1 + ... + b_{jr}X_r, j = 1, 2, ..., t$$

Izenman shows that the solution to the PCA selects coefficients $\mathbf{b}$ such that the variance of projected variables $var\{\xi_j\}$ are arranged in descending order and their are mutually uncorrelated ($cov\{\xi_i, \xi_j\} = 0, \forall i < j$). Opinion Space uses the first two principal components of the covariance matrix of the data to produce the map. The two principal component of the data account for variation in the higher dimensional data.

Since monitors these days have move pixels along the width, in the Opinion Space map the x axes is used for the first principal component and the y axes shows the variance in the second principal component. Other than PCA there are dimensionality reduction models that each have different propertied [Fodor, 2002]. For example Neighbor Embedding models like t-distributed stochastic neighbor embedding (t-SNE) preserve the local properties of the high dimensional data but the global distances do not heave any meaning [van der Maaten and Hinton, 2008]. While there are other dimensionality-reductions like ICA, t-SNE and multi-dimensional scaling (MDS) that could have been used in Opinion Space 1.0 we selected PCA for its scalability.

6.5 USER STUDY

Three interfaces were created by the author. He we call the three interfaces List, Grid, and Space (the last most similar to Opinion Space 1.0). The interfaces were built in Adobe Flash, the same technology that was used for Opinion Space. We populated interface with a set of 200 participant comments randomly selected from the "in the wild" experiment. We first ran an pilot experiment with 9 participant and formed and refined five hypotheses. To confirm the initial hypotheses about the effectiveness of dimensionality reduction and engagement of the interface we performed a follow up in-lab within-subject study with 12 participants using the three interfaces. In the study the the Space interface which was a scaled down version of Opinion Space was the experimental condition and the List and Grid interfaces were the control interfaces. As participants used the interfaces we recorded their ratings and times and also video taped their screens.

List Interface

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. The list interface (shown in Figure 6.8) is designed such that is replicates the way comment lists work in blogs and other internet websites. We present the 200 comments in the list in random order. "Dwell Time", the amount of time that each participant spends on each comment is calculated by the system. Neighboring comments are blurred in order to accurately measure the time that a person spends on each comment. As the participants moves her mouse pointer over a comment the comment get de-blurred.

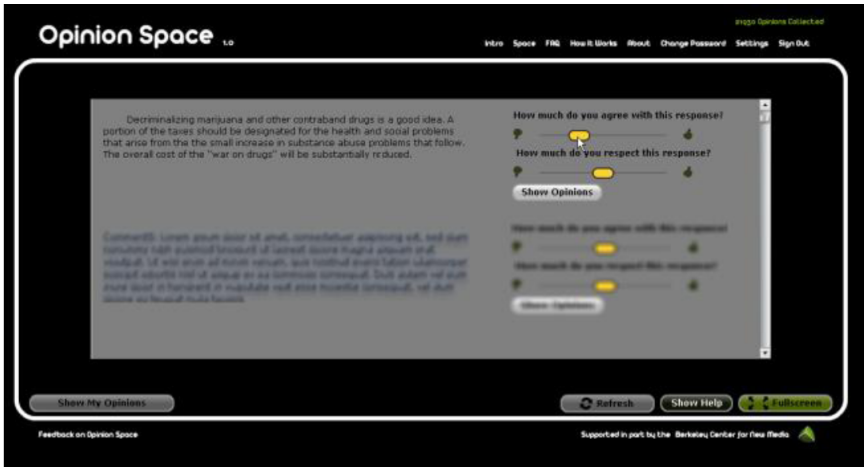


Figure 6.8: List interface presents the comments in chronological order

Grid Interface

The Grid interface (shown in Figure 6.9) was designed as a control interface to represent elements of both the *Space* interface and the *List* interface. Comments in the Grid interface are randomly ordered and presented in a grid as points. But each point, like the points in Opinion Space, is scaled according to its overall rating. Unlike Opinion Space 1.0 the proximity of points from each other is meaningless. This allows us to test the hypothesis that the dimensionality reduction is engaging. Participants were allowed to click on points in any order they wished.

Space Interface

Shown in Fig. 6.4, the space interface is the simplified Opinion Space with a selected number of comments. Some of the features of Opinion Space such as the "twinkling" of points were turned off to avoid intentional bias and influence of participants.

Figure 6.9: Grid interface. Comments are presented in random order like the List interface but each comment is scaled based on its overall rating like the Space interface.

6.6 Hypotheses

Based on the pilot experiment with 9 participants and our observations from the in-the-wild experiment we formed five hypotheses. *Hypothesis 1 (H1):* The average dwell time in Opinion Space is larger than that in the List and the Grid interfaces showing the the participants were more engaged in the Opinion Space environment [H1a] and we expect to confirm it with users' responses to the exit poll [H1b].

Hypothesis 2 (H2): Participants will find Opinion Space as a tool to find more useful comments and ideas.

Recommender engines often work based on recommending the closest items to the user's preference [Resnick and Varian, 1997] this may reduce the possibility of exposing a subject to useful comments that are not necessarily close to her own opinion [Pariser, 2011]. One of the main goals of Opinion Space was to prevent the "Filter Bubble" from occurring in ideation and political discussions. As a results Opinion Space enables participants to view ideas that challenge their own ideas. The diversity of the comments read by each participant is defined as the average Euclidean distance of all the comments viewed by the participant from her own point in the two-dimensional map. This inspired the third hypothesis:

Hypothesis 3 (H3): The average diversity of the comments that participants read in Opinion Space is significantly larger than the ones in the List and Grid interfaces.

The size and brightness of a point in the Space and Grid interface corresponds with the insightfulness of the comment associated with that point. We expect that this UI element in the Grid and Space interface to help participants find more insightful ideas. Hence the fourth hypothesis was formed.

Hypothesis 4 (H4): Compared to the List and Grid interface the participants in Opinion Space will agree more with the comments that they read than the participants in the List

and Grid interface. Similarly the fifth hypothesis was:

Hypothesis 5 (H5): Participants in Opinion Space will, on average, rate the comments that they view higher than the ones that they view in the List or in the Grid interface.

6.7 Method

In order to evaluate the hypothesis a within-subject study was designed to test the Space interface against the traditional List interface and the Grid interface was used as the second control interface. One of the benefits of the Grid interface was that it could isolate some UI elements in the Space interface such as the brightness and the size of dots while the ordering of comments was similar to the list interface. In the within-subject experiment each participant used all three interfaces. The order in which the interfaces were presented was selected randomly to reduce the effect of priming the participants.

Participants

We posted ads across campus and on facebook and twitter. 36 volunteers responded to these adds and completed an online pre-screening survey. The survey was designed to make sure that the participants do not have previous exposure the the Opinion Space interface.

The questions in the pre-screening test were as below:

- Nickname (You can use a nickname, it does not need to be your real name)
- Email Address (We will use this email to contact you)
- Age
- Gender
- Level of Education
- To which political beliefs (or party) do you generally subscribe?
- Which mainstream news sites do you read on a weekly basis? cnn.com (CNN), Nytimes.com (The New York Times), Foxnews.com (Fox News), USAtoday.com (USA Today), News.google.com (Google News), Other
- Which social news sites, blogs, or discussion forums do you read on a weekly basis? Slashdot, Reddit, Digg, BoingBoing, Huffingtonpost.com, Metafilter, Other
- Approximately How many comments you leave on websites per week
- When you visit a discussion forum, are you typically seeking to: (1) Learn more about the topic (2) Read controversial statements (3) Argue with extremists (4) Read a range of arguments in order to solidify your own opinion on the matter (5) Other:

- How tech savvy are you (0-10)
- How familiar are you with the current political issues (0-10)
- How many hours a day do spend reading online news/blogs or watching videos online (1) Less than one hour (2) 1-2 (3) 3-5 (4) 5-8 (5) 8+

Another goal of the pre-screening process was to select the participants that are informed about current issues and could understand current discussions in the comments. Each participant was rewarded with a $10 Amazon gift card at the end of the experiment. From the 36 volunteers who responded to the pre-screening survey 12 people were selected for the experiment. Two of the participants were female. Three were Republican (25%), five Democrats (42%) and 4 Independents (33%). More demographic information about the participant population is provided in 6.10.

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

Figure 6.10: Demographic information of the participants

6.8 Experiment Protocol

We allotted one hour for each participant. We setup the experiment such that participant's screen will be shown to the person who is running each session. This will allow the administrator to help the participant with her questions.

After starting the experiment the participant was asked fill out an entrance poll to determine her level diversity aversion [Munson and Resnick, 2010]. Below are the questions that are used for the entrance poll:

- Nickname
- Have you heard of or played with Opinion space before?
- Please rank the following politicians in terms of how closely you expect to agree on the five propositions: Nanci Pelosi, Ralph Nader, Arnold Schwarzenegger, and Rush Limbaugh (Where 1 is the closest and 4 is the furthest)

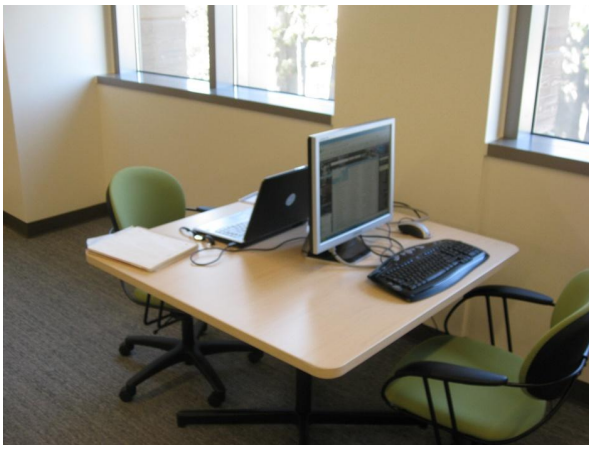


Figure 6.11: The experiment setup. The administrator could see the screen of the participant and help with any question. The complete session was also screen recorded for studying the navigatio strategies for users.

- When visiting an online discussion forum, are you typically more interested in viewing responses by users who disagree with your opinion or those who agree? (1) I am more interested in reading comments by those who disagree with me on the topic. (2) I am more interested in reading comments by those who agree with me on the topic. (3) I'm not sure.
- How confident are you about your stance on the legalization of marijuana (0-5)

After finishing the entrance poll the participant was asked to enter her proposition ratings and her textual response to the question about the legalization of marijuana. The experiment administrator will load one interface. Each participant will see all the interfaces but the order in which interfaces are presented is random. Participant is free to switch to the next interface as long as she has rated at least 10 comments using that interface. The maximum allotted time for each interface in this experiment was 15 minutes meaning that the system automatically switches to the next interface if the participant has not asked for the next interface in 15 minutes. A short questionnaire was presented to the participant after each interface. This was considered self-reported data and was later used to compare behavioral data with self-reported data.

- What strategy did you use to explore the comments in the space?
- Please indicate to what degree you agree with the following statements.
  - I found this version of the system enjoyable to use. (1-5)
  - I learned something interesting while using this version. (1-5)
  - This version is conducive towards finding useful comments. (1-5)

Each interface recorded the dwell time for each comment. After finishing all three experiments participants were presented with an exit poll in which they had to rate each interface on on a series of 7 qualities.

- Re-evaluate individual versions. In this section, you will be asked to re-evaluate each of the three versions you explored. Please indicated on a sliding scale to what degree you agree with the following statements.
  - Enjoyability
    - I found the List version of the system enjoyable to use.
    - I found the Grid version of the system enjoyable to use.
    - I found the Space version of the system enjoyable to use.
  - Learned something interesting
    - I learned something interesting while using the List version of the system.
    - I learned something interesting while using the Grid version of the system.
    - I learned something interesting while using the Space version of the system.
  - Finding useful comments
    - The List version of the system is conducive towards finding useful comments.
    - The Grid version of the system is conducive towards finding useful comments.
    - The Space version of the system is conducive towards finding useful comments.
- *Rank the versions* Please rank the versions (in decreasing order of preference) according to which you would prefer to use in the following situations.
- Which version would you prefer to use if you had to rate 2,000 comments?
- Which version enabled you to read more insightful comments?
- In which version are you more likely to leave your own comment or response?
- In which version are you most likely to rate the comments of others?
- Which version is more familiar to you in terms of existing systems?
- Which version would you prefer to use if you wanted to participate in a discussion about US politics?
- In which version do you expect to spend more time reading comments and browsing?
- In which version do you expect to spend more time reading comments posted by other people?

- Which version highlights the most insightful comments?
- Which version are you most likely to visit repeatedly to follow the discussion as it evolves?
- In which version did you see more diversity among comments?
- Which version do you prefer overall?
- **General questions**
- Do you find having two sliders to be confusing?
- Do you see any value in having a second slider for indicating how much you respect the comment?
- Did you find the size of a dot to be a good indicator of how compelling the corresponding comment is?
- Do you have any other comments about the three versions?

6.9 RESULTS

The mean and the standard deviation of the number of comments that each participant rated in the three interface is shown in Table 6.12. The mean dwell time and "agree" and "respect" values were also reported in the table. Note that we are using a continuous scale between 0 and 1.

	List	Grid	Space
Average number of Comments Rated	23.5 ± 11.2	20.9 ± 9.9	21.1 ± 9.0
Average Dwell Time per Comment (sec)	516.4 ± 242.5	458.4 ± 180.4	582.9 ± 187.1
Average "Agree with" Rating	0.443 ±0. 266	0.515 ± 0.278	0.567 ± 0.269
Average "Respect for" Rating	0.396 ±0. 294	0.479 ± 0.300	0.510 ± 0.284

Figure 6.12: Results of the recorded data for participants (mean ± standard deviation)

Summary of the self-reported data through questionnaires is presented in 6.13. We asked participants how enjoyable, interesting, and useful each interface was and they provided their responses on a 1 to 5 Likert scale. In the exit survey we asked the participants to rank each

interface and compare them together. The summary of exit survey results are available in Table 6.14.

	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

Figure 6.13: Participants rated each interface on a Likert scale. The table highlights the mean for the rating on how enjoyable, interesting and useful each interface is. In the self-reported data the Space interface was leading in all aspects. We later performed a statistical hypothesis testing to see if these higher means are statistically significant.

Analyzing participants' fatigue

As noted in previous sections we presented interfaces in random order. The question is "do users behave differently, in terms of engagement, when an interface is presented in different orders. To see the effect of orders on the total time spent in each interface we ran a two-way ANOVA analysis on the average time spent on the first, second and third interfaces and found out that the order in which the interfaces are presented had zero effect of the average time spent in them. The p-value of the ANOVA test was 0.534 suggesting that the fatigue did not have any significant effect[11]

6.10 Evaluation of Hypotheses

To analyze different types of data (e.g., rank ordered, continuous, Lickert, etc) we used different statistical methods. For example Student t-tests and Welchs test were used for comparing two means. Analysis of Variance (ANOVA), ANOVA on Ranks, , Friedmans test were used for cases that we have multiple treatments (e.g., when we compare the three interfaces). And before doing ANOVA tests we performed Bartletts test for homogeneity of variance. Statistical hypothesis testing often analyzes the mean and variance of data in two population to calculate the probability that the two populations are drawn from the same

[11]The maximum amount of time for the whole experiment for each participant was 45 minutes.

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%

Figure 6.14: Each interface was ranked against the other two in the exit survey.

distribution. This probability is often named the "p-value". Lower p-values correspond to greater significance levels.

ANOVA assumes that residuals are distributed according to a normal distribution and have equal variances. KruskalWallis and the Friedman test are nonparametric versions of ANOVA and are used when the normality assumption is violated. ANOVA is used to reduce the possibility of *type I error* that arises in the cases that multiple t-test is performed [McKillup, 2006]. For large degrees of freedom ANOVA is relatively robust to the assumptions homogenous variances and normal error but these assumptions need to be checked for small datasets [Larson, 2008]. Hence before running an ANOVA we perform a Bartletts test to check for the assumption of homogeneity of variances (homoscedasticity) to hold. A high p-value for Bartletts test confirms that one can go ahead and perform a follow up ANOVA test.

Statistical Tests for H1

Participants had read 959 comments using the three interface (List, Grid, Space). Form this 329 comments were read on the List interface, 285 on the Grid and 345 were read using the space interface. Participants had read the most comments on the grid interface. Our experiment system recorded all of the dwell times for these comments allowing us to compare the average amount of time that participants spent on one comment in each interface. Figure 6.12 lists the average dwell times for each of the interfaces. We use these dwell times and self-reported ranking data to evaluate Hypothesis 1: *Hypothesis 1 (H1): The average dwell time in Opinion Space is larger than that in the List and the Grid interfaces showing the the participants were more engaged in the Opinion Space environment [H1a] and we expect to confirm it with users' responses to the exit poll [H1b].* Bartlett's test rejected the homogeneity of variances for dwell times, as suggested by [Conover and Iman, 1981] a two-way, within-subject ANOVA on ranks was performed on the dwell times data for each interface. The p-value ($1.098 \times 10^{-14} << 0.05$) suggested that the type of interface is a contributing factor to the dwell time values. After running ANOVA a follow up test should be done to highlight which differences were significant. We used Welch's t-test which a generalization of t-test for cases that the two datasets do not have equal variances and the homoscedasticity assumption does not hold [McKillup, 2006]. The pairwise Welch's t-test highlighted that when we compare either the Grid or the Space interfaces with the List interface, dwell times are significantly larger than the List interface (For Grid-List $p = 2.2 \times 10^{-16} \ll 0.05$ and for List-Space $p = 5.387 \times 10^{-10} \ll 0.05$ both well below the significance level). When we compared the dwell times in the Grid interface and the Space interface we observed no significant difference ($p - value = 0.1126 > 0.05$). This established that there are elements in Opinion Space interface that are causing participants to spend more time on that interface.

For further exploring self-reported data we used Friedmans test, a nonparametric extension of ANOVA for cases that the data is ordinal ranks similar to the data in tables 6.13 and 6.14 [McKillup, 2006]. To evaluate H1a on the self-reported date we chose a Friedman's test with a Wilcoxons signed-rank as a post test on the data collected on "In which version do you expect to spend more time reading comments?" in Table 6.14. The p-value for the Friedman's test was $p = 0.0000984 \ll 0.05$ suggesting that the type of interface is a determining factor on the level of preference for participants. The nonparametric Wilcoxons signed-rank test is used as a pairwise post-test. Wilcoxons for Grid-List was $p = 0.02332 < 0.05$ and for Grid and Space is $p = 0.02351 < 0.05$ and for Space-List is $p = 0.002608 < 0.05$ all of these are below the significance threshold suggesting that the self-reported data supported H1a.

In the exit survey, participants were asked to rank the interfaces based on their personal preference. We used this data to asses hypothesis H1b. As shown in Table 6.14 nearly 92% of participants preferred the Space interface over the List and Grid interface (H1b). Similar to H1a a Friedmans ANOVA and a Wilcoxons signed-rank post-test was performed on the data from the exit survey. The p-value for the Friedmans ANOVA was $p = 0.000486512 \ll 0.05$ and for the post tests for List-Space $p = 0.01188 < 0.05$, for Grid-Space $p = 0.03884 < 0.05$ and for Grid-List $p = 0.0209 < 0.05$. They all supported that participants like the Space

interface over both the List and the Grid interface.

Statistical Tests for H2

Hypothesis 2 (H2): Participants will find Opinion Space as a tool to find more useful comments and ideas.

This hypothesis was confirmed using the data collected from questionnaires after each interface 6.13. Participants reported that they had found more useful comments in Opinion Space that the other two interfaces. On this data Friedmans test gave a p-value of $0.00361 \ll 0.05$ and the follow up Wilcoxons post-test confirmed that for all pairs the ordering is significant ($p_{Grid-Space} = 0.003583$, $p_{List-Space} = 0.01868$ and $p_{Grid-List} = 0.03667$. This supported hypothesis H2.

Statistical Tests for H3

Hypothesis 3 (H3): The average diversity[12] of the comments that participants read in Opinion Space is significantly larger than the ones in the List and Grid interfaces.

We calculated the average diversity for participants in each one of the interfaces. The Euclidean distance in five dimension between two participants can be at most 2.23 units. For the Space interface the average diversity value was .960 and this value was .924 and 0.992 for the List and Grid interfaces. The data passed the homogeneity test through the Bartlett's test $p = 0.1628 > 0.05$ and the p-value for ANOVA was $0.7848 \gg 0.05$. This shows that there is not difference between the average diversity of comments read in each interface and this rejects hypothesis H3. However, as shown in question 6 of the exit survey in 6.14 50% of participants reported that they were able read a more diverse set of comments in Opinion Space compared to the Grid (33%) and the List (16%).

Statistical Tests for H4

Hypothesis 4 (H4): Compared to the List and Grid interface the participants in Opinion Space will agree more with the comments that they read than the participants in the List and Grid interface.

The total number of comments that participants rated were 782 comments. From which 281 of them were read in the list interface, 249 in Grid and 252 were shown and rated in the Space interface. The range for the rating is from 0 to 1. Zero is considered when the participant strongly disagrees with the comment and 1 is when she is strongly agreeing. The mean and standard deviation of these values for each interface is reported in Table 6.12. The data passed the homogeneity of variances assumption and the p-value of Bartlett's test was $p = 0.850 \gg 0.05$. The ANOVA analysis resulted in p-value of $0.00002073 \ll 0.05$ suggesting

[12]The average diversity of a set of comments is defines as the average Euclinead distance of the participant from the comment that she reads

that the interface was a contributing factor in the difference between agreement values. We performed a follow up analysis using pairwise t-test $p_{Grid-Space} = 0.03335$, $p_{Space-List} = 0.000000149$ and $p_{List-Grid} = 0.002115$ supporting hypothesis H4.

Statistical Tests for H5

Hypothesis 5 (H5): Participants in Opinion Space will, on average, rate the comments that they view higher than the ones that they view in the List or in the Grid interface.

Similar to H4, we record the ratings for the "respect" value as shown in Table 6.12. The data passed the assumptions for ANOVA and an ANOVA analysis yielded p-value of $p = 0.001105 \ll 0.05$. As a follow up analysis a two-tailed t-test was performed and showed that the "respect" value for the Grid and Space interface (both use Opinion Space UI elements to highlight insightful comments) was higher than the List interface (p-values are: 0.0007299 for List-Grid and 0.00003479 for List-Space). There was no significant difference between the Space and Grid interfaces (p-value of 0.1191). We belive that it is due to the fact that they both use similar visual elements to highlight insightful comments. These elopements include adjusting the brightness and the size of the points for each person.

6.11 Discussion, Conclusion and Future Work

The main problem with list-based interfaces is that they do not scale as the number of participants and comments grows. Traditionally the solution to this problem has been showing the most recent comments or showing the best comments based on binary votes like thumbs ups and thumbs downs. Opinion Space was designed as an scalable, self-organizing tool to collect ideas from the crowd. One of the design objectives of Opinion Space was to enable participants to find the opinions that are different than their own. When we compared Opinion Space with the control interfaces we found that Opinion Space engages the participants more both in terms of the dwell times and user perceived engagements. Our results agreed with the findings of Ludford et al [Ludford et al., 2004]. Meaning that participants agreed more with the comments when their were shown their position in Opinion Space (H4) and they also find the comments that were presented in the Space interface more insightful (H5).

Not surprisingly, the third hypothesis was not supported by experimental results. Participants did not read a more diverse group of comments in the Space interface. The control interfaces also had a relatively large comment diversity. This might be due to the inherent randomness of chronological ordering. In later chapters we focus on improving the underlying dimensionality reduction method in Opinion Space. We also

Chapter 7

Opinion Space, Textual Comments and Canonical Correlation Analysis

7.1 Introduction

In the previous section we focused solely on using numerical proposition ratings for dimensionality reduction. In this chapter we explain how textual comments can be combined with numerical ratings in order to better provide a lower dimension embedding. Canonical Correlation Analysis (CCA) is being used to visualize textual opinions and we explain this method in detail. We then provide results of using CCA on our dataset. Our results suggest that CCA provides better dimensionality reduction quality than the PCA technique that is currently being used in Opinion Space. Additionally, we show how CCA can be used as a topic model to label different regions in the space.

7.2 Dimensionality Reduction Techniques and Textual Responses

In the current version of Opinion Space, Principal Component Analysis is used to project the opinions from a five dimensional space to a two dimensional map. Canonical Correlation Analysis enables us to consider both the statement ratings and textual responses in our dimensionality reduction model.

7.3 Canonical Correlation Analysis

Canonical Correlation Analysis was first introduced by Harold Hotelling in 1936. It has been used when two correlated sets of arrays are present and it finds a linear combination of these two arrays that maximizes the correlation between these two arrays [Hardoon et al., 2004]. Originally proposed by Hotelling as a multivariate analysis method [Hotelling, 1936],

Canonical Correlation Analysis has been used for text processing [Blitzer et al.,], multivariate prediction [Abraham and Merola, 2005, Rai and Daumé III, 2009], data clustering [Chaudhuri et al., 2009], data visualization [Sun and Chen, 2007, Lai and Fyfe, 2006], image retrieval, and search [Hardoon et al., 2004].

In Opinion Space CCA assumes that the two input vectors from the text and sliders are correlated. For the implementation of CCA we have used the technical report by Bach and Jordan [Bach and Jordan, 2005] and have implemented the CCA model based on [Hardoon et al., 2004]. In their report, Bach and Jordan show that the maximum likelihood estimation leads to the canonical correlation directions. The graphical representation of CCA is shown in Fig 7.1. In Opinion Space x_1 is the vector of statement ratings and x_2 is the vector of featurized comments.

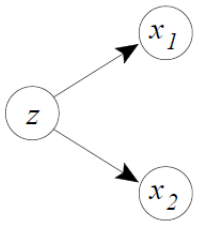


Figure 7.1: Graphical model for canonical correlation analysis ref [Bach and Jordan, 2005]

Canonical Correlation Analysis Formulation

Hotelling in 1936 proposed the following for Canonical Correlation Analysis: CCA finds basis vectors for two sets of correlated multidimensional variables such that the correlation value of the projection of the two variables onto these basis vectors are maximized [Hardoon et al., 2004].

Two sets of multidimensional variables might be highly correlated, but since the correlation analysis strongly depends on the coordinate systems that the variables are explained, this correlation may not be immediately visible. As a result CCA seeks to find two sets of linear transformations such that the two multidimensional variables after transformation are maximally correlated [Hardoon et al., 2004].

In this section we follow the explanation given by Hardoon et. al. [Hardoon et al., 2004]. Let's say we are given a multivariate random vector (x, y). An instance of this vector in the form of $S = ((x_1, y_1), ..., (x_n, y_n))$ is in hand. Let's define vectors S_x as $(x_1, ..., x_n)$ and S_y as $(y_1, ..., y_n)$. We now consider the direction w_x and project x onto this direction to get a definition of x in a new coordinate system.

$$x \rightarrow \langle w_x, x \rangle$$

Where $\langle w_x, x\rangle$ is the inner product of vectors w_x and x and equals $w_x^T x$ (and w_x^T is the transpose of w_x). We can now look at elements of S in the new coordinate system, and thus we will have

$$\begin{aligned} S_{x,w_x} &= (\langle w_x, x_1\rangle, ..., \langle w_x, x_n\rangle) \\ S_{y,w_y} &= (\langle w_y, y_1\rangle, ..., \langle w_y, y_n\rangle) \end{aligned}$$

CCA seeks to maximize the correlation between S_{x,w_x} and S_{y,w_y} thus the goal is to find w_x and w_y such that the following objective function (ρ) is maximized.

$$\begin{aligned} \rho &= \max_{w_x,w_y} corr(S_{x,w_x}, S_{y,w_y}) \\ &= \max_{w_x,w_y} \frac{\langle S_{x,w_x}, S_{y,w_y}\rangle}{\|S_{x,w_x}\|\|S_{y,w_y}\|} \end{aligned}$$

Hardoon et al. [Hardoon et al., 2004] show that this formulation becomes an eigenproblem.

7.4 CCA as a manifold alignment technique

CCA falls under a topic called "Manifold alignment". Manifold[1] alignment assumes that high dimensional can be projected onto a common manifold. The idea was first proposed in Ham, Lee and Saul in 2003[Ham et al., 2003]. Manifold Learning seeks to find and recover a lower-dimension manifold when the data can be projected onto a linear subspace[Izenman, 2008].

According to [Wang et al., 2011a] manifold alignment is defined as the "unifying representation of multiple datasets". Manifold alignment techniques are seeking to find the relationship of different datasets to one latent space. [Wang et al., 2011a] look at the graph Laplacians associated with each dataset and extract the local geometry using that. The goal of all manifold alignment techniques is to find a lower dimensional manifold that at the same time preserves higher dimensional correspondence across different datasets. The idea behind the manifold alignment techniques is that the high dimensional data can be projected onto the lower dimensional manifold discretely approximated by a graph. Early methods of extending linear dimensionality reduction models to nonlinear models involved improving the results from PCA and MDS to better capture the nonlinear nature of the data[Kohonen, 1988, Bishop et al., 1998, Silva and Tenenbaum, 2003].

Isomap: Unlike PCA and multidimensional scaling, Isomap is capable of discovering nonlinear degrees of freedom[Tenenbaum et al., 2000]. Isomap builds on the linear dimensionality reduction, MDS, by using a three step procedure. In the first step it finds the

[1] A manifold is a space for which any small enough neighborhood around each point resembles a Euclidean space. See [Cayton, 2005] for more.

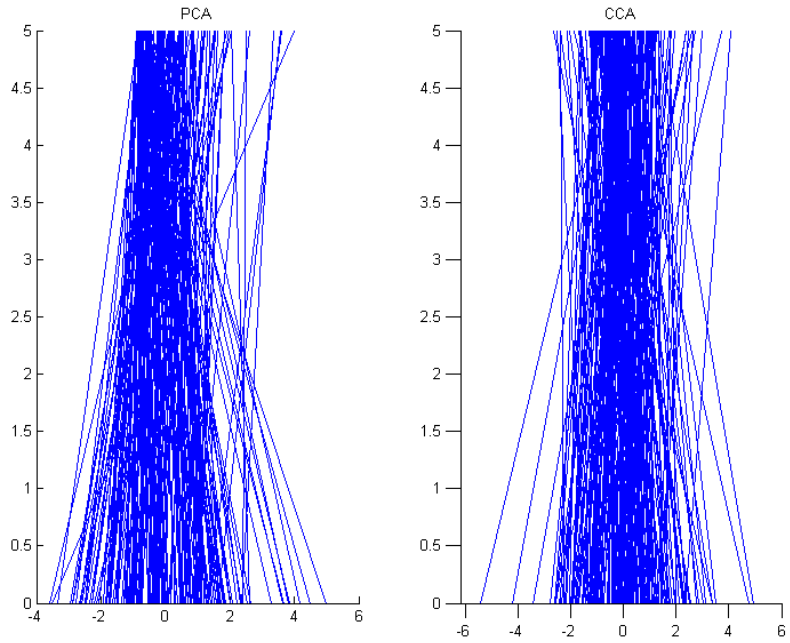


Figure 7.2: Using CCA as a manifold alignment technique

neighboring datapoints on the lower dimension manifold by looking at the distance between all pairs in the higher dimension input space. In the second step, Isomap finds the lower dimension distances between pairs of datapoints by running a shortest path algorithm on a graph generated in the previous step. Finally, it runs the classical MDS algorithm on the distance matrix generated in the second step. Similar to PCA and MDS, Isomap is guaranteed to produce better approximation as the amount of input data increases[Tenenbaum et al., 2000]. In this dissertation we only focus on linear CCA.

7.5 Using CCA for Opinion Space

The corpus in this study is the text from 531 responses with highest confidence. We have developed the following algorithm to be used on this corpus. Algorithm 5 summarizes this approach.

Algorithm Layout

Algorithm 5: Performing CCA on the comments feature vector and proposition ratings

Input: Set of comments and numerical values for each person
Output: 2D locations in the canonical space
foreach *Comment i* **do**
    Remove stop words (i.e. and, the, he, they);
    foreach *word j* **do**
        Stemm each word j by porter stemmer using the nltk library for Python (example: ponies $\rightarrow$ poni);
        Assign each stemmed word to one of the top 9 clusters (to make a dense featured vector);
        Append the new array Y_i to the featurized word vector Y
Feed the featurized comments matrix and numerical statement ratings to a CCA solver;
Output 2D locations ;

In following sections we explain each step in our algorithm.

Featurization

We use the "bag-of-words" approach. Each comment is split into individual words and is filtered by using a list of `stop words`. The remaining is passed through the `Porter Stemmer`. We then count the frequency of each word appearing in the comment [Manning et al., 1999]. The 9 clustered were manually and intuitively chosen based on topics that appeared in the comments. More clusters mean more sparse representations for comments while fewer clusters may combine too many dissimilar topics into one. An LDA model was later used to do an automated topic clustering but later in the evaluation section we show that the manual clustering showed 90% improvement over the LDA automated clustering.

7.6 Cluster Analysis and Region Labels

One interesting aspect of CCA is that it provides and interesting topic model. Recall that CCA gives linear transformations w_x and w_y that can translate two vectors (featurized comments and numerical values) to a 2D canonical space. We can alternatively use w_y^{-1} to go from the canonical space to the text space. So each point in the canonical space will have a likelihood of each topic associated to it. By numerically integrating over a region we can find the main topic in that specific region. The following simple algorithm shows how we can use CCA to extract topics from the corpus.

Algorithm 6: Labeling the regions in the canonical space

Input: 2D projections in the canonical space w_y^{-1}
Output: Topic labels for each region
Run a K-means on the list of comments and cluster them into k clusters;
foreach *cluster* c **do**
  initialize vector $r_c = [0, 0, ..., 0]$;
  foreach *point* $p_{x,y}$ **do**
    Calculate $r_c = r_c + w_y^{-1} p_{x,y}$
  Tag the cluster with the topic that has the maximum value in r_c

A representation of clusters is shown in 7.3. The algorithm can be summerized as the following procedure:

1. The CCA projection is calculated in our Kernel CCA solver from 1826 comments (originally we took about 2040 but some of them are not valuable or contains spam) we use 1638 Keywords for the feature vector.

2. We take the projection and cluster them using a k-means clustering algorithm

3. In the canonical space each point in the 2D space has keywords associated to it (with specific probability of occurrence)

4. We numerically add these probabilities and find the keyword with the highest probability for each region and visualize it

De-stemming (mapping from the stems back to the words) is done manually by visually examining the top stems. For most of the string processing We are using the python NLP package called nltk. We also wrote a smaller NLP package called "PyNLP" that is available for researchers to use[2].

7.7 Topic clusters

For this project we have manually selected 9 main topics as following. Note that each keyword is presented by its stemmed form.

MiddleEast (palestin) (israel) (palestinian) (middl) (east) (region) (iran) (isra)

US (american) (clinton) (america) (citizen) (usa) (secretari)

economy (govern) (econom) (invest) (resourc) (interest) (monei) (china)

democracy (democraci) (polit) (support) (polici) (diplomaci)

[2]https://github.com/faridani/PyNLP

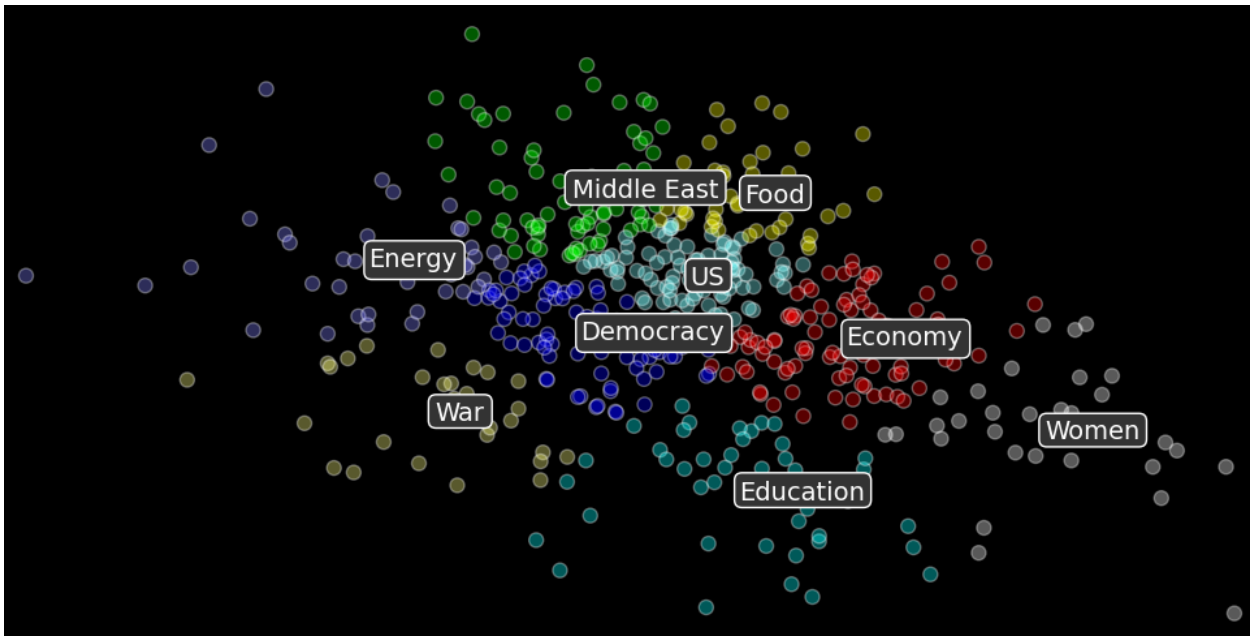


Figure 7.3: Cluster Analysis: CCA enables us to look at how participants' responses cluster in the 2D space. Opinion Space 2.0 dataset is used for this analysis and participants provided responses on what would they tell Secretary Clinton if they see her. Responses varied from issues about women rights to energy policies. We used the cluster analysis and the region labeling detailed in section . As we see CCA has placed topics like *Women* and *Education* near one another. Also *Middle-East* and *Energy* are also placed close to each other.

food (food) (import) (aid)

educ (educ) (student) (visa)

war (war) (conflict) (militari) (threat) (secure) (peac) (stabil)

women (women) (human) (right) (peopl)

energi (energi) (climat) (nuclear)

7.8 Results

In this chapter we projected each user by the new method (CCA) and by the older method (PCA). We then looked at the distance between each two users and looked at their agreement rating. The correlation between these two values are shown in the following table. One of the fundamental assumptions in Opinion Space is that similar opinions are placed closer to each other. As we see CCA provides the highest correlation and we conclude that CCA is a better dimensionality reduction method when compared to PCA. Additionally we can

DM Method	Pearsons Correlation
CCA	-0.352
PCA	-0.134
Random	0.002

Table 7.1: State Department Dataset

run PCA on the fearurized textual responses. This approach is also compared with CCA in Table 7.3. Based on this idea, we built an Opinion Space for Twitter that is called "Social Space". Interested readers can download the source code for Social Space from the author's Github page[3]. A similar idea can be used in systems like ConsiderIT[4]

This evaluation method is illustrated in Figure 7.4.

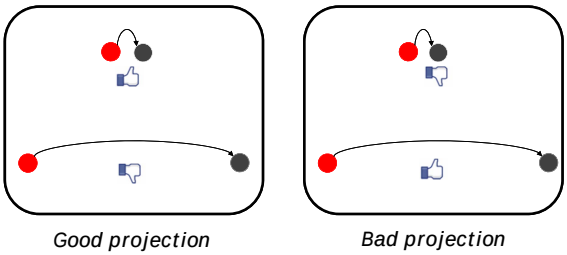


Figure 7.4: Similar participants often agree, dissimilar participants often disagree, we compute the correlation between the Euclidean distance and agreement ratings

Additionally we looked into how participants have rated other comments. In the State Department dataset, the total number of ratings recorded is 21,191 ratings. This means that if we consider a directional graph with all the participants as the nodes, an arc from node A to node B exists if participant A has rated the comment of participant B. Out of 21,191 links in this graph, 20721 of them were among pairs in which only one of the participants has rated the other comment and there is no arc going back from the commenter. There are only 235 links that are bidirectional (meaning we have two rating values, one from A to B and one from B to A). We calculated the correlation coefficient between the values of the forward arcs and the backward arcs and they show the positive correlation coefficient

[3]https://github.com/faridani/Social-Space

[4]We can take all the "pro" points that are collected in ConsiderIT and run a topic model on them (for example we run David Blei's LDA or a pLSI). We then have, for each comment, a distribution of topics presented in it. For example if we run this for legalization of marijuana we will have these topics (tax, state revenues, prisons, gateway drug, ...). we can then run our PCA on the output of LDA and generate a 2D map which places the comments about taxes and state revenues closer and will put comments about prisons and minorities close together too. This setting will allow a user to read only a couple of comments in each cluster and form her opinion and then move to a new cluster. It will expose a user to all different ideas. For more on ConsiderIT please see this video https://vimeo.com/35645960

Table 7.2: Collected ratings

Number of ratings collected	21,191
Number of pairs of people that have both rated each other	235 (470 ratings)
Overall number of participants	5,711
Number of participants who have rated other comments	1,610
Number of comments that have been rated	1,376

of 0.349. Therefore from the dataset we have a very small set of ratings (470 out of 21191 ratings) that are between the same two pairs and they show a positive correlation to each there. For extreme cases when we have only two commenters S and T, and S rates T highly positively (1) and T rates S highly negatively (-1), the correlation coefficient that we use is undefined (since their distance is the same say x= [1,1] and their rating vector is y=[-1,1] and since correlation value is calculated by $corr(x, y) = \frac{E[(x-\tilde{x})(y-\tilde{y})]}{\sigma_x \sigma_y}$) . Hence the value of correlation is not defined in this extreme case (0/0). But the empirical results from the data show that this case does not happen. The Table 7.2 summarizes these results from our dataset. We have also included a histogram of ratings for participants (7.5). This histogram shows a long tail.

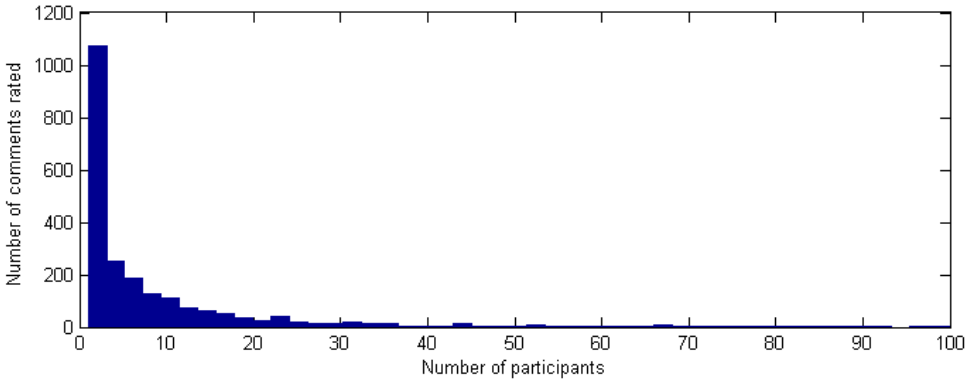


Figure 7.5: Histogram of the number of comments rated for participants.

Zero distance textual responses

Reducing the dimensions of the text space to only 9 topics will cause multiple comments to be featurized into the same numerical vector. A graph of these comments is shown in Fig. 7.6. Each disjoint graph shows a group of comments that are being reduced to the same feature vector. One of the advantages of CCA is that it will include the numerical ratings in addition to these featurized vectors. This allows Opinion Space to differentiate different opinions even if their textual repousse is being reduced to the same feature vector.

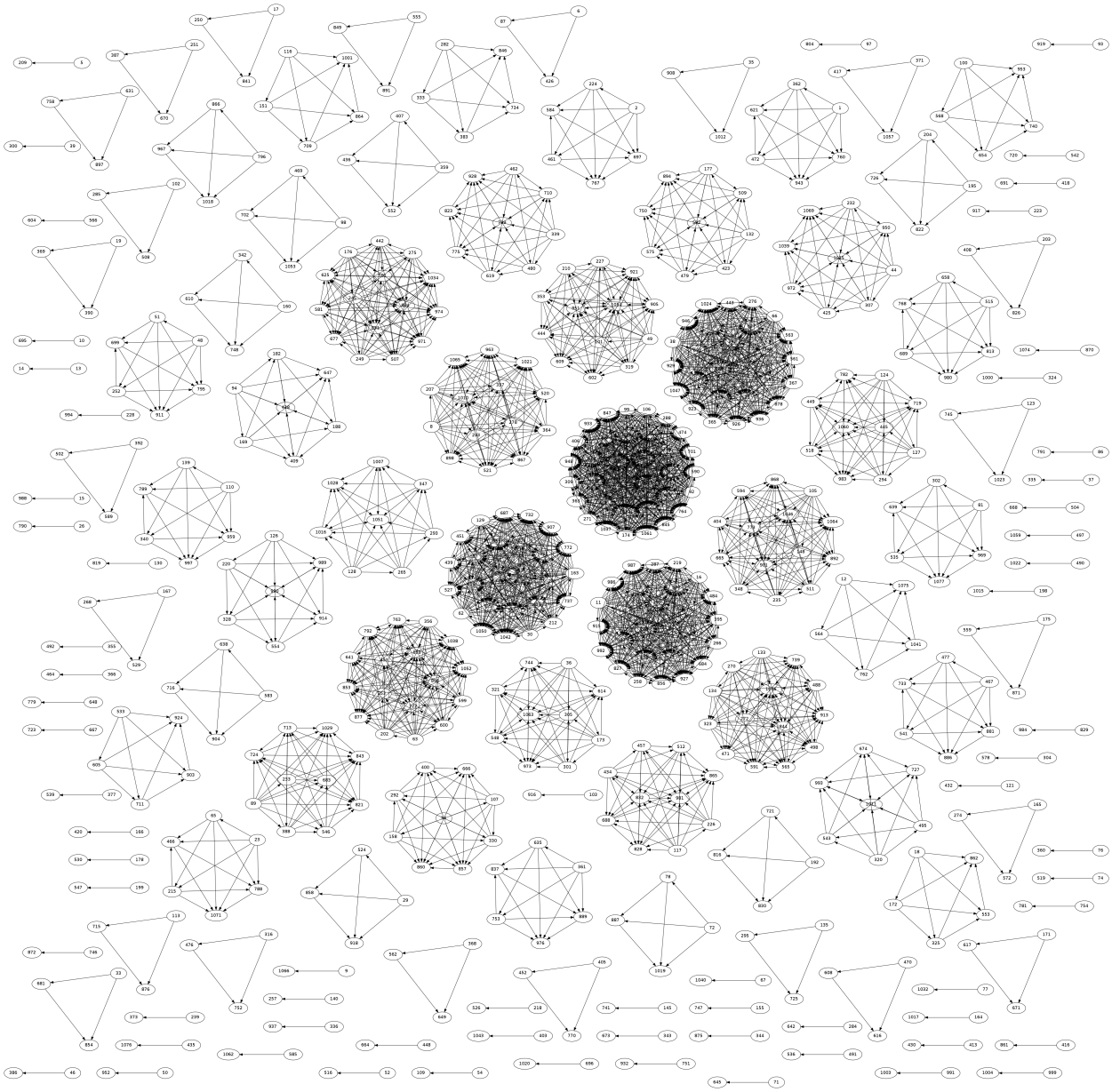


Figure 7.6: Cluster of comments that are reduced to the same numerical feature vector.

This type of clustering allows us to quickly identify the textual responses that talk about the same subjects. Figure 7.7 shows the number of times a keyword from a topic appears in the corpus collected from the deployment of Opinion Space for a car company.

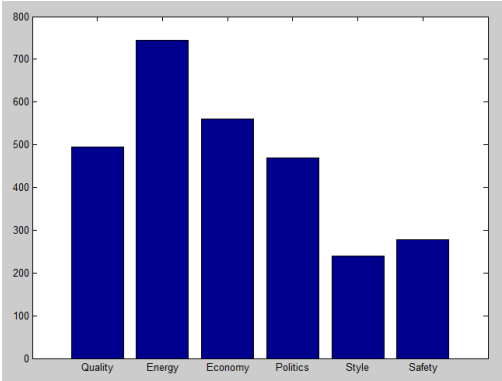


Figure 7.7: Histogram for the topics in the GM corpus

DM Method	Pearsons Correlation
Random Baseline	0.00%
PCA on Featurized Textual Responses	3.46%
PCA on Numerical Ratings	4.09%
CCA (Individual uncertainty model)	7.58%
CCA (Gaussian Model)	9.85%
CCA (varying the weight of convex combination)	10.07%

Table 7.3: CCA Analysis for the GM Dataset (The Gaussian Model and the individual uncertainty model are explained in more details in the next chapter

DM Method	Pearsons Correlation
Correlation Coefficient for 6 dim text	1.20%
Correlation Coefficient for 5 dim ratings	3.13%

Table 7.4: GM Dataset (with no dimensionality reduction)

7.9 Extending the CCA model to include comment ratings in the projection

So far in our work we have used CCA on initial inputs from users. In the current model, as users rate other people the projection will stay the same (since comment ratings are not included in the projection). Here we propose a method to extend the current CCA model to consider comment ratings in the final projections. Meaning that as users provide more comment rating we update the projection mappings and will update their location in the two dimensional space.

Description of the Proposed Mathematical Model

In our current implementation of CCA we have two sets of vectors. The vectors of proposition ratings X (which has 5 elements for each user). For example

$$X_{12} = (.4, .5, 0, 0, .9) \tag{7.1}$$

and the featurized comment vectors Y (which is a bag-of-words representation of the comment provided by users:

$$Y_{12} = (\text{“}like\text{”} : 3, \text{“}I\text{”} : 2, \text{“}Beautiful\text{”} : 1, \text{“}mediocre\text{”} : 0, ...) \tag{7.2}$$

Let's assume user 12 gives user 24's comment the rating of $\beta = 0.7$. We assume this rating is solely based on the comment that user 24 has provided and shows slight agreement of user 12 with the comment provided by 24. We can update the featurized comment of user 12 by some weights of the featurized vector of user 24. In the formulation below w is some weight that is used for mixing the original comment with the rated comments.

$$Y_{12} := Y_{12} + w(\beta - 0.5)Y_{24} \tag{7.3}$$

In general if user i has rated comments in the set C_i the former formulation will become:

$$Y_i := Y_i + w \sum_{j \in C_i} (\beta_j - 0.5)Y_j \tag{7.4}$$

The new value of Y_i can go into the CCA solver to update the Canonical projection.

7.10 The Diversity Donut: A user interface element to enable participant control over the diversity of visible comments

Based on the CCA projection, we designed Diversity Donut (Figure 7.8), an HCI tool for directly indicating the level of diversity that they want to see in Opinion Space.

Diversity Donut is an interactive recommender tool based on direct manipulation [Shneiderman, 1997]. Participants indicate their desired diversity by drawing an outer circle and inner circle of a “donut” on the 2D visualization of the Opinion Space. This allows for more control in the visible set of comments specially when the number of comments is large.

Diversity Donut is based on the idea presented in Munson and Resnick [Munson and Resnick, 2010] that participants in online environments can be clustered into two disjoint groups: “Diversity Seeking” and “Challenge Averse”. Diversity Seeking participants actively look for opinions that challenge or oppose their current stance on issues while Challenge Averse participants only look for ideas from like minded people. To evaluate the effectiveness of the tool we performed an experiment with 13 participants. The details of the tool are

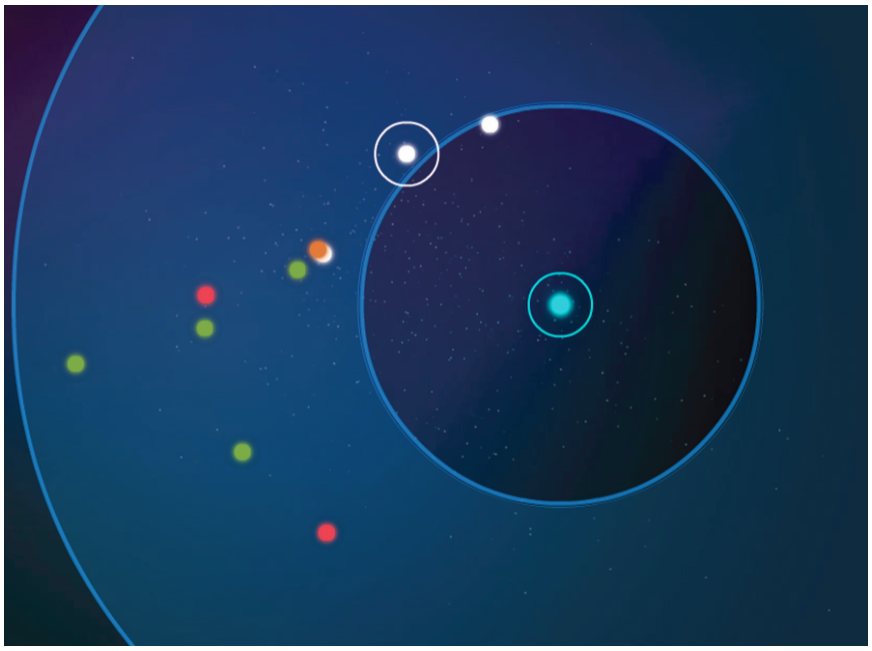


Figure 7.8: The Diversity Donut is a user interface element that allows direct manipulation of the diversity of visible comments in the 2D visualization in Opinion Space.

presented in [Wong et al., 2011]. Participant self-reported data suggests that participants found the Diversity Donut to be an effective tool for recommending diverse responses. We refer interested readers to [Wong et al., 2011] for the details of the design of the study.

7.11 Applications of the Model: Using Canonical Correlation Analysis for Generalized Sentiment Analysis, Product Recommendation and Search

Standard Sentiment Analysis applies Natural Language Processing methods to assess an "approval" value of a given text, categorizing it into "negative", "neutral", or "positive" or on a linear scale. Sentiment Analysis can be used to infer ratings values for users based on textual reviews of items such as books, films, or products. We propose an approach to generalizing the concept to multiple dimensions to estimate user ratings along multiple axes such as "service", "price" and "value". We use Canonical Correlation Analysis (CCA) and derive a mathematical model that can be used as a multivariate regression tool. This model has a number of valuable properties: it can be trained offline and used efficiently on a live stream of texts like blogs and tweets, can be used for visualization and data clustering and labeling, and, finally, it can potentially be incorporated into natural language product search algorithms. At the end we propose an evaluation procedure that can be used on live data

when a ground truth is not available.

Product reviews on websites sometimes allow for ratings on a number of different dimensions. For example the online shoes and clothing store, Zappos[5], allows customers to review each pair of shoes on six numerical dimensions (comfort, style, size, width, arch support and overall). Similarly TripAvisor[6], a website for reviews and advice on hotels and flights, allows users to rate each hotel on six dimensions (value, rooms, location, cleanliness, service, and sleep quality). In addition to these numerical values, each reviewer provides a textual review of the product or service. In traditional approaches to recommender systems these textual reviews are sometimes ignored because of the complexity that they introduce to the models. In this paper we utilize these numerical and textual feedbacks to train our model. A new user can then express the properties of her desired product either in textual form or on numerical scales. We later show that our model is capable of using either of these sets of inputs to come up with a set of product recommendations for the user.

We use Canonical Correlation Analysis (CCA) to perform an offline learning on corpuses that have similar structures to Zappos and TripAdvisor in that they provide both textual reviews and numerical ratings. CCA is often used when two sets of data (x and y) are present and some underlying correlation is believed to exist between the two sets [Lai and Fyfe, 2006]. In our model x is the featurized representation of the textual review (i.e. an N-gram or a tf-idf representation of the text) and y is the vector of numerical ratings for each review. We hypothesize that combining both texts and numerical values enriches the recommendations and training. By using the data collected from our system, Opinion Space[7], we have validated this hypothesis. We have also developed an evaluation framework that enables us to test our models on live data when a ground truth is not present.

The mathematical structure of CCA allows for separation of offline learning and online use. Expensive learning processes can be done offline on the dataset, and learned mappings can be performed efficiently on live data streams coming from twitter and blogs. Additionally CCA considers the interdependence of response variables and we use this property to design a sentiment analysis model. This capability that we call "Generalized Sentiment Analysis" can be used for predicting the sentiment and its strengths on a number of different dimensions even in cases in which these dimensions are not independent from each other. Additionally, CCA can give the variance of the predicted values on different scales, giving a confidence value for each predicted value.

7.12 Generalized Sentiment Analysis

Sentiment Analysis is traditionally performed on one attribute of the target products. We extend the model by looking at many different dimensions of the product together. CCA provides a supervised learning model for extracting the attributes of products and services

[5]http://www.zappos.com

[6]http://www.tripadvisor.com/

[7]http://www.state.gov/opinionspace/

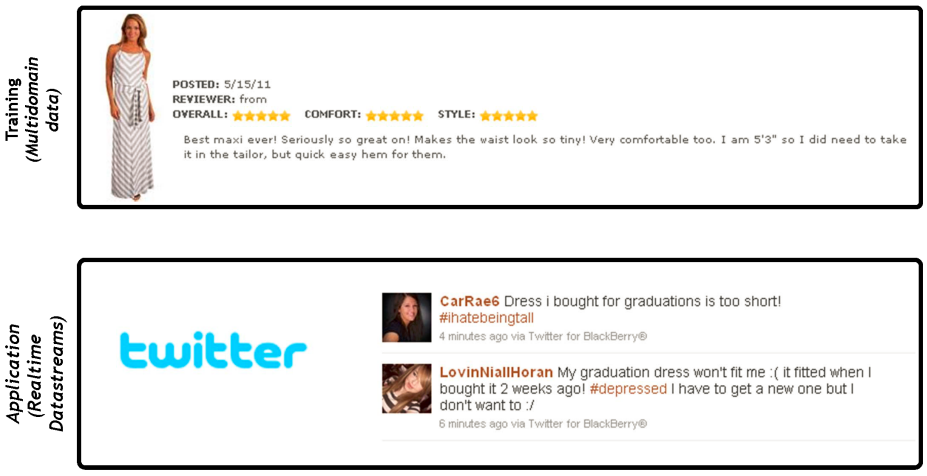


Figure 7.9: In the example above we can train the model on dress reviews that are available on Zappos and then use the trained model as a regression model to find the expected numerical ratings for the live text streams from twitter. This model can be combined with product recommender systems for twitter users, or used as a sentiment analysis tool on live data.

from textual reviews. Unlike univariate Support Vector Machine models (SVM), CCA allows us to consider the interdependence among response variables.

One aspect of the CCA model is that it can be trained on datasets like Zappos or TripAdvisor and then used online to extract the sentiment of the market from sources like blogs and tweeter feeds that lack the numerical value for reviews. It can also be used to highlight the key words that contribute to major changes in the numerical scales. We can calculate the effect of increasing the frequency of each word to the changes in each numerical scale. For example we hypothesize that words such as "comfortable" in "this shoe is comfortable" will cause major changes in the numerical value of the "comfort" scale while in the sentence "my uncle wears this shoe" the term "my uncle" will cause no change in the numerical value of comfortability. Also fitting parameters to the CCA model that is the most expensive part can be done offline. The online procedure which is multiplying the learned transformation matrices (w_x and w_y) with the featurized text can be done cheaply in real-time. Following applications are proposed for this model: filling the missing values for reviews (for example if the design of the website is changed and there is no numerical values for reviews before some certain time), inferring the expected ratings for unstructured reviews that are expressed outside the company's website (for example if we observe a blog post that reviews a hotel we can use our CCA model to find the expected ratings associated with that post on different dimensions). Another application would be to have a textbox for users to enter their desired properties for their trip. For example something similar to the following: "I am looking for a hotel that is pet friendly, in a good neighborhood of the city and I don't care about hotel amenities I just want it to be affordable". Our CCA model

can then infer and extract numerical values for each dimension of the numerical scale and then by running a K-Nearest Neighbors algorithm, search on hotels that have the closest properties to the provided query. This can serve as part of a natural language search engine for products or a natural language product recommender engine.

Other benefits of the CCA model

Another benefit of the CCA model is to find the comments that are not consistent with their ratings (perhaps these comments are submitted by malicious users to promote their product, or they are simply of a very low quality). Where the two projections of $E(z|x)$ and $E(z|y)$ have a very large Euclidean distance in the canonical space we can infer that they text and ratings are inconsistent. A list of reviews that are sorted based on the value of their $\|E(z|y) - E(z|x)\|$ can be used to find these inconsistent reviews and flag them for website admins. This property is shown in Figure 7.10.

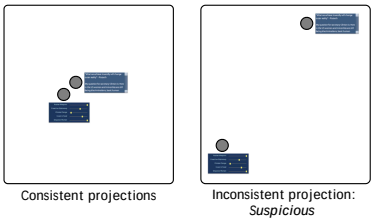


Figure 7.10: The CCA model has interesting security applications. It can flag inconsistent comments when the text and numerical inputs are not consistent (submitted by robots or is just very low quality and can be removed)

Algorithm 7: Labeling the regions in the canonical space by finding the topic with the maximum expectation in each cluster

Input: 2D projections in the canonical space Z, w_x^{-1}
Output: Topic labels for each region
Run a K-means on the list of comments and cluster them into k clusters;
foreach *cluster c* **do**
    initialize vector $r_c = \mathbf{0}$;
    foreach *point in Z* **do**
        Calculate $r_c = r_c + w_x^{-1} Z$
    Tag the cluster with the topic that has the maximum value in r_c

7.13 Next Steps

In the next chapter we look more closely at the generalized sentiment analysis problem. We explain how the CCA regression can be used to perform multi-aspect sentiment extraction on product reviews. We test the new CCA regression on product reviews collected from zappos.con and will show that our model is faster than SVM while provides a smaller prediction error.

Chapter 8

Learning to Estimate Multi-Aspect Quantitative Ratings from Textual Customer Reviews

In the process of designing recommender systems, textual data are often ignored. In the previous chapter we discussed how Canonical Correlation Analysis can improve the dimensionality reduction in Opinion Space. In this chapter we start by looking at the "Generalized Sentiment Analysis" problem where a multi-aspect rating is extracted automatically from a textual review. In an aspect-based review an item is reviewed on many different attributes. We present our regression model based on an adjusted Canonical Correlation Analysis (CCA) to perform generalized sentiment analysis. We show the effectiveness of the algorithm on a dataset that was collected from an online retailer.

Creating technologies that find and analyze online opinions is becoming key for effective recommendation systems. We propose a supervised regression learning model using a modified form of Canonical Correlation Analysis for automatically mining multi-aspect numerical ratings from textual reviews. We show that the algorithm outperforms other conventional models based on a dataset collected from an online retailer.

Major e-commerce sites such as Amazon and Netflix allow users to provide a numerical rating for an item. While these ratings provide helpful summaries of user sentiment that are very easy for automatic systems to analyze, they miss most of the fine-grained information present in textual reviews. Both humans and automatic systems have trouble processing textual reviews, but for very different reasons. Research suggests that for humans the sheer number of online opinions can be overwhelming[Pang and Lee, 2008], and relevant reviews can be hard to distinguish from irrelevant ones. Automatic systems, on the other hand, have no problem processing extremely large numbers of text reviews, but have trouble extracting useful information from raw text.

A user's "opinions" of an item can be expressed as (multi-dimensional) numerical ratings and/or as textual reviews. We propose an automatic method that leverages the information present in numerical ratings to better analyze the unstructured information present in tex-

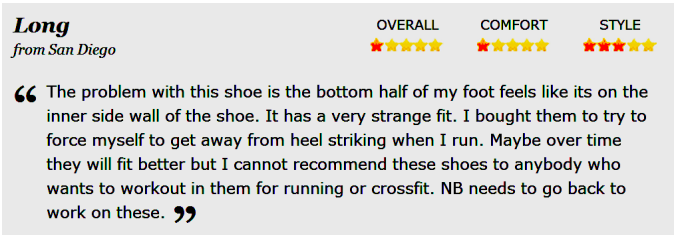


Figure 8.1: A sample datapoint collected from zappos.com. The textual review is accompanied with a three dimensional multi-aspect numerical rating.

tual reviews. We use canonical correlation analysis (CCA) [Bach and Jordan, 2005] to learn a low-dimensional projection of both numerical ratings and textual reviews simultaneously. Intuitively, when a user provides both a numerical rating and a textual review of a product, these two types of information can be thought of as two different views of the same underlying opinion. CCA seeks a low-dimensional projection of such paired data that explains the variance in both the views using a single underlying cause. This means that CCA simultaneously explains the variance in and identifies the correlations between numerical ratings and textual reviews. We give a technical introduction to CCA, and describe its interpretation as a generative probabilistic model [Bach and Jordan, 2005].

Because CCA corresponds to a generative model of both numerical ratings and textual reviews, we can use the projections learned by CCA to carry out further analysis. In particular, when numerical ratings are not provided by users, we can reconstruct them from textual reviews. We develop a regression method based on the parameters learned by CCA and apply it to datasets derived from the e-commerce site Zappos (Figure 8.1). Our experimental results suggest that our CCA method competes with state-of-the-art ordinal regression methods [Baccianella et al., 2010] used for predicting numerical ratings from text.

8.1 Related Work

A lot of work has been done on automatic review rating prediction using supervised learning, however, most of the current studies [Baccianella et al., 2010] focus on only one-dimension ordinal regression problems of inferring the "overall" rating using SVM regression methods. Qu et. al look at the problem of negation detection in the rating inference problem [Qu et al., 2010]. They consider a set of rated documents (i.e., Amazon product reviews), $\mathbf{x}_i, y_i{}_{i=1}^{N}$. In this case $\mathbf{x}i$ is a sequence of word-level unigrams $(w_1, ..., w_N)$ and $y_i \in R$ is a rating. They want to find the mapping from x_i to y_i. The bag-of-words approach has the problem of not being able to capture negations in the text. Our approach is to use a completely different supervised regression algorithm- CCA, which can naturally infer multi-dimensional aspect-based ratings in one round. There are some other existing studies on multi-aspect rating extraction, but they are also different from our work. Liu et al. [Liu et al., 2005] proposed

a system "Opinion Observer" for multi-aspect opinion analysis, but their method is mainly based on a series of heuristic rules and domain-specific opinion lexicons, which require a lot of labor-intensive human work. Wang et al.[Wang et al., 2010] also presented an algorithm to automatically extract fine-grained aspect-based ratings from overall ratings and review texts using a probabilistic topic model, and this method also involves creating domain-specific opinion lexicons.

8.2 Construction of Canonical Correlation Regression

CCA is used to analyze datasets that consist of two views. In our case, a dataset is a sequence of pairs, one for each user's opinion. Each pair consists of a feature vector x, corresponding to the textual comment submitted by the user, and a vector y of the numerical ratings submitted by the same user. We refer to the full sequence of text comment vectors as X and the full sequence of rating vectors as Y.

CCA finds vectors u and v such that the projections of X and Y onto u and v, respectively, are maximally correlated:

$$(u, v) \quad \leftarrow \quad \underset{u,v}{\text{argmax}}\ corr(u^\top X, v^\top Y)$$

Intuitively, a (u, v) pair corresponds to a single principal component discovered by PCA. As with PCA, we can use CCA to learn a sequence of directions $((u_1, v_1), \ldots, (u_d, v_d))$. In CCA, this sequence corresponds to a pair of matrices U and V that project the two views of the data into a multi-dimensional space where correlation is maximized. The additional directions are uncovered by continuing to maximize correlation, but with the constraint that the new direction must be orthogonal to those discovered so far. The details of this optimization are described by Hardoon et al. [Hardoon et al., 2004], who also show that the entire optimization can be formulated as a generalized eigenvalue problem and be solved efficiently. It is interesting to note that, unlike PCA, CCA is invariant to the scale of the data. This may be a desirable property for some applications.

Corresponding Generative Model

Bach and Jordan [Bach and Jordan, 2005] show that the optimization performed by CCA actually computes maximum likelihood parameters for a simple and intuitive generative model. Specifically, this generative model hypothesizes that the paired dataset is generated in the following way: first, a low-dimensional vector z that represents an underlying opinion is drawn from a multivariate Gaussian distribution, then a low-rank projection matrix U projects z into a higher dimensional space, where Gaussian noise is added to produce x, the text comment vector. Similarly, another low-rank projection matrix V projects z into a different higher dimensional space, where Gaussian noise is again added, this time generating

y, the vector of ratings. This process can be summarized as follows:

$$z \backsim \mathcal{N}(0, \sigma_1 I)$$
$$x \backsim \mathcal{N}(U^\top z, \sigma_2 I)$$
$$y \backsim \mathcal{N}(V^\top z, \sigma_3 I)$$

The important parameters for this model are the projection matrices U and V, since they define the relationship with the latent space of underlying opinions. Given a dataset of X and Y, with the sequence of z variables left unobserved, the choice of U and V matrices that maximize the probability of the data under the generative model are given by CCA.

The fact the CCA has a corresponding generative model is very useful. This means, for instance, that it is easy to compute the most likely underlying opinion vector z given the two views x and y, or, for example, to compute the most likely ratings vector y given the comment vector x. We will make use of both of these operations in our experiments. We omit the details of computing such quantities in order to save space, but note that descriptions can be found in Bach and Jordan [Bach and Jordan, 2005].

8.3 Gaussian Uncertainty Model

Let us assume that $S_{x,wx}$ and $S_{y,wy}$ are drawn from two multivariate Gaussian variables ($S_{x,wx} \backsim N(A, \Sigma)$ and $S_{y,wy} \backsim N(B, \Gamma)$). We assume that these two Gaussians are two different uncertain observations of the same variable. To combine the two views into one, we perform a Kalman Filter updating step on these two variables assuming that one view is the Gaussian prior and the other is the observation. The resulting distribution is still Gaussian. In the case that the dimension is reduced to one we have $S_{x,wx} \backsim N(\mu, \sigma^2)$ and $S_{y,wy} \backsim N(\nu, r^2)$ the resulting combination would be $S_z \backsim N(\frac{r^2\mu+\sigma^2\nu}{r^2+\sigma^2}, \frac{1}{\sigma^{-2}+r^{-2}})$. Note that the variance is always smaller than either of the initial variances providing lower uncertainty for the value of variable S_z. The mean for S_z can be written as a convex combination of the two means $\lambda\mu + (1-\lambda)\nu$ where $\lambda = \frac{r^2}{r^2+\sigma^2}$. The derivation for multivariate Gaussian data is available in [Thrun et al., 2005]. In this model each new datapoint (x_i, y_i) is projected onto a lower dimension space by $z_i = \lambda w_X.x_i + (1-\lambda)w_Y.y_i$

Uncertainty model for each individual response

We also consider, as an alternative, an uncertainty model for individual textual responses where the value of the weight λ'i in the above convex combination formulation varies for each data point (x_i, y_i) and is found by Equation 8.1 in which $|x_i|$ is the length of comment i and $\max|X|$ is the length of the longest comment. And the convex combination formula for z_i becomes $z_i = \lambda_i w_X.x_i + (1-\lambda_i)w_Y.y_i$

$$\lambda_i = \frac{|x_i| - \min|X|}{\max|X| - \min|X|} \tag{8.1}$$

By using the probabilistic interpretation we show that the same CCA model can be used to predict the values of one view if the values are not at hand. As shown in Figure 8.1 a better review model is followed by some e-commerce or opinion survey sites such as Zappos, Tripadviser, and Opinion Space. These websites collect multi-aspect numerical ratings in addition to textual comments. With review resources from these sites, we learn about the correlation between ratings and comments using CCA. We compare this method with a regression model for continues variables, linear regression, and with two classification models, SVM and Naïve Bayes for cases in which the values of the view are discrete. Experiential results show that the learning process in the CCA regression is significantly faster than SVM and the prediction is more accurate than both SVM and Naïve Bayes. Compared to the linear regression, our CCA regression shows more robustness to the type of featurization and provides low errors in all featurizations while maintaining a faster learning process. Linear regression gives slightly lower MSE error for the tf-idf featurization while producing significantly larger prediction errors for two other types of featurization.

8.4 Experiments

We demonstrate the effectiveness of CCA in two experiments. First, we demonstrate CCA's success as a *dimensionality reduction* method on two different datasets collected from Opinion Space. We report experiments that suggest that, compared to PCA, users consistently agree more with the comments that are placed close to them in the latent space learned by CCA. This claim is validated on both datasets. In the second experiment we use a *CCA regression* model to predict multi-aspect numerical ratings from textual reviews for shoes from Zappos. We find the CCA competes well with state-of-the-art baselines on this task.

In all experiments a bag-of-words approach is used to convert textual comments into numerical feature vectors. The text featurizer takes each review and computes feature activations by counting word types. In this work we use three types of well-known features templates: Bernoulli, multinomial, and tf-idf [Manning et al., 2008]. The Bernoulli model uses indicator features that simply indicate whether or not a word type occurs at least once in the text. In contrast, multinomial features directly specify the number of times a word type occurs. Interestingly, we found that stemming (using Porter stemmer) actually hurts performance in our model because it removes information that is useful to CCA. We also found that counting bigrams (and higher-order ngrams) instead of just single words (unigrams) did not significantly improve performance, and came at the cost of a drastically increased number of features. Thus, in our final experiments we only use unigram features.

An implementation of the featurizer is available online as part of the open source MatlabNLP library[1].

[1]https://github.com/faridani/MatlabNLP

Datasets

Next we describe in detail the datasets used in our experiments. First we describe the two datasets used to evaluate dimensionality reduction. Then we describe the dataset used to evaluate multi-apsect rating prediction.

Opinion Space

We obtained datasets from two different deployments of Opinion Space: the first by a major car manufacturer and and the second by the U.S. Department of State.

Each participant of Opinion Space enters their opinion on five statements using a numerical scale from -1 to 1. These statements are often differentiating statements like "Reducing oil consumption should be a higher priority for American car manufacturers". In addition, participants answer a discussion question with a textual response. For example, in the car manufacturer's dataset, users responded to "How can US Auto Makers improve their image among car buyers like you?". These two sources of user input comprise the two views in the dataset.

Users also expressed their agreement or disagreement with other participants' textual comments on similar numerical scales. This is a great source of information for evaluating the dimensionality reductions learned by various methods since we can compare distance in the latent space with similarities specified by users.

For the car manufacturing dataset 1,198 consumers in the automotive industry gave numerical and textual responses, and an additional 95,905 of agreement values were also collected. For the dataset from the U.S. State Department we have more than 2,400 numerical and textual responses and over 17,400 agreement values. More information about the U.S. State Department's deployment can be found on DipNote blog[2]. A detailed description of Opinion Space overall is provided in [Faridani et al., 2010].

Our dataset contained 95,905 of these comment ratings. On average each participant rated 108.7 textual comments.

Table 8.1: Dataset from Opinion Space

Overall number of participants	1,198
Number of ratings collected	95,905
Average number of ratings per person	108.7
Participants who have rated other comments	882
Average time spent on system	36 (mins)

[2]http://1.usa.gov/IbwY6b

Zappos

We used a web-crawler to gather inromation from zappos.com and collected 26,548 reviews on 1,107 individual shoes. For each review the dataset contains three discrete numerical ratings (style, comfort and overall) and the text of the review. The textual data collected as part of this dataset should be directly predictive of the numerical ratings given by the user since both are describing the same product. This property makes this dataset ideally suited for evaluating rating predictions. It is in the interest of online retailers to remove under-performing products from their inventories. As a result, ratings for products on these website are highly skewed toward positive numbers as shown in Table 8.2. Because of this property a majority predictor that always predicts a vector of fives will perform well. However, we will see that our algorithm significantly outperforms this majority predictor model.

Rating value	Style	Comfort	Overall
1	167	936	534
2	229	887	802
3	894	1866	1883
4	3,088	4,484	4,144
5	22,170	18,375	19,185
Total	26,548	26,548	26,548

Table 8.2: Number of reviews for each value of rating values

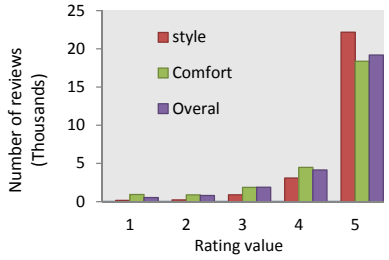


Figure 8.2: distribution of ratings in each category

Multicore featurizer

Using a simple map-reduce method. We have re-implemented the featurizer to take advantage of multicore processing in Matlab. The featurizer supports up to 4 cores and supports bigrams and unigrams. For fast lookup a hash table is used. Step 1: Scheduler randomly assigns a core number to each review Step 2: Mapper sends segments of the dataset to workers Step 3: Each worker featurizes the segment and sends back a hash table Step 4: Reducer puts the hash tables together

8.5 Efficient Code for Comment Featurization

In this work we use "Featurization" instead of "Tokenization" Tokenization is one of the efficient methods to convert textual data to numerical data that is usable in algorithms. In this work we have used Tokenization and Featurization interchangeably. Tokenization breaks strings into atomic entities like stems or words and these entities can then be associated with a numerical value. There are different ways to tokenize a string. In this text we talked about Bernoulli, Multinomial, and Term Frequency - Inverse Document Frequency (tf-idf). The tokenization code for this dissertation is available in Python[3] and Matlab[4]. For this work, I have developed a constant time $O(1)$ algorithm for comment tokenization.

8.6 Constant time $O(1)$ implementation of string featurizer

Consider for an example that we would like to featurize the qoute by Albert Einstein "Try not to become a person of success but a person of Value". Assuming that no word is removed and no word stemming is used, the final contents of the word frequency vector are shown in Table 8.3.

Table 8.3: Featurized vector for "Try not to become a person of success but a person of Value"

Word	Freq.
try	1
not	1
to	1
become	1
a	2
person	2
of	2
success	1
but	1
value	1

A naive and slow implementation of this algorithm will take a word like "try" and will search through all the words that exist in the feature vector thus requiring $O(n)$ comparisons in the worst case. Since older versions of Matlab lacked the required data structures for a

[3]This tokenizer is part of the PyNLP package that was developed for this dissertation https://github.com/faridani/PyNLP

[4]The tokenizer is part of the MatlabNLP package that was developed for this dissertation and can be found at https://github.com/faridani/MatlabNLP

more efficient algorithm, the naive implementation was used, thus taking about 8 hours when it was used to process Afghanistan War Logs[5]

One can realize that using a hashmap will increase the efficiency of this method. Looking up a value in a hash table is a constant time operation and it takes $O(1)$ to look up the word "try" in a large hash table. By using a hash table we were able to increase the efficiency of the featurizer, and the processing time for the War Log corpus decreased to seconds.

In the recent versions of Matlab a `containers.Map` can be used to store this data.

CCA for dimensionality reduction

Originally, PCA was used as the main dimensionality reduction algorithm for visualization in Opinion Space[Faridani et al., 2010]. Each participant in Opinion Space evaluates a subset of the textual responses of other participants using one slider to indicate level of agreement with a given textual response. The ideal projection seeks to maximize the global correlation between distance and agreement. We assume similar participants often agree and dissimilar participants often disagree. Therefore we record participant i's agreement value for j's response d_{ij} and the agreement value that participant i has given to j (φ_{ij}). Thus, the correlation coefficient between d and φ is used as a metric for comparing different dimensionality reduction techniques. A good dimensionality reduction technique will put participants that agree with each other closer together in the latent space and thus gives a larger global correlation coefficient between d and φ. Table 8.4 summarizes the improvements of the correlation coefficient over the baseline PCA on numerical ratings that was used in the original version of Opinion Space[Faridani et al., 2010]. CCA provides 140.8% improvement in the dataset that was collected from an Opinion State deployment for the car manufacturer. This improvement amount is consistent with the 161.2% improvement from the second dataset that was collected from the deployment on the US State Department website.

In Table 8.4 the best CCA result is found by finding the optimal λ through a line search. The optimal value in this case occurs at $\lambda = 0.4$ (the theoretical value from the Gaussian model in this case was $= 0.47$.

CCA for multi-aspect rating prediction

By assuming $U^{-1\top}X \approx z \approx V^{-1\top}Y$ we can devise a regression model for predicting multidimensional numerical ratings y from a textual review x:

$$y \leftarrow \quad V^{\top}U^{-1\top}X$$

Essentially, we are taking the generative model that we learned from paired data with CCA and using it to project the text comment x back to its underlying latent opinion z. Then,

[5]Afghan War Logs consists of 91,731 documents allegedly leaked to WikiLeaks by Bradley Manning. The corpus is available in a 77MB csv file and can be retrieved from the Internet Archive http://archive.org/details/WikileaksWarDiaryCsv.

Dimensionality Reduction Method	Improvement
PCA on Featurized Textual Responses	-15.4%
PCA on Numerical Ratings (Baseline)	0.0%
CCA (Individual uncertainty model)	85.3%
CCA on Both	140.8%
CCA (Gaussian Model)	140.8%
CCA (Optimal λ)	145.5%

Table 8.4: Improvement of the dimensionality reduction over a PCA baseline. Car Manufacturer Dataset

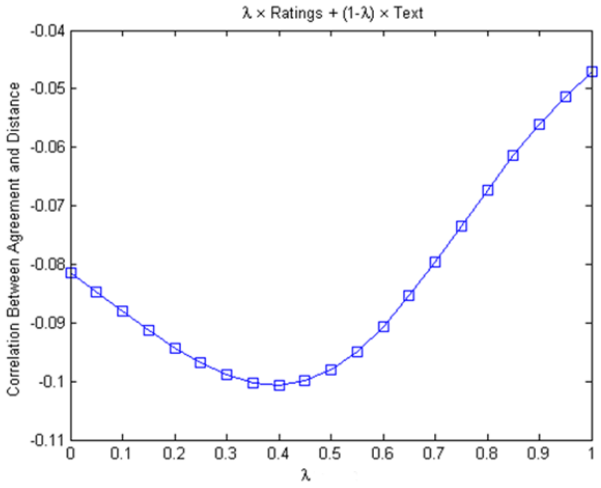


Figure 8.3: Evaluation of the convex combination of two projections. Combining the two projections improves the final visualization. This graph shows different values for the correlation coefficient after varying the weight (λ) in the convex combination of two projections $(1-\lambda)S_{x,wx} + (\lambda)S_{y,wy}$. As we move from right to left the weight for the text is increased. The optimal value in this case occurs at $\lambda = 0.4$ (the theoretical value from the Gaussian model in this case was $= 0.47$)

we take z and project it to its most likely numerical rating. In the end, this just corresponds to multiplying x by some matrix.[6] This means that CCA regression has the same set of parameters that multivariate regression does, but these parameters are chosen in a different way. Interestingly, CCA chooses regression parameters jointly, in a way that does not simply factor over the different output dimensions.

[6]For linear predictors, frequent appearance of positive words will result in values of y that are larger than the maximum allowed rating, similarly negative words may result in a value of y below the minimum possible rating. Thus, in our experiements if CCA regression or linear regression predicts a value of y outside the allowed range, y is adjusted back to the nearest endpoint.

Prediction Model	Featurization	MSE	Learning Time (s)
Naïve Bayes	Multinomial	2.26	9
Naïve Bayes	Bernoulli	2.13	9
SVM	Bernoulli	2.04	2,589
SVM	tf-idf	2.28	1,126
SVM	Multinomial	2.14	3,265
CCA Regression	Bernoulli	1.70	33
CCA Regression	tf-idf	1.69	33
CCA Regression	Multinomial	1.71	36
Linear Regression	Bernoulli	4.21	110
Linear Regression	tf-idf	1.47	113
Linear Regression	Multinomial	8.88	110
Random		15.76	
Random Samples from the Actual Distribution		4.40	
Naïve Majority Predictor		2.78	

Table 8.5: Results for 10-fold cross-validation for predicting the rating values from textual reviews on the Zappos dataset. The worst attainable error is 48.

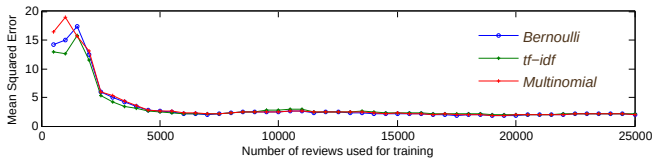


Figure 8.4: The MSE is plotted against the number of reviews used for training. The results suggest that the model is not significantly sensitive to the featurization method and reliable after at least 5,000 reviews are used in the training set.

We use the CCA regression model to predict the multi-aspect numerical rating values on the zappos dataset. Mean Square Error is used to evaluate. In order to evaluate our method, we also compare it with SVM and Naïve Bayes [Baccianella et al., 2010], which are widely used learning methods for ordinal regression problems. We used the SVM classifier from the LibSVM package with default settings. Experimental results in Table 8.5 show that the learning process in the CCA regression is significantly faster than SVM, and the prediction is more accurate than both SVM and Naïve Bayes. Compared to linear regression, CCA regression appears to be more robust to the type of featurization: it achieves comparably low errors using all of the different featurizations. Linear regression gives slightly lower error with the tf-idf featurization while producing significantly larger prediction errors with the other two types of featurization. We observed that CCA needed at least 5,000 textual reviews for training to provide a small error (Figure 8.4).

Applications of the model for recommender systems

Compared to PCA, CCA learns latent spaces where the distance metric corresponds more closely to the metric induced by agreement reported by users. The immediate application of this model is an "Opinion Recommender" for systems such as Opinion Space [Faridani et al., 2010]. If the users want to see the comments that she agrees with, the system can pick closest comments in the CCA space and generate the recommended list of comments. Similarly if the user is interested in reading comments that she may disagree with, the system can pick the farthest comments in the lower dimension space and generate the recommendations. Another application of this model in product recommendation is in finding users with similar tastes. Imagine the following scenario: one buyer may buy a shoe that is pricey and uncomfortable but is stylish. This user will rate this product negatively for "price" and "comfort" and positively on "style". At the end she expresses her overall positive satisfaction with the product in her comment despite the fact that she has rated the shoe negatively on two scales. CCA enables us to find another person who has expressed the same taste for this specific product and identify this new user as a potential anchor for the original buyer. Other products that the anchor has liked might be of the interest of the original buyer. In addition, the regression model that is derived from CCA can be used for sentiment analysis, filling in missing data and extracting multi-aspect ratings from text when the numerical ratings are not at hand.

8.7 Fair Error Measures for Skewed Data

Problem Description

Consider this scenario: An online retailer like Amazon continuously removes underperforming products from their marketplace. Products that receive low star ratings are removed frequently. As a result if we crawl Amazon, the distribution for the ratings in the training dataset will be very skewed. This raises this problem that a majority predictor that predicts 5 regardless of the input performs well on this dataset. Fig. 8.5 is the distribution for ratings on zappos.com that cause a naive majority predictor (all 5 predictor) to perform well on this dataset.

Type of response/output variables

This problem is solved for binary response variables by using precision and recall. But how about ordinal and continuous variables? In this chapter we look at error measured for skewed data in the following cases:

- Binary (solved: use precision, recall, accuracy measures)
- Ordinal (We solve it by using the inverse of frequency of each class)

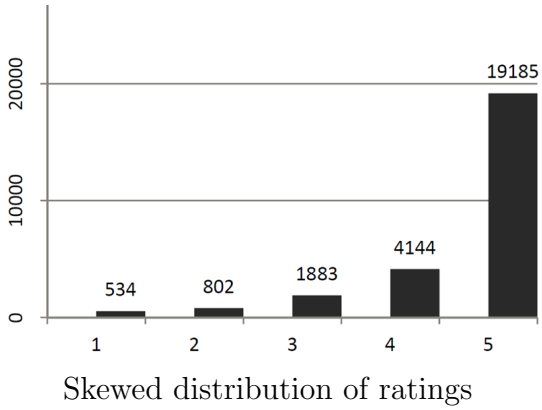


Figure 8.5: Skewed distribution of ratings

Table 8.6: The problem with the majority predictor

Actual (test set)	Majority predictor	Our predictor
1	4.5	1
5	4.5	4.4
5	4.5	4.4
5	4.5	4.4
5	4.5	4.4
5	4.5	4.4
5	4.5	4.4
.	.	.
.	.	.
.	.	.
5	4.5	4.4

- Continuous (We build an epsilon-range method)

Example

Lets construct an example that highlights this problem. Majority predictor predicts 4.5 for everything while our predictor predicts 1 correctly but predicts 4.4 for 5s (Table 8.6)

We define the Mean Squared Error in Eqn. 8.4. In Fig. 8.6 the Y axis shows the magnitude of MSE and X axis is the number of ratings that are used from Table 8.6. As we bring more fives, the error for the majority predictor goes down and eventually outperforms our predictor.

$$MSE = \frac{\sum_i^M (\tilde{\theta}_i - \theta_i)^2}{M} \tag{8.2}$$

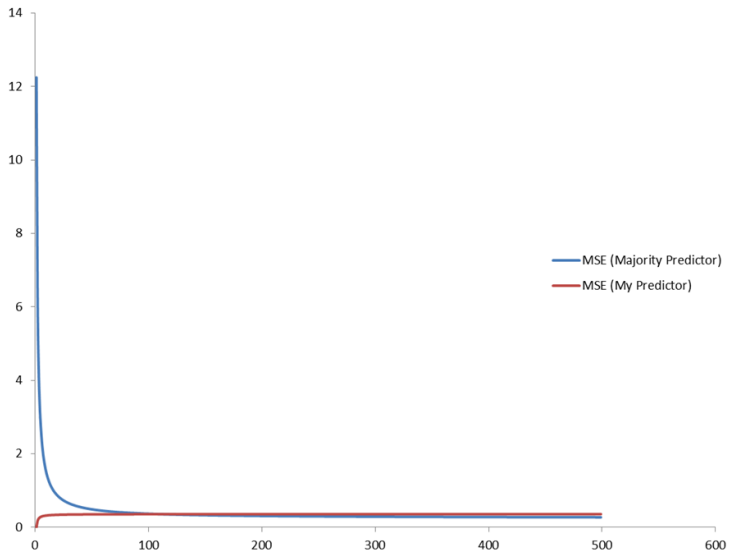


Figure 8.6: Unfair MSE

Δ-MSE: An Unbiased MSE For Ordinal Data

Multiply each residual by the inverse of frequency

$$\Delta MSE = \frac{\sum_i^M \frac{N}{|\tau_i|} (\tilde{\theta}_i - \theta_i)^2}{M} \tag{8.3}$$

Δ-MSE: For Continuous Data

For continuous data we can now multiply each residual by the inverse of the area under the pdf curve.

$$\Delta MSE_c = \frac{1}{M} \sum_i^M \frac{1}{\int_{\theta_i - \Delta}^{\theta_i + \Delta}} (\tilde{\theta}_i - \theta_i)^2 \tag{8.4}$$

Δ is a measure of how much emphasis we want to put on smaller values. Small Δ puts more emphasis and large Δ makes it equal to MSE.

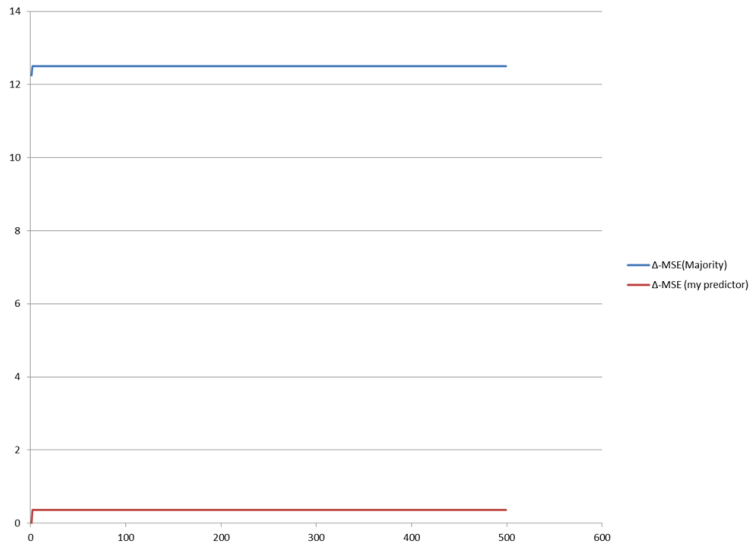


Figure 8.7: Δ MSE for fair analysis

8.8 Conclusion and Next Steps

Our experimental results suggest that CCA outperforms PCA as a dimensionality reduction tool when used for opinion visualization. Our second experiment suggests that CCA regression competes with state-of-the-art baselines when used for predicting multi-aspect ratings for products. These experiments demonstrate the potential of CCA as a useful technique in building recommendation systems. We were able to use very simple featurizations for raw text and still achieve large improvements. It is very likely that using more sophisticated language features such as parts-of-speech or grammatical dependencies will result in further gains.

Chapter 9

Conclusion and future work

Crowdsourcing is still a new field of research. This dissertation showcases a number of problems that can be solved by crowdsourcing. In CONE crowdsourcing is used to collect and classify a large number of bird photos. This allowed us to study bird migration patterns at Welder refuge. By comparing our data to the documents of bird presence at the same place in 30 years ago we were able to confirm the breeding population of 8 species that were not known to be in the area before. Crowdsourcing allowed us to perform this task accurately and in scale.

We later used crowdsourcing to show evaluate a model of human motor on a large number of participants with (potentially) a diverse age and geographic distribution. We compared the data that was collected from more than 3,600 online participants with the data from 46 participants in our controlled in-lab study and showed that they are both consistent.

We looked at Opinion Space, an interactive visualization system that is developed to help organizations scalably engage large audiences in generating ideas on topics of interest. It allows participants to visualize their relationship to other community members, express thoughtful ideas and suggestions, and consider and evaluate the most insightful ideas of other community members. Principal Component Analysis (PCA) has been used Opinion Space to visualize these numerical values in a 2D space. We have developed and evaluated a model based on Canonical Correlation Analysis that combines both the textual responses and numerical responses together to develop the 2D projection. The model is evaluated by using the voting data that is collected from participants. We assumed that better dimensionality reduction models should put the responses that are more desirable to the participant closer to her. Thus we look at the Pearson's correlation value between the distance between the response and the participant's location in the 2D space and the rating values that the participant has assigned to the response. Our preliminary analysis shows that when compared to PCA and a random scatter of points in a 2D space, CCA provides a %40 higher Pearson's correlation value. We also like to explore the relationship between how cascading incentive structures increase participation in Opinion Space.

Opinion Space inspired another work on online labor markets. We looked two questions. 1) What is the expected completion time for a crowdsourced task on an online labor market

2) What is the optimal pricing model for a task that is posted on a labor market like Amazon Mechanical Turk.

In order to seamlessly integrate a human computation component (e.g., Amazon Mechanical Turk) within a larger production system, we need to have some basic understanding of how long it takes to complete a task posted for completion in a crowdsourcing platform. We present an analysis of the completion time of tasks posted on Amazon Mechanical Turk. We model the completion time as a stochastic process and build a statistical method for predicting the expected time for task completion. We use a survival analysis model based on Cox proportional hazards regression. We present the preliminary results of our work, showing how time-independent variables of posted tasks (e.g., type of the task, price of the HIT, day posted, etc) affect completion time. We consider this a first step towards building a comprehensive optimization module that provides recommendations for pricing, posting time, in order to satisfy the constraints of the requester.

One of the most important challenges for task requesters on crowdsourcing markets like Amazon Mechanical Turk (AMT) is to properly price and schedule their tasks. Improper pricing or scheduling often results in task starvation and loss of capital on these markets. For example it is believed that workers have an expected hourly wage in mind and they tend to not accept underpriced tasks that need more time per unit reward than what they have in mind. Tasks that are not accepted stay in the system (they are often called starved HITs). Starved HITs may be canceled or reposted by the requester resulting in expenditure of more time and money than planned for the task. Overpriced tasks are also undesirable since requesters can invest excess capital in quality assurance for the data that they have collected. By using a survival analysis model we devise an algorithm for determining the optimal reward for a crowdsourced task. We have only focused on the most important attribute (reward) but our approach is generalizable and practitioners can use it to optimize the completion rate, bonus and even keywords of their tasks.

The generalized rating inference model (generalized sentiment analysis) is another major part of this work. We have collected a large amount of data from Zappos.com and used CCA to perform multidimensional rating inference and demonstrated how our CCA regression model can be used for this purpose.

A number of topics are left for future work and might be of interest to young PhD students:

- The question of judge quality in an open crowdsourcing market is an interesting research topic that was out of the scope of this dissertation. I made some effort to use CCA for flagging low quality judges in Opinion Space. We did not study this a general purpose crowdsourcing environment and that topic by itself deserves attention.

- In this dissertation we manually cluster keywords by topics. I studied and experimented with automatic topic models that were available at the time (e.g., LDA and pLSI) but in all cases the manual clustering significantly outperformed the automated clustering on our correlation metric. Developing a topic clustering algorithm that can perform close to the manual clustering for Opinion Space is the natural next step for this work.

- In the last chapter we built a model for multivariate sentiment analysis and prototyped a working version in Matlab that could work on offline datasets. Building such a large scale system to perform in real-time is an interesting research challenge. Here we built a distributed featurizer that could run on multi-core and multi-processor platforms but the CCA regression was not distributed. Any online implementation of this model requires a distributed CCA solver and this deserves further studies.

- Modeling a crowdsourcing markets based on Nash equilibrium was another topic that was out of the scope of this dissertation. Nevertheless, it is a natural extension of our work and is worthy of more work.

I personally hope that the material presented in this dissertation can be used by both researchers and practitioners in the area of crowdsourcing.

Bibliography

[Abraham and Merola, 2005] Abraham, B. and Merola, G. (2005). Dimensionality reduction approach to multivariate prediction. *Computational statistics & data analysis*, 48(1):5–16.

[Andersen and Gill, 1982] Andersen, P. and Gill, R. (1982). Cox's regression model for counting processes: a large sample study. *The annals of statistics*, 10(4):1100–1120.

[Andreasen et al., 2007] Andreasen, M., Nielsen, H., Schrøder, S., and Stage, J. (2007). What happened to remote usability testing?: an empirical study of three methods. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1405–1414. ACM.

[Antin and Churchill, 2011] Antin, J. and Churchill, E. (2011). Badges in social media: A social psychological perspective. *Human Factors*, pages 1–4.

[Arora et al., 2011] Arora, A., Cunningham, A., Chawdhary, G., Vicini, C., Weinstein, G., Darzi, A., and Tolley, N. (2011). Clinical applications of telerobotic ent-head and neck surgery. *International Journal of Surgery*.

[Baccianella et al., 2010] Baccianella, S., Esuli, A., and Sebastiani, F. (2010). Feature selection for ordinal regression. In *SAC*, pages 1748–1754.

[Bach and Jordan, 2005] Bach, F. and Jordan, M. (2005). A probabilistic interpretation of canonical correlation analysis. *Dept. Statist., Univ. California, Berkeley, CA, Tech. Rep*, 688.

[Bakshy et al., 2012] Bakshy, E., Rosenn, I., Marlow, C., and Adamic, L. (2012). The role of social networks in information diffusion. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 519–528, New York, NY, USA. ACM.

[Barabasi, 2005] Barabasi, A. (2005). The origin of bursts and heavy tails in human dynamics. *Nature*, 435(7039):207–211.

[Belkin and Niyogi, 2003] Belkin, M. and Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396.

[Benkoski et al., 1991] Benkoski, S., Monticino, M., and Weisinger, J. (1991). A survey of the search theory literature. *Naval Research Logistics*, 38(4):469–494.

[Berinsky, 1999] Berinsky, A. (1999). The two faces of public opinion. *American Journal of Political Science*, pages 1209–1230.

[Bernstein et al., 2010] Bernstein, M., Little, G., Miller, R., Hartmann, B., Ackerman, M., Karger, D., Crowell, D., and Panovich, K. (2010). Soylent: A Word Processor with a Crowd Inside.

[Berry et al., 1995] Berry, S., Levinsohn, J., and Pakes, A. (1995). Automobile prices in market equilibrium. *Econometrica*, 63(4):841–890.

[Bigham et al., 2010a] Bigham, J., Jayant, C., Ji, H., Little, G., Miller, A., Miller, R., Tatarowicz, A., White, B., White, S., and Yeh, T. (2010a). VizWiz: nearly real-time answers to visual questions. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A)*, pages 1–2. ACM.

[Bigham et al., 2010b] Bigham, J. P., Jayant, C., Ji, H., Little, G., Miller, A., Miller, R. C., Miller, R., Tatarowicz, A., White, B., White, S., and Yeh, T. (2010b). Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, UIST '10, pages 333–342, New York, NY, USA. ACM.

[Bishop et al., 1998] Bishop, C., Svensén, M., and Williams, C. (1998). Gtm: The generative topographic mapping. *Neural computation*, 10(1):215–234.

[Bishop, 2007] Bishop, J. (2007). Increasing participation in online communities: A framework for human-computer interaction. *Computers in Human Behavior*, 23(4):1881–1893.

[Bitton, 2009] Bitton, E. (2009). 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.

[Bitton, 2012] Bitton, E. (2012). *Geometric Models for Collaborative Search and Filtering*. PhD thesis, UNIVERSITY OF CALIFORNIA, BERKELEY.

[Bitton and Goldberg, 2008] Bitton, E. and Goldberg, K. (2008). Hydra: A Framework and Algorithms for Mixed-Initiative UAV-Assisted Search and Rescue. In *IEEE International Conference on Automation Science and Engineering, 2008. CASE 2008*, pages 61–66.

[Blacklock, 1984] Blacklock, G. (1984). Checklist of Birds of the Welder Wildlife Refuge. In *Welder Wildlife Foundation, Sinton, Texas.*

[Blei et al., 2003a] Blei, D., Ng, A., and Jordan, M. (2003a). Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022.

[Blei et al., 2003b] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003b). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.

[Blitzer et al.,] Blitzer, J., Foster, D., and Kakade, S. Domain adaptation with coupled subspaces.

[Brandtzæg and Heim, 2009] Brandtzæg, P. and Heim, J. (2009). Explaining participation in online communities. *Handbook of Research on Socio-Technical Design and Social Networking Systems*.

[Candes et al., 2009] Candes, E., Li, X., Ma, Y., and Wright, J. (2009). Robust principal component analysis? *Arxiv preprint ArXiv:0912.3599*.

[Carrington et al., 2005] Carrington, P., Scott, J., and Wasserman, S. (2005). *Models and methods in social network analysis*. Cambridge Univ Pr.

[Carver, 2006] Carver, E. (2006). Birding in the united states: A demographic and economic analysis. *Addendum to the*, pages 2006–4.

[Cayton, 2005] Cayton, L. (2005). Algorithms for manifold learning. *University of California, San Diego, Tech. Rep. CS2008-0923*.

[Chaudhuri et al., 2009] Chaudhuri, K., Kakade, S., Livescu, K., and Sridharan, K. (2009). Multi-view clustering via canonical correlation analysis. In *Proceedings of the 26th annual international conference on machine learning*, pages 129–136. ACM.

[Chen et al., 2001] Chen, S., Donoho, D., and Saunders, M. (2001). Atomic decomposition by basis pursuit. *SIAM review*, pages 129–159.

[Chen et al., 2003] Chen, Y., Kao, T., and Sheu, J. (2003). A mobile learning system for scaffolding bird watching learning. *Journal of Computer Assisted Learning*, 19:347–359.

[Conover and Iman, 1981] Conover, W. and Iman, R. (1981). Rank transformations as a bridge between parametric and nonparametric statistics. *American statistician*, pages 124–129.

[Corredor and Sofrony, 2011] Corredor, J. and Sofrony, J. (2011). Shared control based on roles for telerobotic systems. In *Robotics Symposium, 2011 IEEE IX Latin American and IEEE Colombian Conference on Automatic Control and Industry Applications (LARC)*, pages 1–6. IEEE.

[Dahl, 2007a] Dahl, A. (2007a). Implementation of a Collaborative Observatory for Natural Environments.

[Dahl, 2007b] Dahl, A. B. C. (2007b). Implementation of a collaborative observatory for natural environments. Technical Report UCB/EECS-2007-71, EECS Department, University of California, Berkeley.

[Dalgaard, 2008] Dalgaard, P. (2008). *Introductory statistics with R*. Springer Verlag.

[Deterding et al., 2011a] Deterding, S., Dixon, D., Khaled, R., and Nacke, L. (2011a). From game design elements to gamefulness: defining gamification. In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, pages 9–15. ACM.

[Deterding et al., 2011b] Deterding, S., Sicart, M., Nacke, L., O'Hara, K., and Dixon, D. (2011b). Gamification. using game-design elements in non-gaming contexts. In *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems*, pages 2425–2428. ACM.

[Eckart and Young, 1936] Eckart, C. and Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218.

[Elgammal et al., 2000] Elgammal, A., Harwood, D., and Davis, L. (2000). Non-parametric model for background subtraction. *Lecture Notes in Computer Science*, 1843:751–767.

[Erdmann and Mason, 1988] Erdmann, M. and Mason, M. (1988). An exploration of sensorless manipulation. *IEEE Journal of Robotics and Automation*, 4(4):369–379.

[Erickson et al., 2012] Erickson, D., Lacheray, H., Lambert, J., Mantegh, I., Crymble, D., Daly, J., and Zhao, Y. (2012). Haptics-based immersive telerobotic system for improvised explosive device disposal: Are two hands better than one? In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 8387, page 38.

[Faridani et al., 2010] Faridani, S., Bitton, E., Ryokai, K., and Goldberg, K. (2010). 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.

[Faridani et al., 2009] Faridani, S., Lee, B., Glasscock, S., Rappole, J., Song, D., and Goldberg, K. (2009). A networked telerobotic observatory for collaborative remote observation of avian activity and range change. In *Proceedings of the 2009 IFAC Workshop on Networked Robotics (Moore KL, Ed.). International Federation of Automatic Control. Elsevier, Oxford, United Kingdom*. Citeseer.

[Fine et al., 2003] Fine, J., Glidden, D., and Lee, K. (2003). A simple estimator for a shared frailty regression model. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, 65(1):317–329.

[Fishkin and Luskin, 2005] Fishkin, J. and Luskin, R. (2005). Experimenting with a democratic ideal: Deliberative polling and public opinion. *Acta Politica*, 40(3):284–298.

[Fitts, 1954] Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47:381–391.

[Fodor, 2002] Fodor, I. (2002). A survey of dimension reduction techniques. *Livermore, CA: US DOE Office of Scientific and Technical Information*, 18.

[Freeman, 2000] Freeman, L. (2000). Visualizing social networks. *Journal of social structure*, 1(1):4.

[Geyer, 2008] Geyer, C. (2008). Active target search from UAVs in urban environments. In *IEEE International Conference on Robotics and Automation, 2008. ICRA 2008*, pages 2366–2371.

[Glasscock and Blankenship, 2007] Glasscock, S. and Blankenship, T. (2007). Checklist of the Birds of the Welder Wildlife Refuge. In *San Patricio County, Texas. Rob & Bessie Welder Wildlife Foundation, Sinton, Texas.*

[Goebel and Gruenwald, 1999] Goebel, M. and Gruenwald, L. (1999). A survey of data mining and knowledge discovery software tools. *ACM SIGKDD Explorations Newsletter*, 1(1):20–33.

[Goldberg et al., 2012] Goldberg, K., Faridani, S., and Alterovitz, R. (2012). A New Derivation and Dataset for Fitts Law of Human Motion. `http://tele-actor.net/fitts-dataset/fitts-paper.pdf/`. [Online; accessed 10-Dec-2012].

[Goldberg et al., 1995] Goldberg, K., Mascha, M., Gentner, S., Rothenberg, N., Sutter, C., and Wiegley, J. (1995). Desktop teleoperation via the world wide web. In *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, volume 1, pages 654–659. IEEE.

[Goldberg and Siegwart, 2002] Goldberg, K. and Siegwart, R. (2002). *Beyond webcams: an introduction to online robots.* The MIT Press.

[Goldberg et al., 2003] Goldberg, K., Song, D., and Levandowski, A. (2003). Collaborative teleoperation using networked spatial dynamic voting. *Proceedings of the IEEE*, 91(3):430–439.

[Guiard et al., 2001] Guiard, Y., Bourgeois, F., Mottet, D., and Beaudouin-Lafon, M. (2001). Beyond the 10-bit barrier: Fitts' law in multi-scale electronic worlds. *Proc. Interaction Homme-Machine / Human-Computer Interaction (IHM-HCI 2001), People and Computers XV - Interactions without frontiers*, pages 573–588.

[Gunduz and Ozsu, 2003] Gunduz, S. and Ozsu, M. (2003). A poisson model for user accesses to web pages. *Lecture Notes in Computer Science*, pages 332–339.

[Haidegger et al., 2011] Haidegger, T., Sndor, J., and Beny, Z. (2011). Surgery in space: the future of robotic telesurgery. *Surgical Endoscopy*, 25:681–690. 10.1007/s00464-010-1243-3.

[Ham et al., 2003] Ham, J., Lee, D., and Saul, L. (2003). Learning high dimensional correspondences from low dimensional manifolds.

[Hardoon et al., 2004] Hardoon, D., Szedmak, S., and Shawe-Taylor, J. (2004). Canonical correlation analysis: an overview with application to learning methods. *Neural Computation*, 16(12):2639–2664.

[Heer and Boyd, 2005] Heer, J. and Boyd, D. (2005). Vizster: Visualizing online social networks. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pages 32–39. IEEE.

[Hoffmann et al., 1997] Hoffmann, E., Chang, W., and Yim, K. (1997). Computer mouse operation: is the left-handed user disadvantaged? *Applied Ergonomics*, 28(4):245–248.

[Hoffmann and Hui, 2010] Hoffmann, E. and Hui, M. (2010). Movement times of different arm components. *Ergonomics*, 53(8):979–993.

[Hotelling, 1936] Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*, 28(3/4):321–377.

[Howe, 2006] Howe, J. (2006). The rise of crowdsourcing. *Wired magazine*, 14(6):1–4.

[Ipeirotis, 2010a] Ipeirotis, P. G. (2010a). Analyzing the amazon mechanical turk marketplace. *XRDS*, 17:16–21.

[Ipeirotis, 2010b] Ipeirotis, P. G. (2010b). Analyzing the Amazon Mechanical Turk marketplace. *XRDS*, 17:16–21.

[Izenman, 2008] Izenman, A. (2008). *Modern multivariate statistical techniques: regression, classification, and manifold learning*. Springer Verlag.

[Jagacinski and Flach, 2003] Jagacinski, R. J. and Flach, J. M. (2003). *Control Theory for Humans: Quantitative Approaches to Modeling Performance*. Lawrence Erlbum Associates, Mahwah, New Jersey, 1st edition.

[Johnson and Wichern, 2002] Johnson, R. and Wichern, D. (2002). *Applied multivariate statistical analysis*, volume 4. Prentice hall Upper Saddle River, NJ.

[Jolliffe and MyiLibrary, 2002] Jolliffe, I. and MyiLibrary (2002). *Principal component analysis*, volume 2. Wiley Online Library.

[Karmann and von Brandt, 1990] Karmann, K. and von Brandt, A. (1990). Moving object recognition using an adaptive background memory. *Time-varying image processing and moving object recognition*, 2:289–296.

[Keller, 2010] Keller, C. (2010). Applying optimal search theory to inland sar: Steve fossett case study. In *Information Fusion (FUSION), 2010 13th Conference on*, pages 1–8. IEEE.

[Kim et al., 2002] Kim, J., Choi, B., Park, S., Kim, K., and Ko, S. (2002). Remote control system using real-time mpeg-4 streaming technology for mobile robot. In *Consumer Electronics, 2002. ICCE. 2002 Digest of Technical Papers. International Conference on*, pages 200–201. IEEE.

[Kimber et al., 2002] Kimber, D., Liu, Q., Foote, J., and Wilcox, L. (2002). Capturing and presenting shared multi-resolution video. In *SPIE ITCOM 2002. Proceeding of SPIE, Boston*, volume 4862, pages 261–271. Citeseer.

[Kittur et al., 2008] Kittur, A., Chi, E., and Suh, B. (2008). Crowdsourcing user studies with Mechanical Turk. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 453–456. ACM.

[Kittur et al., 2009] Kittur, A., Chi, E., and Suh, B. (2009). What's in wikipedia?: mapping topics and conflict using socially annotated category structure. In *Proceedings of the 27th international conference on Human factors in computing systems*, pages 1509–1512. ACM.

[Kleinbaum and Klein, 2005] Kleinbaum, D. and Klein, M. (2005). *Survival analysis: a self-learning text*. Springer Verlag.

[Kochhar et al., 2010] Kochhar, S., Mazzocchi, S., and Paritosh, P. (2010). The anatomy of a large-scale human computation engine. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP '10, pages 10–17.

[Kohonen, 1988] Kohonen, T. (1988). Self-organization and associative memory. *Self-Organization and Associative Memory, 100 figs. XV, 312 pages.. Springer-Verlag Berlin Heidelberg New York. Also Springer Series in Information Sciences, volume 8*, 1.

[Koopman, 1979] Koopman, B. (1979). Search and its optimization. *American Mathematical Monthly*, pages 527–540.

[Kriplean et al., 2012] Kriplean, T., Morgan, J., Freelon, D., Borning, A., and Bennett, L. (2012). Supporting reflective public thought with considerit. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 265–274. ACM.

[Kvålseth, 1980] Kvålseth, T. (1980). An alternative to Fitts' law. *Bulletin of the psychonomic Society*, 16:371–373.

[Lai and Fyfe, 2006] Lai, P. and Fyfe, C. (2006). A latent variable implementation of canonical correlation analysis for data visualisation. In *Neural Networks, 2006. IJCNN'06. International Joint Conference on*, pages 1143–1149. IEEE.

[Larson, 2008] Larson, M. (2008). Analysis of variance. *Circulation*, 117(1):115–121.

[Lee, 2008] Lee, B. (2008). Interface design and implementation of a collaborative observatory for natural environments. Master's thesis, EECS Department, University of California, Berkeley.

[Li et al., 2011a] Li, B., Ghose, A., and Ipeirotis, P. G. (2011a). Towards a theory model for product search. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 327–336, New York, NY, USA. ACM.

[Li et al., 2011b] Li, C., Huang, R., Ding, Z., Gatenby, C., Metaxas, D. N., and Gore, J. C. (2011b). A level set method for image segmentation in the presence of intensity inhomogeneities with application to mri. *IEEE Trans. Image Process.*, 20(7):2007–2016.

[Liu et al., 2005] Liu, B., Hu, M., and Cheng, J. (2005). Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of the 14th international conference on World Wide Web*, pages 342–351. ACM.

[Liu et al., 2002] Liu, Q., Kimber, D., Wilcox, L., Cooper, M., Foote, J., and Boreczky, J. (2002). Managing a camera system to serve different video requests. In *Multimedia and Expo, 2002. ICME'02. Proceedings. 2002 IEEE International Conference on*, volume 2, pages 13–16. IEEE.

[Ludford et al., 2004] Ludford, P., Cosley, D., Frankowski, D., and Terveen, L. (2004). Think different: increasing online community participation using uniqueness and group dissimilarity. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 631–638. ACM.

[LUSKIN et al., 2002] LUSKIN, R. C., FISHKIN, J. S., and JOWELL, R. (2002). Considered opinions: Deliberative polling in britain. *British Journal of Political Science*, 32(03):455–487.

[MacKenzie, 1992] MacKenzie, I. S. (1992). Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction*, 7:91–139.

[MacKenzie and Buxton, 1992] MacKenzie, I. S. and Buxton, W. (1992). Extending Fitts' law to two-dimensional tasks. In *Proc. ACM CHI '92*, pages 219–226.

[Macki and Strauss, 1982] Macki, J. and Strauss, A. (1982). *Introduction to optimal control theory*. Springer.

[Manning et al., 2008] Manning, C., Raghavan, P., and Schutze, H. (2008). *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge.

[Manning et al., 1999] Manning, C., Sch
"utze, H., and MITCogNet (1999). *Foundations of statistical natural language processing*. MIT Press.

[Marín et al., 2005] Marín, R., Sanz, P., Nebot, P., and Wirz, R. (2005). A multimodal interface to control a robot arm via the web: A case study on remote programming. *Industrial Electronics, IEEE Transactions on*, 52(6):1506–1520.

[Mason, 1985] Mason, M. (1985). The mechanics of manipulation. In *1985 IEEE International Conference on Robotics and Automation. Proceedings*, volume 2.

[Mason and Watts, 2010] Mason, W. and Watts, D. (2010). Financial incentives and the performance of crowds. *ACM SIGKDD Explorations Newsletter*, 11(2):100–108.

[McFadden, 1972] McFadden, D. (1972). *Conditional logic analysis of qualitative choice behavior*. Institute of Urban & Regional Development, University of California.

[McKillup, 2006] McKillup, S. (2006). *Statistics explained: an introductory guide for life scientists*. Cambridge Univ Pr.

[Meyer et al., 1988] Meyer, D., Abrams, R., Kornblum, S., Wright, C., and Keith Smith, J. (1988). Optimality in human motor performance: Ideal control of rapid aimed movements. *Psychological Review*, 95(3):340.

[Munson and Resnick, 2010] Munson, S. and Resnick, P. (2010). 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.

[Nadimpalli et al., 2006] Nadimpalli, U., Price, R., Hall, S., and Bomma, P. (2006). A comparison of image processing techniques for bird recognition. *Biotechnology progress*, 22(1).

[Nelder and Wedderburn, 1972] Nelder, J. and Wedderburn, R. (1972). Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, pages 370–384.

[Oberholser, 1974] Oberholser, H. (1974). The bird life of texas. *Univ. Tex. Press, Austin*.

[Pang and Lee, 2008] Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135.

[Paolacci et al., 2010] Paolacci, G., Chandler, J., and Ipeirotis, P. (2010). Running experiments on amazon mechanical turk. *Judgment and Decision Making*, 5(5):411–419.

[Papadimitriou and Tsitsiklis, 1987] Papadimitriou, C. and Tsitsiklis, J. (1987). The complexity of Markov decision processes. *Mathematics of operations research*, pages 441–450.

[Pariser, 2011] Pariser, E. (2011). *The filter bubble: What the Internet is hiding from you*. Penguin Pr.

[Perer and Shneiderman, 2006] Perer, A. and Shneiderman, B. (2006). Balancing systematic and flexible exploration of social networks. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):693–700.

[Pineau et al.,] Pineau, J., Gordon, G., and Thrun, S. Point-based value iteration: An anytime algorithm for POMDPs.

[Plamondon and Alimi, 1997] Plamondon, R. and Alimi, A. M. (1997). Speed/accuracy trade-offs in target-directed movements. *Behavioral and Brain Sciences*, 20:279–349.

[Qu et al., 2010] Qu, L., Ifrim, G., and Weikum, G. (2010). The bag-of-opinions method for review rating prediction from sparse text patterns. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 913–921. Association for Computational Linguistics.

[Rai and Daumé III, 2009] Rai, P. and Daumé III, H. (2009). Multi-label prediction via sparse infinite cca. *Advances in Neural Information Processing Systems*, 22:1518–1526.

[Rappole and Blacklock, 1985] Rappole, J. and Blacklock, G. (1985). *Birds of the Texas Coastal Bend: abundance and distribution*, volume 84. Texas A&M University Press, College Station.

[Rappole et al., 2007] Rappole, J., Blacklock, G., and Norwine, J. (2007). Apparent rapid range change in south texas birds: response to climate change. *The changing climate of south Texas 1900*, 2100:133–146.

[Rappole and Faridani., 2011] Rappole, J. H., S. G. K. G. D. S. and Faridani., S. (2011). Range change among new world tropical and subtropical birds. *In Tropical vertebrates in a changing world (K.-L. Schuchmann, ed.), Bonner Zoologische Monographen, Bonn, Germany.*, 57:151–167.

[Resnick and Varian, 1997] Resnick, P. and Varian, H. R. (1997). Recommender systems. *Commun. ACM*, 40(3):56–58.

[Rioulo and Guiard, 2012] Rioulo, l. and Guiard, Y. (2012). Power vs. logarithmic model of fitts' law: A mathematical analysis. *"Math. Sci. hum. / Mathematics and Social Sciences"*. To Appear.

[Rosencrantz et al., 2003] Rosencrantz, M., Gordon, G., and Thrun, S. (2003). Locating moving entities in indoor environments with teams of mobile robots. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 233–240. ACM New York, NY, USA.

[Rourke and Kanuka, 2007] Rourke, L. and Kanuka, H. (2007). Barriers to online critical discourse. *International Journal of Computer-Supported Collaborative Learning*, 2(1):105–126.

[Sack, 2000] Sack, W. (2000). Conversation map: An interface for very large-scale conversations. *Journal of Management Information Systems*, 17(3):73–92.

[Sack, 2005] Sack, W. (2005). Discourse architecture and very large-scale conversation. *Digital Formations: IT and New Architectures in the Global Realm*, pages 242–282.

[Schiff et al., 2009] Schiff, J., Meingast, M., Mulligan, D., Sastry, S., and Goldberg, K. (2009). Respectful cameras: Detecting visual markers in real-time to address privacy concerns. *Protecting Privacy in Video Surveillance*, pages 65–89.

[Schmidt, 2007] Schmidt, T. (2007). Design, algorithms, and architecture of an improved collaborative observatory for natural environments. Master's thesis, UC Berkeley.

[Sebrechts et al., 1999] Sebrechts, M. M., Cugini, J. V., Laskowski, S. J., Vasilakis, J., and Miller, M. S. (1999). Visualization of search results: a comparative evaluation of text, 2d, and 3d interfaces. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 3–10, New York, NY, USA. ACM.

[Shannon, 2001] Shannon, C. (2001). A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55.

[Shneiderman, 1993] Shneiderman, B. (1993). 1.1 direct manipulation: astep beyond programming languages. *Sparks of Innovation in Human-Computer Interaction*.

[Shneiderman, 1997] Shneiderman, B. (1997). Direct manipulation for comprehensible, predictable and controllable user interfaces. In *Proceedings of the 2nd international conference on Intelligent user interfaces*, pages 33–39. ACM.

[Silva and Tenenbaum, 2003] Silva, V. and Tenenbaum, J. (2003). Global versus local methods in nonlinear dimensionality reduction. *Advances in neural information processing systems*, 15:705–712.

[Song, 2009a] Song, D. (2009a). *Sharing a vision: systems and algorithms for collaboratively-teleoperated robotic cameras*, volume 51. Springer Verlag.

[Song, 2009b] Song, D. (2009b). *Sharing a Vision: Systems and Algorithms for Collaboratively-Teleoperated Robotic Cameras (Springer Tracts in Advanced Robotics)*. Springer.

[Song et al., 2003] Song, D., Pashkevich, A., and Goldberg, K. (2003). Sharecam part II: Approximate and distributed algorithms for a collaboratively controlled robotic webcam. In *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings*, volume 2.

[Song et al., 2008a] Song, D., Qin, N., and Goldberg, K. (2008a). Systems, control models, and codec for collaborative observation of remote environments with an autonomous networked robotic camera. *Autonomous Robots*, 24(4):435–449.

[Song et al., 2008b] Song, D., Qin, N., Xu, Y., Kim, C., Luneau, D., and Goldberg, K. (2008b). System and algorithms for an autonomous observatory assisting the search for

the ivory-billed woodpecker. In *Automation Science and Engineering, 2008. CASE 2008. IEEE International Conference on*, pages 200–205. IEEE.

[Song et al., 2006] Song, D., van der Stappen, A., and Goldberg, K. (2006). Exact algorithms for single frame selection on multiaxis satellites. *Automation Science and Engineering, IEEE Transactions on*, 3(1):16–28.

[Stauffer and Grimson, 1999] Stauffer, C. and Grimson, W. (1999). Adaptive background mixture models for realtime tracking, IEEE Comput Soc Conf Comput Vis Pattern Recogn (Proceedings CVPR99), Ft. *Collins, CO, USA*.

[Stone, 1978] Stone, L. (1978). Theory of optimal search. *Bull. Amer. Math. Soc. 84 (1978), 649-652. DOI: 10.1090/S0002-9904-1978-14513-3 PII: S*, 2(9904):14513–3.

[Sun and Chen, 2007] Sun, T. and Chen, S. (2007). Locality preserving cca with applications to data visualization and pose estimation. *Image and Vision Computing*, 25(5):531–543.

[Sunstein, 2007] Sunstein, C. (2007). *Republic. com 2.0*. Princeton Univ Pr.

[Sunstein, 2001] Sunstein, C. R. (2001). *Republic.com*. Princeton University Press, Princeton, NJ, USA.

[Surowiecki, 2005] Surowiecki, J. (2005). *The wisdom of crowds*. Anchor.

[Tenenbaum et al., 2000] Tenenbaum, J., De Silva, V., and Langford, J. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323.

[Thomas, 2002] Thomas, M. (2002). Learning within incoherent structures: The space of online discussion forums. *Journal of Computer Assisted Learning*, 18(3):351–366.

[Thompson, 1999] Thompson, P. (1999). Method for open loop camera control using a motion model to control camera movement. US Patent 5,872,594.

[Thrun et al., 2005] Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents series)*. Intelligent robotics and autonomous agents. The MIT Press.

[Timothy H. Chung, Moshe Kress, and Johannes O. Royset, 2009] Timothy H. Chung, Moshe Kress, and Johannes O. Royset (2009). Probabilistic Search Optimization and Mission Assignment for Heterogeneous Autonomous Agents. In *Int'l. Conference on Robotics and Automation, To appear*.

[Train, 2003] Train, K. (2003). *Discrete choice methods with simulation*. Cambridge Univ Pr.

[van der Maaten and Hinton, 2008] van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85.

[van der Maaten and Hinton, 2011] van der Maaten, L. and Hinton, G. (2011). Visualizing non-metric similarities in multiple maps. *Machine learning*, pages 1–23.

[Viégas and Donath, 2004] Viégas, F. and Donath, J. (2004). Social network visualization: Can we go beyond the graph. In *Workshop on Social Networks, CSCW*, volume 4, pages 6–10.

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

[Von Ahn et al., 2006] Von Ahn, L., Liu, R., and Blum, M. (2006). Peekaboom: a game for locating objects in images. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 55–64. ACM New York, NY, USA.

[Vulcano et al., 2010] Vulcano, G., van Ryzin, G., and Chaar, W. (2010). OM Practice—Choice-Based Revenue Management: An Empirical Study of Estimation and Optimization. *Manufacturing & Service Operations Management*, 12(3):371–392.

[Vulcano et al., 2008] Vulcano, G., van Ryzin, G., and Ratliff, R. (2008). Estimating primary demand for substitutable products from sales transaction data. Technical report, Working paper.

[Wang et al., 2011a] Wang, C., Krafft, P., and Mahadevan, S. (2011a). Manifold alignment.

[Wang et al., 2010] Wang, H., Lu, Y., and Zhai, C. (2010). Latent aspect rating analysis on review text data: a rating regression approach. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 783–792. ACM.

[Wang et al., 2011b] Wang, J., Faridani, S., and Ipeirotis, P. (2011b). Estimating the Completion Time of Crowdsourced Tasks Using Survival Analysis Models. *Crowdsourcing for Search and Data Mining (CSDM 2011)*, page 31.

[Washburn, 1981] Washburn, A. (1981). Search and detection. *Operations Research Society of America*.

[Weber et al., 2008] Weber, I., Robertson, S., and Vojnovic, M. (2008). Rethinking the esp game. In *Proc. of 27th intl. conf. on Human factors in Computing Systems, ser. CHI*, volume 9, pages 3937–3942.

[Willett et al., 2011] Willett, W., Heer, J., Hellerstein, J. M., and Agrawala, M. (2011). Commentspace: structured support for collaborative visual analysis. In *CHI*, pages 3131–3140.

[Wong et al., 2011] Wong, D., Faridani, S., Bitton, E., Hartmann, B., and Goldberg, K. (2011). The diversity donut: enabling participant control over the diversity of recommended responses. In *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems*, pages 1471–1476. ACM.

[Wren et al., 1997] Wren, C., Azarbayejani, A., Darrell, T., and Pentland, A. (1997). Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785.

[Xu, 2012] Xu, Y. (2012). *Systems and algorithms for automated collaborative observation using networked robotic cameras*. PhD thesis, TEXAS A&M UNIVERSITY.

[Yu et al.,] Yu, C., Chuang, J., Computing, S., Math, C., Gerkey, B., Gordon, G., and Ng, A. Open-loop plans in multi-robot POMDPs. Technical report, Technical Report, Stanford CS Department, 2005.