

# Statistical Analysis and Design of Crowdsourcing Applications

Adam Kapelner

A DISSERTATION

in

Statistics

For the Graduate Group in  
Managerial Science and Applied Economics

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the  
Degree of Doctor of Philosophy

2014

## Co-Supervisor of Dissertation

Signature \_\_\_\_\_

Name: Abba Krieger

Title: Robert Steinberg Professor of  
Statistics and Operations Research

## Co-Supervisor of Dissertation

Signature \_\_\_\_\_

Name: Ed George

Title: Universal Furniture Professor  
Professor of Statistics

## Graduate Group Chairperson

Signature \_\_\_\_\_

Name: Eric Bradlow

## Dissertation Committee

Abba Krieger, Professor

Ed George, Professor

Larry Brown, Professor

Dean Foster, Professor

Statistical Analysis and Design of Crowdsourcing Applications

Copyright © 2014

Adam Kapelner

## Acknowledgments

I would like to first and foremost thank my parents and sisters. Without their constant support and love, none of this would have been possible. I thank my close friends for being there for me. I thank my graduate student colleagues most notably Justin Bleich and Alex Goldstein, not only for coauthoring many of these thesis chapters, but for all the fun we've had over the years. I also thank my advisors Abba Krieger, Ed Geoge and Dean Foster for their support and guidance. I also want to thank Larry Brown and Paul Rosenbaum who have been a tremendous help over the years on different stages of my thesis. I would also like to thank Andrea Troxel and Kate Probert from the UPenn Biostatistics department for their collaboration and mentoring. I also acknowledge the National Science Foundation for support from their Graduate Research Fellowship Program.

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Motivation in Crowdsourcing Markets</b>	<b>5</b>
2.1	Introduction . . . . .	6
2.2	Mechanical Turk and its potential for field experimentation . . . . .	9
2.3	Experimental Design . . . . .	13
2.4	Experimental Results and Discussion . . . . .	20
2.5	Conclusion . . . . .	28
<b>3</b>	<b>Preventing Satisficing in Surveys</b>	<b>30</b>
3.1	Introduction . . . . .	30
3.2	Methods . . . . .	33
3.3	Results . . . . .	42
3.4	Discussion . . . . .	52
3.5	Future directions . . . . .	54
<b>4</b>	<b>Estimating the Number of Objects in Images</b>	<b>59</b>
4.1	Introduction . . . . .	59
4.2	Crowdsourcing Object Identification in Images . . . . .	60
4.3	Engineering . . . . .	62
4.4	Statistical Model and Implementation . . . . .	68
4.5	Experiment and Results . . . . .	73
4.6	Conclusion . . . . .	85
<b>5</b>	<b>Collecting labels for Word Sense Disambiguation</b>	<b>86</b>
5.1	Introduction . . . . .	86
5.2	Methods and data collection . . . . .	88
5.3	Results and data analysis . . . . .	91
5.4	Conclusion . . . . .	97



<b>6</b>	<b>Well-Being Prediction</b>	<b>99</b>
6.1	Introduction . . . . .	100
6.2	Background . . . . .	101
6.3	Method . . . . .	105
6.4	Evaluation . . . . .	110
6.5	Discussion: Well-Being Insights . . . . .	115
6.6	Conclusions . . . . .	117
<b>7</b>	<b>Bayesian Additive Regression Trees Implementation</b>	<b>119</b>
7.1	Introduction . . . . .	119
7.2	Overview of BART . . . . .	121
7.3	The <code>bartMachine</code> package . . . . .	128
7.4	Regression Features . . . . .	134
7.5	Classification Features . . . . .	154
7.6	Discussion . . . . .	158
<b>8</b>	<b>Incorporating Missingness into BART</b>	<b>160</b>
8.1	Introduction . . . . .	160
8.2	Background . . . . .	163
8.3	Missing Incorporated in Attributes within BART . . . . .	172
8.4	Generated Data Simulations . . . . .	175
8.5	Real Data Example . . . . .	181
8.6	Discussion . . . . .	183
<b>9</b>	<b>Variable Selection for BART</b>	<b>187</b>
9.1	Introduction . . . . .	188
9.2	Techniques for Variable Selection . . . . .	190
9.3	Calibrating BART output for Variable Selection . . . . .	193
9.4	Simulation Evaluation of BART-based Variable Selection . . . . .	206
9.5	Application to Gene Regulation in Yeast . . . . .	218
9.6	Conclusion . . . . .	226
<b>10</b>	<b>BART with Parametric Models of Heteroskedasticity</b>	<b>229</b>
10.1	Introduction . . . . .	229
10.2	Bayesian Heteroskedastic Regression . . . . .	231
10.3	Augmenting BART to Incorporate Heteroskedasticity . . . . .	232
10.4	Simulations . . . . .	239
10.5	Real Data Examples . . . . .	244
10.6	Discussion . . . . .	246
<b>11</b>	<b>Statistical Learning Model Visualization</b>	<b>249</b>
11.1	Introduction . . . . .	250
11.2	Background . . . . .	251
11.3	The ICE Toolbox . . . . .	254
11.4	Simulations . . . . .	261

11.5	Real Data . . . . .	266
11.6	A Visual Test for Additivity . . . . .	272
11.7	Discussion . . . . .	276
<b>12</b>	<b>Detecting Heterogeneous Effects via Crowdsourcing</b>	<b>279</b>
12.1	Introduction . . . . .	279
12.2	Experimental Methods and Design . . . . .	283
12.3	Results . . . . .	290
12.4	Conclusions and Future Directions . . . . .	293
<b>13</b>	<b>Matching on-the-fly in Sequential Experiments</b>	<b>295</b>
13.1	Introduction . . . . .	295
13.2	The Algorithm, Estimation, and Testing . . . . .	299
13.3	Simulation Studies . . . . .	307
13.4	Demonstration Using Real Data . . . . .	313
13.5	Discussion . . . . .	318
	<b>List of Figures</b>	<b>323</b>
	<b>List of Tables</b>	<b>326</b>
<b>A</b>	<b>Appendices</b>	<b>328</b>
A.1	Supplement for Chapter 2 . . . . .	328
A.2	Supplement for Chapter 7 . . . . .	339
A.3	Supplement for Chapter 9 . . . . .	350
A.4	Supplement for Chapter 10 . . . . .	353
A.5	Supplement for Chapter 11 . . . . .	362
A.6	Supplement for Chapter 12 . . . . .	363
A.7	Supplement for Chapter 13 . . . . .	368
A.8	BART for Panel Data . . . . .	379
	<b>Bibliography</b>	<b>401</b>

## **Abstract**

This thesis develops methods for the analysis and design of crowdsourced experiments and crowdsourced labeling tasks. Much of this document focuses on applications including running natural field experiments, estimating the number of objects in images, collecting labels for word sense disambiguation and collecting training samples to predict individual well-being. Observed shortcomings of the crowdsourced experiments inspired the development of methodology for running more powerful experiments via matching on-the-fly. Using the label data to estimate response functions inspired work on non-parametric function estimation using Bayesian Additive Regression Trees (BART). This work then inspired extensions to BART such as incorporation of missing data, variable selection, heteroskedastic modeling as well as a user-friendly R package. Using black-box function estimation as a practitioner inspired development of visualization procedures that give a glimpse into the inner workings of these estimations.

## Introduction

By Adam Kapelner

Imagine a world where you can release a small task to an anonymous person located anywhere in the world. This modern phenomenon is known as “crowdsourcing” which has experienced exponential growth since its inception in both industry and in academia.

In my observations over the past seven years, academic use of crowdsourcing falls into three categories:

- (a) **Label Collection** — Using the crowd to do a prespecified task such as mark an image, or determine the meaning of a word in a paragraph, write about a musical melody, etc. In this task, a worker is allowed to provide as many labels as he wishes.
- (b) **Experimentation** — Using the crowd to run an experiment to elicit a causal effect of an experimental condition on a prespecified outcome measure. With proper Institutional Review Board approval, these experiments can be natural field experiments, where the participants do not know they are part of an experiment with a randomized experimental condition. A worker is only allowed to complete one experimental task.

- (c) **Surveying** — Using participants in the crowd to answer a survey. One person per survey is allowed.

There are plenty of excellent primers on crowdsourcing in academia (e.g. my talk given at 2013’s Joint Statistical Meeting, Kapelner and Ungar, 2013).

The chapters of this thesis document are a collection of the work I have done in graduate school relevant to the analysis and design of crowdsourcing applications in all three of these academic use areas. The majority of this work is joint with my colleagues and each chapter’s contributors are appropriately marked at the beginning of the chapter and acknowledgements are appropriately marked at the close of each chapter. The chapters follow in approximate chronological order.

First, I describe the work we have done on crowdsourcing beginning with experimentation. We did an experiment to compare motivation of workers based on meaningfulness in Chapter 2. Then we did an experiment to see if using a method of presentation can improve worker survey responses in Chapter 3.

My interest then moved to using crowdsourcing for data labeling. I first create a system called **DistributeEyes** which collects markings of objects of interest in images from crowdsourced workers. We then attempt to estimate the total number of objects in a given image by combining labelings from the different workers. The system and estimation strategy is outlined in Chapter 4. I then built a crowdsourcing data collection app for a computational linguistics project whose goal was automatic word sense disambiguation in Chapter 5. I then joined a project where we used crowdsourcing labels to predict well-being. This is a work in progress and is detailed in Chapter 6.

From here, I became interested in using these crowdsourced data labels to make predictions. It is well known that black-box algorithms have better prediction than parametric models (see Breiman, 2001b). My interest turned towards Bayesian Additive Regression Trees (**BART**, Chipman et al., 2010), a statistical learning algorithm

for Bayesian non-parametric function estimation. I and a colleague programmed a parallel implementation of **BART** implemented as an **R** package with many user-friendly added features in Chapter 7. Then, I began to work on extensions to **BART**.

Chapter 8 details a method for incorporating missingness into the **BART** procedure with high predictive accuracy. An important problem in building machines for labeling is finding out which collected features drive accuracy. Chapter 9 outlines a procedure based on permutation testing to find features that drive the response function. Chapter 10 outlines a procedure to generalize **BART** to fit heteroskedastic models by incorporating linear models of heteroskedasticity. Many times in crowdsourcing applications, workers complete more than one labeling task. This induces a correlation structure in the design matrix. Extending **BART** to accommodate this type of panel data is a work in progress. The current implementation details are found in Appendix A.8.

It became frustrating to be a user of **BART**, a black-box algorithm, without being able to look inside and understand what the model fit was doing. To this end, I worked on a visualization procedure for general statistical learning which illustrates predictions by varying the values of the features. This work and its implementation in **R** as the package “**ICEbox**” is found in Chapter 11.

I then attempted to use crowdsourcing to test whether selection bias into a randomized experiment can affect the average treatment effect measured. Chapter 12 details this study and its results. I believe the flaw in the study was lack of power: the differences in effects is very small, so even at high sample size, no result was found. This failed experiment inspired me to think about a procedure that can increase power in crowdsourcing experiments. These experiments are *sequential* by nature, meaning that subjects enter over time and must be immediately assigned a treatment. We propose matching-on-the fly, a procedure detailed in Chapter 13.

## Motivation in Crowdsourcing Markets

By Dana Chandler and Adam Kapelner

### Abstract

We conduct the first natural field experiment to explore the relationship between the “meaningfulness” of a task and worker effort. We employed about 2,500 workers from Amazon’s Mechanical Turk (MTurk), an online labor market, to label medical images. Although given an identical task, we experimentally manipulated how the task was framed. Subjects in the *meaningful* treatment were told that they were labeling tumor cells in order to assist medical researchers, subjects in the *zero-context* condition (the control group) were not told the purpose of the task, and, in stark contrast, subjects in the *shredded* treatment were not given context and were additionally told that their work would be discarded. We found that when a task was framed more meaningfully, workers were more likely to participate. We also found that the meaningful treatment increased the quantity of output (with an insignificant change in quality) while the shredded treatment decreased the quality of output (with no change in quantity). We believe these results will generalize to other short-term labor markets. Our study also discusses MTurk as an exciting platform for running natural field experiments in economics.

## 2.1 Introduction

Economists, philosophers, and social scientists have long recognized that non-pecuniary factors are powerful motivators that influence choice of occupation. For a multidisciplinary literature review on the role of meaning in the workplace, we recommend Rosso et al. (2010). Previous studies in this area have generally been based on ethnographies, observational studies, or laboratory experiments. For instance, Wrzesniewski et al. (1997) used ethnographies to classify work into jobs, careers, or callings. Using an observation study, Preston (1989) demonstrated that workers may accept lower wages in the non-profit sector in order to produce goods with social externalities. Finally, Ariely et al. (2008) showed that labor had to be both recognizable and purposeful to have meaning. In this paper, we limit our discussion to the role of meaning in economics, particularly through the lens of competing differentials. We perform the first *natural field experiment* (Harrison and List, 2004) in a real effort task that manipulates levels of meaningfulness. This method overcomes a number of shortcomings of the previous literature, including: interview bias, omitted variable bias, and concerns of external validity beyond the laboratory.

We study whether employers can deliberately alter the perceived “meaningfulness” of a task in order to induce people to do more and higher quality work and thereby work for a lower wage. We chose a task that would appear meaningful for many people if given the right context — helping cancer researchers mark tumor cells in medical images. Subjects in the *meaningful* treatment were told the purpose of their task is to “help researchers identify tumor cells;” subjects in our *zero-context* group were not given any reason for their work and the cells were instead referred to as mere “objects of interest” and laborers in the *shredded* group were given zero context but also explicitly told that their labelings would be discarded upon submission. Hence, the pay structure, task requirements, and working conditions were identical, but we



added cues to alter the perceived meaningfulness of the task.

We recruited workers from the United States and India from Amazon’s Mechanical Turk (MTurk), an online labor market where people around the world complete short, “one-off” tasks for pay. The MTurk environment is a spot market for labor characterized by relative anonymity and a lack of strong reputational mechanisms. As a result, it is well-suited for an experiment involving the meaningfulness of a task since the variation we introduce regarding a task’s meaningfulness is less affected by desires to exhibit pro-social behavior or an anticipation of future work (career concerns). We ensured that our task appeared like any other task in the marketplace and was comparable in terms of difficulty, duration, and wage.

Our study is representative of the kinds of natural field experiments for which MTurk is particularly suited. Section 2.2.2 explores MTurk’s potential as a platform for field experimentation using the framework proposed in Levitt and List (2007, 2009).

We contribute to the literature on compensating wage differentials (Rosen, 1986) and the organizational behavioral literature on the role of meaning in the workplace (Rosso et al., 2010). Within economics, Stern (2004) provides quasi-experimental evidence on compensating differentials within the labor market for scientists by comparing wages for academic and private sector job offers among recent Ph.D. graduates. He finds that “scientists pay to be scientists” and require higher wages in order to accept private sector research jobs because of the reduced intellectual freedom and a reduced ability to interact with the scientific community and receive social recognition. Ariely et al. (2008) use a laboratory experiment with undergraduates to vary the meaningfulness of two separate tasks: (1) assembling Legos and (2) finding 10 instances of consecutive letters from a sheet of random letters. Our experiment augments experiment 1 in Ariely et al. (2008) by testing whether their results extend to the field. Additionally, we introduce a richer measure of task effort, namely *task*

*quality*. Where our experiments are comparable, we find that our results parallel theirs.

We find that the main effects of making our task more meaningful is to induce a higher fraction of workers to complete our task, hereafter dubbed as “induced to work.” In the meaningful treatment, 80.6% of people labeled at least one image compared with 76.2% in the zero-context and 72.3% in the shredded treatments.

After labeling their first image, workers were given the opportunity to label additional images at a declining piecerate. We also measure whether the treatments increase the quantity of images labeled. We classify participants as “high-output” workers if they label five or more images (an amount corresponding to roughly the top tercile of those who label) and we find that workers are approximately 23% more likely to be high-output workers in the meaningful group.

We introduce a measure of task quality by telling workers the importance of accurately labeling each cell by clicking as close to the center as possible. We first note that MTurk labor is high quality, with an average of 91% of cells found. The meaning treatment had an ambiguous effect, but the shredded condition in both countries lowered the proportion of cells found by about 7%.

By measuring both quantity and quality we are able to observe how task effort is apportioned between these two “dimensions of effort.” Do workers work “harder” or “longer” or both? We found an interesting result: the meaningful condition seems to increase quantity without a corresponding increase in quality and the shredded treatment decreases quality without a corresponding decrease in quantity. Investigating whether this pattern generalizes to other domains may be a fruitful future research avenue.

Finally, we calculate participants’ average hourly wage based on how long they spent on the task. We find that subjects in the meaningful group work for \$1.34 per hour, which is 6 cents less per hour than zero context participants and 14 cents less

per hour than shredded condition participants.

We expect our findings to generalize to other short-term work environments such as temporary employment or piecework. In these environments, employers may not consider that non-pecuniary incentives of meaningfulness matter; we argue that these incentives do matter, and to a significant degree.

Section 2.2 provides background on MTurk and discusses its use as a platform for conducting economic field experiments. Section 2.3 describes our experimental design. Section 3.3 presents our results and discussion and Section 2.5 concludes. Appendices in section A.1 provide full details on our experimental design and suggestions for conducting experiments using the MTurk platform.

## **2.2 Mechanical Turk and its potential for field experimentation**

Amazon’s Mechanical Turk (MTurk) is the largest online, task-based labor market and is used by hundreds of thousands of people worldwide. Individuals and companies can post tasks (known as Human Intelligence Tasks, or “HITs”) and have them completed by an on-demand labor force. Typical tasks include image labeling, audio transcription, and basic internet research. Academics also use MTurk to outsource low-skilled resource tasks such as identifying linguistic patterns in text (Sprouse, 2011) and labeling medical images (Chapter 4 of this document). The image labeling system from the latter study, known as “DistributeEyes,” was originally used by breast cancer researchers and was modified for our experiment.

Beyond simply using MTurk as a source of labor, academics have also begun using MTurk as a way to conduct online experiments. The remainder of the section highlights some of the ways this subject pool is used and places special emphasis on the suitability of the environment for natural field experiments in economics.

### 2.2.1 General use by social scientists

As Henrich et al. (2010) argue, many findings from social science are disproportionately based on what he calls “W.E.I.R.D.” subject pools (**W**estern, **E**ducated, **I**ndustrialized, **R**ich, and **D**emocratic) and as a result it is inappropriate to believe the results generalize to larger populations. Since MTurk has users from around the world, it is also possible to conduct research across cultures. For example, Eriksson and Simpson (2010) use a cross-national sample from MTurk to test whether differential preferences for competitive environments are explained by females’ stronger emotional reaction to losing, hypothesized by Croson and Gneezy (2009).

It is natural to ask whether results from MTurk generalize to other populations. Paolacci et al. (2010a) assuage these concerns by replicating three classic framing experiments on MTurk: The Asian Disease Problem, the Linda Problem and the Physician Problem; Horton et al. (2011) provide additional replication evidence for experiments related to framing, social preferences, and priming. Berinsky et al. (2012) argues that the MTurk population has “attractive characteristics” because it approximates gold-standard probability samples of the US population. All three studies find that the direction and magnitude of the effects line up well compared with those found in the laboratory.

An advantage of MTurk relative to the laboratory is that the researcher can rapidly scale experiments and recruit hundreds of subjects within only a few days and at substantially lower costs.<sup>1</sup>

---

<sup>1</sup>For example, in our study we paid 2,471 subjects \$789 total and they worked 701 hours (equating to 31 cents per observation). This includes 60 subjects whose data were not usable.

### 2.2.2 Suitability for natural field experiments in Economics

Apart from general usage by academics, the MTurk environment offers additional benefits for experimental economists and researchers conducting natural field experiments. We analyze the MTurk environment within the framework laid out in Levitt and List (2007, 2009).

In the ideal natural field experiment, “the environment is such that the subjects naturally undertake these tasks and [do not know] that they are participants in an experiment.” Additionally, the experimenter must exert a high degree of control over the environment without attracting attention or causing participants to behave unnaturally. MTurk’s power comes from the ability to construct customized and highly-tailored environments related to the question being studied. It is possible to collect very detailed measures of user behavior such as precise time spent on a webpage, mouse movements, and positions of clicks. In our experiment, we use such data to construct a precise quality measure.

MTurk is particularly well-suited to using experimenter-as-employer designs (Gneezy and List, 2006) as a way to study worker incentives and the employment relationship without having to rely on cooperation of private sector firms.<sup>2</sup> For example, Barankay (2010) posted identical image labeling tasks and varied whether workers were given feedback on their relative performance (i.e., ranking) in order to study whether providing rank-order feedback led workers to return for a subsequent work opportunity. For a more detailed overview of how online labor markets can be used in experiments, see Horton et al. (2011).

Levitt and List (2007) enumerate possible complications that arise when experimental findings are extrapolated outside the lab: *scrutiny*, *anonymity*, *stakes*, *selection*, and *artificial restrictions*. We analyze each complication in the context of our

---

<sup>2</sup>Barankay (2010) remarks that “the experimenter [posing] as the firm [gives] substantial control about the protocol and thereby eliminates many project risks related to field experiments.

experiment and in the context of experimentation using MTurk in general.

*Scrutiny and anonymity.* In the lab, experimenter effects can be powerful; subjects behave differently if they are aware their behavior is being watched. Relatedly, subjects frequently lack anonymity and believe their choices will be scrutinized after the experiment. In MTurk, interaction between workers and employers is almost non-existent; most tasks are completed without any communication and workers are only identifiable by a numeric identifier. Consequently, we believe that MTurk experiments are less likely to be biased by these complications.

*Stakes.* In the lab or field, it’s essential to “account properly for the differences in stakes across settings” (Levitt and List, 2007). We believe that our results would generalize to other short-term work environments, but would not expect them to be generalizable to long-term employment decisions such as occupational choice. Stakes must also be chosen adequately for the environment and so we were careful to match wages to the market average.

*Selection.* Experiments fail to be generalizable when “participants in the study differ in systematic ways from the actors engaged in the targeted real-world setting.” We know that within MTurk, it is unlikely that there is selection into our experiment since our task was designed similar in appearance to real tasks. The MTurk population also seems representative along a number of observable demographic characteristics (Berinsky et al., 2012); however, we acknowledge that there are potentially unobservable differences between our subject pool and the broader population. Still, we believe that MTurk subject behavior would generalize to workers’ behavior in other short-term labor markets.

*Artificial restrictions.* Lab experiments place unusual and artificial restrictions on the actions available to subjects and they examine only small, non-representative windows of time because the experimenter typically doesn’t have subjects and time horizons for an experiment. In structuring our experiment, workers had substan-

tial latitude in how they performed their task. In contrast with the lab, subjects could “show-up” to our task whenever they wanted, leave at will, and were not time-constrained. Nevertheless, we acknowledge that while our experiment succeeded in matching short-term labor environments like MTurk, that our results do not easily generalize to longer-term employment relationships.

Levitt and List (2009) highlight two limitations of field experiments vis-a-vis laboratory experiments: the *need for cooperation* with third parties and the difficulty of *replication*. MTurk does not suffer from these limitations. Work environments can be created by researchers without the need of a private sector partner, whose interests may diverge substantially from that of the researcher. Further, MTurk experiments can be replicated simply by downloading source code and re-running the experiment. In many ways, this allows a push-button replication that is far better than that offered in the lab.

## 2.3 Experimental Design

### 2.3.1 Subject recruitment

In running our randomized natural field experiment, we posted our experimental task so that it would appear like any other task (image labeling tasks are among the most commonly performed tasks on MTurk). Subjects had no indication they were participating in an experiment. Moreover, since MTurk is a market where people ordinarily perform one-off tasks, our experiment could be listed inconspicuously.

We hired a total of 2,471 workers (1,318 from the US and 1,153 from India). Although we tried to recruit equally from both countries, there were fewer Indians in our sample since attrition in India was higher. We collected each worker’s age and gender during a “colorblindness” test that we administered as part of the task. These

and other summary statistics can be found in Table 3.2. By contracting workers from the US and India, we can also test whether workers from each country respond differentially to the meaningfulness of a task.

Our task was presented so that it appeared like a one-time work opportunity (subjects were barred from doing the experiment more than once) and our design sought to maximize the amount of work we could extract during this short interaction. The first image labeling paid \$0.10, the next paid \$0.09, etc, leveling off at \$0.02 per image. This wage structure was also used in Ariely et al. (2008) and has the benefit of preventing people from working too long.

### 2.3.2 Description of experimental conditions

Upon accepting our task, workers provided basic demographic information and passed a color-blindness test. Next, they were randomized into either the *meaningful*, the *zero-context*, or the *shredded* condition. Those in the shredded condition were shown a warning message stating that their labeling will not be recorded and we gave them the option to leave. Then, all participants were forced to watch an instructional video which they could not fast-forward. See Appendix A.1.1 for the full script of the video as well as screenshots.

The video for the meaningful treatment began immediately with cues of meaning. We adopt a similar working definition of “meaningfulness” as used in Ariely et al. (2008): “Labor [or a task] is meaningful to the extent that (a) it is recognized and/or (b) has some point or purpose.”

We varied the levels of meaningfulness by altering the degree of recognition and the detail used to explain the purpose of our task. In our meaningful group, we provided “recognition” by thanking the laborers for working on our task. We then explained the “purpose” of the task by creating a narrative explaining how researchers were inundated with more medical images than they could possibly label and that



they needed the help of ordinary people. In contrast, the zero-context and shredded groups were not given recognition, told the purpose of the task, or thanked for participating; they were only given basic instructions. Analyzing the results from a post-manipulation check (see section 2.4.4), we are confident that these cues of meaning induced the desired affect.

Both videos identically described the wage structure and the mechanics of how to label cells and properly use the task interface (including zooming in/out and deleting points, which are metrics we analyze). However, in the meaningful treatment, cells were referred to as “cancerous tumor cells” whereas in the zero-context and shredded treatments, they were referred to as nondescript “objects of interest.” Except for this phrase change, both scripts were identical during the instructional sections of the videos. To emphasize these cues, workers in the meaningful group heard the words “tumor,” “tumor cells,” “cells,” etc. 16 times before labeling their first image and similar cues on the task interface reminded them of the purpose of the task as they labeled.

### **2.3.3 Task interface, incentive structure, and response variables**

After the video, we administered a short multiple-choice quiz testing workers’ comprehension of the task and user interface. In the shredded condition, we gave a final question asking workers to again acknowledge that their work will not be recorded.

Upon passing the quiz, workers were directed to a task interface which displayed the image to be labeled and allowed users to mark cancerous tumor cells (or “objects of interest”) by clicking (see figure 2.1). The image shown was one of ten look-alike photoshopped images displayed randomly. We also provide the workers with controls — *zoom functionality* and the ability to *delete points* — whose proper use would allow them to produce high-quality labelings.

During the experiment, we measured three response variables: (1) induced to work, (2) quantity of image labelings, and (3) quality of image labelings.

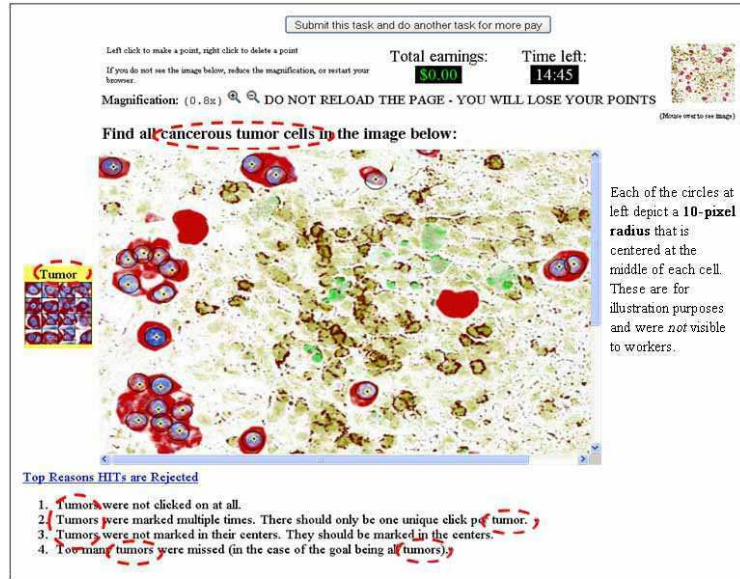


Figure 2.1: **Main task portal for a subject in the meaningful treatment**  
 Workers were asked to identify all tumors in the image. Each image had 90 cells and took 5 minutes on average. Our interface reminds the workers in 8 places that they are identifying tumor cells. The black circles around each point were *not* visible to participants. We display them to illustrate the size of a 10-pixel radius.

Many subjects can – and – do stop performing a task even after agreeing to complete it. While submitting bad work on MTurk is penalized, workers can abandon a task with only nominal penalty. Hence, we measure attrition with the response variable *induced to work*. Workers were only counted as induced to work if they watched the video, passed the quiz, and completed one image labeling. Our experimental design deliberately encourages attrition by imposing an upfront and unpaid cost of watching a three-minute instructional video and passing a quiz before moving on to the actual task.

Workers were paid \$0.10 for the first image labeling. They were then given an option to label another image for \$0.09, and then another image for \$0.08, and so

on.<sup>3</sup> At \$0.02, we stopped decreasing the wage and the worker was allowed to label images at this pay rate indefinitely. After each image, the worker could either collect what they had earned thus far, or label more images. We used the *quantity of image labelings* for our second response variable.

In our instructional video, we emphasized the importance of marking the exact center of each cell. When a worker labeled a cell by clicking on the image, we measured that click location to the nearest pixel. Thus, we were able to detect if the click came “close” to the actual cell. Our third response variable, *quality of image labelings* is the proportion of objects identified based on whether a worker’s click fell within a pixel radius from the object’s true center. We will discuss the radii we picked in the following section.

After workers chose to stop labeling images and collect their earnings, they were given a five-question PMC survey which asked whether they thought the task (a) was enjoyable (b) had purpose (c) gave them a sense of accomplishment (d) was meaningful (e) made their efforts recognized. Responses were collected on a five-point Likert scale. We also provided a text box to elicit free-response comments.<sup>4</sup>

### 2.3.4 Hypotheses

**Hypothesis 1** We hypothesize that at equal wages, the meaningful treatment will have the highest proportion of workers induced to work and the shredded condition will have the lowest proportion. In the following section, we provide theoretical justification for this prediction.

**Hypothesis 2** As in Ariely et al. (2008), we hypothesize that *quantity* of images labeled will be increasing in the level of meaningfulness.

---

<sup>3</sup>Each image was randomly picked from a pool of ten look-alike images.

<sup>4</sup>About 24% of respondents left comments (no difference across treatments).

**Hypothesis 3** In addition to quantity, we measure the *quality* of image labelings and hypothesize that this is increasing in the level of meaningfulness.

**Hypothesis 4** Based upon prior survey research on MTurk populations, we hypothesize that *Indian workers are less responsive to meaning*. Ipeirotis (2010) finds that Indians are more likely to have MTurk as a primary source of income (27% vs. 14% in the US). Likewise, people in the US are nearly twice as likely to report doing tasks because they are fun (41% vs. 20%). Therefore, one might expect financial motivations to be more important for Indian workers.<sup>5</sup>

### 2.3.5 Theoretical Framework

Hypothesis 1 is justified by modeling workers’ labor participation decisions using the modified compensating differentials framework in Rosen (1986). Instead of analyzing a worker’s choice between two jobs, a “dirty” and “clean” one, the worker has preferences over jobs with different levels of meaningfulness as characterized by our three treatments.<sup>6</sup> Each worker is given a take-it-or-leave-it offer that includes a sequence of wages and a meaning level and workers subsequently decide whether to participate (defined as labeling one or more images).

We abstract from the multi-stage decision process and focus on the worker’s initial decision. We suppose that workers have quasi-concave preferences  $U(C, T)$  that are strictly increasing in the continuous variables of consumption  $C$  and the degree of meaning associated with a task  $T$ . Depending on their treatment, workers are presented with one of three wage-meaning tuples  $(w, T)$  (points  $e$ ,  $b$ , or  $f$ ) that correspond to identical wage offers for our task with three levels of meaning ( $T \in \{S, ZC, M\}$ , with  $M > ZC > S$ ). Figure 2.2 plots indifference curves representing the total utility

---

<sup>5</sup>Although Horton et al. (2011) find that workers of both types are strongly motivated by money.

<sup>6</sup>Unlike (Rosen, 1986), workers are not presented with all three options at once and we do not model the market environment since each worker only receives a take-it-or-leave-it offer.

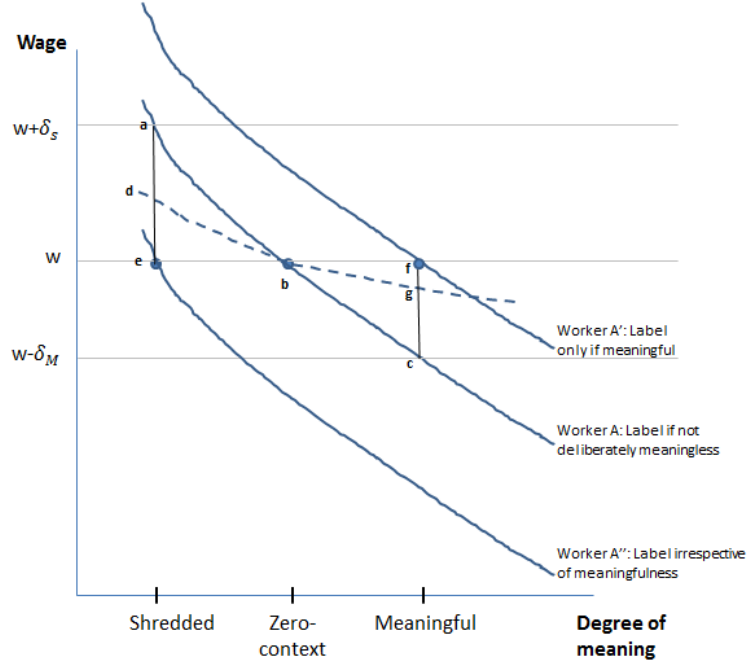


Figure 2.2: Example preferences as a function of task meaningfulness.

obtained from our offer. The three indifference curves plotted for workers  $A$ ,  $B$ , and  $C$  depict marginal participants in each group whose IR constraint would stop binding if the wage or meaning levels were any lower.<sup>7</sup> Consequently, workers' preferences for meaningfulness induce differential participation rates.

To illustrate, Worker  $A$  (whose indifference curve intersects points  $a$ ,  $b$ , and  $c$ ) would participate in all but the shredded treatment since the points  $b$  and  $f$  are above and to the right of the indifference curve. The line segments  $ae$  and  $fc$  represent hypothetical compensating differentials that worker  $A$  would require to participate in either the zero-context or meaningful treatment. We also link this framework to hypothesis 4, which discusses whether some workers may be less responsive to meaning. The flatter indifference curve represented by arc  $dbg$  shows a worker who is less responsive to meaning.

The data from our experiment help estimate the proportion of people who fall

---

<sup>7</sup>If a worker is indifferent between working or not, we assume he works.

between marginal workers  $A$ ,  $B$ , and  $C$ . However, we are unable to estimate the curvature of the utility function, changing costs of effort, or the extent of learning since we don't introduce additional treatments that vary wage and meaning simultaneously.<sup>8</sup> In future experiments, it would be worthwhile to create new treatments that target these elements.

	Shredded	Zero Context	Meaningful	US only	India only
% Induced to Work	.723	.762	.806	.85	.666
# Images (if $\geq 1$ )	$5.94 \pm 6.8$	$6.11 \pm 6.9$	$7.12 \pm 7.6$	$5.86 \pm 6.1$	$7.17 \pm 8.3$
Did $\geq 2$ labelings	.696	.706	.75	.797	.627
Did $\geq 5$ labelings	.343	.371	.456	.406	.373
Avg Hourly Wage	\$1.49	\$1.41	\$1.34	\$1.50	\$1.29
% Male	.616	.615	.58	.483	.743
Age	$29.6 \pm 9.3$	$29.6 \pm 9.5$	$29.3 \pm 9.1$	$31.8 \pm 10.5$	$26.9 \pm 6.8$
$N$	828	798	845	1318	1153
Coarse quality	$.883 \pm .21$	$.904 \pm .18$	$.930 \pm .14$	$.924 \pm .15$	$.881 \pm .21$
Fine quality	$.614 \pm .22$	$.651 \pm .21$	$.676 \pm .18$	$.668 \pm .19$	$.621 \pm .26$
PMC Meaning	$3.44 \pm 1.3$	$3.54 \pm 1.2$	$4.37 \pm 0.9$	$3.67 \pm 1.3$	$3.98 \pm 1.1$

Table 2.1: Summary statistics for response variables and demographics by treatment and country. The statistics for the quality metrics are computed by averaging each worker's average quality (only for workers who labeled one or more images). The statistics for the PMC meaning question only include workers who finished the task and survey.

## 2.4 Experimental Results and Discussion

We ran the experiment on  $N = 2,471$  subjects (1,318 from the United States and 1,153 from India). Table 3.2 shows summary statistics for our response variables (induced to work, number of images, and quality), demographic variables, and hourly wage.

<sup>8</sup>For this reason, we omit these important factors from our model.

Broadly speaking, as the level of meaning increases, subjects are more likely to participate and they label more images and with higher quality. Across all treatments, US workers participate more often, label more images, and mark points with greater accuracy. Table 2.2 uses a heatmap to illustrate our main effect sizes and their significance levels by treatment, country, and response variable. Each cell indicates the size of a treatment effect relative to the control (i.e., zero context condition). Statistically significant *positive* effects are indicated using green fill where darker green indicates higher levels of significance. Statistically significant *negative* effects are indicated using red fill where darker red indicates higher levels of significance. Black text without fill indicates effects that are marginally significant ( $p < 0.10$ ). Light gray text indicates significance levels above 0.10.

	Induced to work	Did $\geq 5$ labelings	Fine Quality	Average Hourly Wage
Meaningful	↑ 4.6%*	↑ 8.5%***	↑ 0.7%	↓ 4.5%
Meaningful (US)	↑ 5.1%*	↑ 8.9%**	↑ 3.9%	↓ 7.7%
Meaningful (India)	↓ 2.3%	↑ 7.0%*	↓ 3.1%	↑ 0.5%
Shredded	↓ 4.0%	↓ 2.8%	↓ 7.2%***	↑ 5.6%
Shredded (US)	↓ 2.3%	↓ 5.0%	↓ 6.1%*	↑ 9.5%
Shredded (India)	↓ 6.8%	↓ 1.6%	↓ 8.7%**	↓ 1.4%

\*  $p < .05$ , \*\*  $p < .01$ , \*\*\*  $p < .001$ , black text indicates  $p < .10$  and grey text indicates  $p > 0.10$

Table 2.2: A heatmap illustration of our results. Rows 1 and 4 consider data from both America and India combined. Columns 1, 2, 3 show the results of regressions and column 4 shows the result of two-sample t-tests. Results reported are from regressions without demographic controls.

Overall, we observe that the meaningful condition induces an increase in quantity without significantly increasing quality, and the shredded condition induces a quality decrease with quantity remaining constant. This “checkerboard effect” may indicate that meaning plays a role in moderating how workers trade quantity for quality i.e.

how their energy is channeled in the task.

We now investigate each response variable individually.

### **2.4.1 Labor Participation Results: “Induced to work”**

We investigate how treatment and country affects whether or not subjects chose to do our task. Unlike in a laboratory environment, our subjects were workers in a relatively anonymous labor market and were not paid a “show-up fee.” On MTurk, workers frequently start but do not finish tasks; attrition is therefore a practical concern for employers who hire from this market. In our experiment, on average, 25% of subjects began, but did not follow-through by completing one full labeling.

Even in this difficult environment, we were able to increase participation among workers by roughly 4.6% by framing the task as more meaningful (see columns 1 and 2 of table 2.3). The effect is robust to including various controls for age, gender, and time of day effects. As a subject in the meaningful treatment told us, “It’s always nice to have [HITs] that take some thought and mean something to complete. Thank you for bringing them to MTurk.” The shredded treatment discouraged workers and caused them to work 4.0% less often but the effect was less significant ( $p = 0.057$  without controls and  $p = 0.082$  with controls). Thus, hypothesis 1 seems to be correct.

Irrespective of treatment, subjects from India completed an image 18.5% less often ( $p < 0.001$ ) than subjects from the US. We were interested in interactions between country and treatment, so we ran the separate induced-to-work regression results by country (unshown). We did not find significant effects within the individual countries because we were underpowered to detect this effect when the sample size was halved. We find no difference in the treatment effect for induced to work between India and the United States ( $p = 0.97$ ). This is inconsistent with hypothesis 4 where we predicted Indian subjects to respond more strongly to pecuniary incentives.



	Induced	Induced	Did $\geq 2$	Did $\geq 2$	Did $\geq 5$	Did $\geq 5$
Meaningful	0.046*	0.046*	0.047*	0.050*	0.085***	0.088***
	(0.020)	(0.020)	(0.022)	(0.022)	(0.024)	(0.024)
Shredded	-0.040	-0.037	-0.012	-0.005	-0.028	-0.023
	(0.021)	(0.021)	(0.022)	(0.022)	(0.024)	(0.024)
India	-0.185***	-0.183***	-0.170***	-0.156***	-0.035	-0.003
	(0.017)	(0.018)	(0.018)	(0.019)	(0.019)	(0.021)
Male		0.006		-0.029		-0.081***
		(0.018)		(0.019)		(0.021)
Constant	0.848***	0.907***	0.785***	0.873***	0.387***	0.460***
Controls						
Age		0.23		0.29		0.92
Time of Day		0.16		0.06		0.46
Day of Week		0.08		0.00**		0.55
$R^2$	0.05	0.06	0.04	0.05	0.01	0.02
$N$	2471	2471	2471	2471	2471	2471

\*  $p < .05$ , \*\*  $p < .01$ , \*\*\*  $p < .001$

Table 2.3: Robust linear regression results for the main treatment effects on quantity of images. Columns 1, 3 and 5 only include treatments and country. Columns 2, 4, and 6 control for gender, age categories, time of day, and day of week. Rows 6-8 show  $p$ -values for the partial  $F$ -test for sets of different types of control variables.

It is also possible that the effects for induced to work were weak because subjects could have still attributed meaning to the zero context and shredded conditions, a problem that will affect our results for quantity and quality as well. This serves to bias our treatment effects downward suggesting that the true effect of meaning would be larger. For instance, one zero-context subject told us, “I assumed the ‘objects’ were cells so I guess that was kind of interesting.” Another subject in the zero-context treatment advised us, “you could put MTurkers to good use doing similar work with images, e.g. in dosimetry or pathology ... and it would free up medical professionals to do the heavier work.”

### 2.4.2 Quantity Results: Number of images labeled

Table 3.2 shows that the number of images increased with meaning. However, this result is conditional on being induced to work and is therefore contaminated with selection bias. We follow Angrist (2001) and handle selection by creating a dummy variable for “did two or more labelings” and a dummy for “did five or more labelings” and use them as responses (other cutoffs produced similar results).

We find mixed results regarding whether the the level of meaningfulness affects the quantity of output. Being assigned to the meaningful treatment group *did* have a positive effect, but assignment to the shredded treatment did not result in a corresponding decrease in output.

Analyzing the outcome “two or more labelings,” column 3 of table 2.3 shows that the meaningful treatment induced 4.7% more subjects to label two or more images ( $p < 0.05$ ). The shredded treatment had no effect. Analyzing the outcome “five or more labelings” (column 5), which we denote as “high-output workers,”<sup>9</sup> the meaningful treatment was highly significant and induced 8.5% more workers ( $p < 0.001$  with and without controls), an increase of nearly 23 percent, and the shredded treatment again has no effect.

Hypothesis 2 (quantity increases with meaningfulness) seems to be correct only when comparing the meaningful treatment to the zero-context treatment. An ambiguous effect of the shredded treatment on quantity is also reported by Ariely et al. (2008).

We didn’t find differential effects between the United States and India. In an unshown regression, we found that Americans were 9.5% more likely to label five or more images ( $p < 0.01$ ) and Indians were 8.4% more likely to label five or more ( $p < 0.05$ ). These two effects were not found to be different ( $p = 0.84$ ) which is inconsistent

---

<sup>9</sup>Labeling five or more images corresponds to the top tercile of quantity among people who were induced to work.

with hypothesis 4 that Indians are more motivated by pecuniary incentives than Americans.

Interestingly, we also observed a number of “target-earners” who stopped upon reaching exactly one dollar in earnings. A mass of 16 participants stopped at one dollar, while one participant stopped at \$1.02 and not one stopped at \$0.98, an effect also observed by Horton et al. (2011). The worker who labored longest spent 2 hours and 35 minutes and labeled 77 images.

### 2.4.3 Quality Results: Accuracy of labeling

Quality was measured by the fraction of cells labeled at a distance of five pixels (“coarse quality”) and two pixels (“fine quality”) from their true centers. In presenting our results (see table 2.4), we analyze the treatment effects using our fine quality measure. The coarse quality regression results were similar, but the fine quality had a much more dispersed distribution.<sup>10</sup>

Our main result is that fine quality was 7.2% lower in the shredded treatment, but there wasn’t a large corresponding increase in the meaningful treatment.<sup>11</sup> This makes sense; if the workers knew their labelings weren’t going to be checked, there is no incentive to mark points carefully. This result was not different across countries (regression unshown). The meaningful treatment has a marginally significant effect only in the United States, where fine quality increased by 3.9% ( $p = 0.092$  without controls and  $p = 0.044$  with controls), but there was no effect in India. Thus, hypothesis 3 (quality increases with meaningfulness) seems to be correct *only* when

---

<sup>10</sup>The inter-quartile range of coarse quality overall was [93.3%, 97.2%] whereas the IQR of fine quality was overall [54.7%, 80.0%].

<sup>11</sup>One caveat with our quality results is that we only observe quality for people who were induced to work and selected into our experiment (we have “attrition bias”). Attrition was 4% higher in the shredded treatment and we presume that the people who opted out of labeling images would have labeled them with far worse quality had they remained in the experiment.

	Both Countries		Fine Quality United States		India	
Meaningful	0.007 (0.017)	0.014 (0.014)	0.039 (0.023)	0.039* (0.019)	-0.031 (0.025)	-0.013 (0.021)
Shredded	-0.072*** (0.021)	-0.074*** (0.017)	-0.061* (0.027)	-0.066** (0.023)	-0.087** (0.031)	-0.073** (0.023)
India	-0.053*** (0.015)	-0.057*** (0.013)				
Male		0.053*** (0.013)		0.014 (0.017)		0.100*** (0.021)
Labelings 6—10		-0.018** (0.006)		-0.024** (0.008)		-0.016* (0.008)
Labelings $\geq 11$		-0.140*** (0.017)		-0.116*** (0.029)		-0.148*** (0.020)
Constant	0.666***	0.645***	0.651***	0.625***	0.634***	0.588***
Controls						
Image		0.00***		0.00***		0.00***
Age		0.10		0.01**		0.25
Time of Day		0.33		0.29		0.78
Day of Week		0.12		0.46		0.26
$R^2$	0.04	0.15	0.04	0.12	0.02	0.20
$N$	12724	12724	6777	6777	5947	5947

\*  $p < .05$ , \*\*  $p < .01$ , \*\*\*  $p < .001$

Table 2.4: Robust linear regression clustered by subject for country and treatment on fine quality as measured by the number of cells found two pixels from their exact centers. Columns 1, 3 and 5 include only treatments and country. Columns 2, 4, and 6 control for number of images, the particular image (of the ten images), gender, age categories, time of day, and day of week.

comparing the shredded to the zero context treatment which is surprising.

Although Indian workers were less accurate than United States workers and had 5.3% lower quality ( $p < 0.001$  and robust to controls), United States and Indian workers did not respond differentially to the shredded treatment ( $p = 0.53$ ). This again is inconsistent with hypothesis 4.

Experience matters. Once subjects had between 6 and 10 labelings under their belt, they were 1.8% less accurate ( $p < 0.01$ ), and if they had done more than 10 labelings, they were 14% less accurate ( $p < 0.001$ ). This result may reflect negative

selection — subjects who labeled a very high number of images were probably working too fast or not carefully enough.<sup>12</sup> Finally, we found that some of the ten images were substantial harder to label accurately than others (a partial F-test for equality of fixed effects results in  $p < 0.001$ ).

#### 2.4.4 Post Manipulation Check Results

In order to understand how our treatments affected the perceived meaningfulness of the task, we gave a post manipulation check to all subjects who completed at least one image and did not abandon the task before payment. This data should be interpreted cautiously given that subjects who completed the tasks and our survey are *not* representative of all subjects in our experiment.<sup>13</sup>

We found that those in the meaningful treatment rated significantly higher in the post manipulation check in both the United States and India. Using a five-point Likert scale, we asked workers to rate the perceived level of meaningfulness, purpose, enjoyment, accomplishment, and recognition. In the meaningful treatment, subjective ratings were higher in all categories but the self-rated level of meaningfulness and purpose were the highest. The level of meaningfulness was 1.3 points higher in the US and 0.6 points higher in the India; the level of perceived porposefulness was 1.2 points higher in America and 0.5 points higher in India. In the United States, the level of accomplishment only increased by 0.8 and the level of enjoyment and recognition increased by 0.3 and 0.5 respectively with a marginal increase in India. As a US participant told us, “I felt it was a privilege to work on something so important and

---

<sup>12</sup>Anecdotaly, subjects from the shredded condition who submitted comments regarding the task were less likely to have expressed concerns about their accuracy. One subject from the meaningful group remarked that “[his] mouse was too sensitive to click accurately, even all the way zoomed in,” but we found no such apologies or comments from people in the shredded group.

<sup>13</sup>Ideally, we would have collected this information immediately after introducing the treatment condition. However, doing so would have compromised the credibility of our natural field experiment.

I would like to thank you for the opportunity.”

We conclude that the meaningful frames accomplished their goal. Remarkably, those in the shredded treatment in either country did not report significantly lower ratings on any of the items in the post manipulation check. Thus, the shredded treatment may not have had the desired effect.

## 2.5 Conclusion

Our experiment is the first that uses a natural field experiment in a real labor market to examine how a task’s meaningfulness influences labor supply.

Overall, we found that the greater the amount of meaning, the more likely a subject is to participate, the more output they produce, the higher quality output they produce, and the less compensation they require for their time. We also observe an interesting effect: high meaning increases *quantity* of output (with an insignificant increase in quality) and low meaning decreases *quality* of output (with no change in quantity). It is possible that the level of perceived meaning affects how workers substitute their efforts between task quantity and task quality. The effect sizes were found to be the same in the US and India.

Our finding has important implications for those who employ labor in any short-term capacity besides crowdsourcing, such as temp-work or piecework. As the world begins to outsource more of its work to anonymous pools of labor, it is vital to understand the dynamics of this labor market and the degree to which non-pecuniary incentives matter. This study demonstrates that they do matter, and they matter to a significant degree.

This study also serves as an example of what MTurk offers economists: an excellent platform for high internal validity natural field experiments while evading the external validity problems that may occur in laboratory environments.

## Acknowledgements

Both authors contributed equally to this work. The authors wish to thank Professor Susan Holmes of Stanford University for comments and for allowing us to adapt the DistributeEyes software for our experiment (funded under NIH grant #R01GM086884-02). They gratefully acknowledge financial support from the George and Obie Schultz Fund and the NSF Graduate Research Fellowship Program. The authors also thank Iwan Barankay, Lawrence Brown, Rob Cohen, Geoff Goodwin, Patrick DeJarnette, John Horton, David Jiménez-Gomez, Emir Kamenica, Abba Krieger, Steven Levitt, Blakeley McShane, Susanne Neckermann, Paul Rozin, Martin Seligman, Jesse Shapiro, Jörg Spenkuch, Jan Stoop, Chad Syverson, Mike Thomas, Adi Wyner, seminar participants at the University of Chicago, and our reviewers.

## Preventing Satisficing in Surveys

By Adam Kapelner and Dana Chandler

### Abstract

Researchers are increasingly using online labor markets such as Amazon’s Mechanical Turk (MTurk) as a source of inexpensive data. One of the most popular tasks is answering surveys. However, without adequate controls, researchers should be concerned that respondents may fill out surveys haphazardly in the unsupervised environment of the Internet. Social scientists refer to mental shortcuts that people take as “satisficing” and this concept has been applied to how respondents take surveys. We examine the prevalence of survey satisficing on MTurk. We present a question-presentation method, called *Kapcha*, which we believe reduces satisficing, thereby improving the quality of survey results. We also present an open-source platform for further survey experimentation on MTurk.

### 3.1 Introduction

It has been well established that survey-takers may “satisfice” (i.e., take mental shortcuts) to economize on the amount of effort and attention they devote to filling out



a survey (Krosnick, 1991).<sup>1</sup> As a result, the quality of data in surveys may be lower than researchers’ expectations. Because surveys attempt to measure internal mental processes, they are by their very nature not easily verifiable by external sources. This presents a potential problem for the many researchers who are beginning to employ Amazon’s Mechanical Turk (MTurk) workers to answer surveys and participate in academic research (Paolacci et al., 2010b, Horton and Chilton, 2010, Chandler and Kapelner, 2013 which is also Chapter 2 of this document). Moreover, unlike other tasks completed on MTurk, inaccuracies in survey data cannot be remedied by having multiple workers complete a survey, nor is there an easy way to check them against “gold-standard” data.<sup>2</sup>

In our experiment, we examine alternative ways to present survey questions in order to make respondents read and answer questions more carefully.

Our first treatment “exhorts” participants to take our survey seriously. We ask for their careful consideration of our questions by placing a message in prominent red text on the bottom of every question. Surprisingly, this has no effect.

Our two other treatments took a more economic approach and attempted to alter the incentives of survey-takers who ordinarily have an incentive to fill out questions as quickly as possible in order to maximize their hourly wage and exert minimal cognitive effort. More specifically, both treatments force the participant to see the question for a certain “waiting period”. Combined, these waiting period treatments improved the quality of survey responses by 10.0% ( $p < 0.001$ ). Under the waiting period treatments, the participant is forced to spend more time on each question

---

<sup>1</sup>Krosnick (1991) applies Simon (1955)’s famous idea of satisficing to how respondents complete surveys.

<sup>2</sup>In an MTurk context, “gold-standard” data refers to asking workers questions to which the surveyor already knows the answer as a way to identify bad workers. Although this is straightforward for an image labeling task (e.g. Holmes and Kapelner, 2010 which is also Chapter 4 of this document and Sorokin and Forsyth, 2008), it is less clear how to apply this concept to surveys.

and once there, we hypothesize that they will spend more time thinking about and thoughtfully answering questions.

The first of these two treatments, called simply the *Timing control* treatment, features a disabled continue button for the duration of the waiting period. The second of these treatments, referred to as the *Kapcha*<sup>3</sup> has a waiting period equal to that of the *Timing control* treatment, but also attempts to attract the attention of respondents by sequentially “fading in” each word in the question’s directions, its prompt, and its answer choices. This treatment was the most effective and improved quality by approximately 13%.

To proxy for quality, which is largely unobservable, we introduce a “trick question” into the survey as a way of measuring whether people carefully read instructions. We echo the methodology from Oppenheimer et al. (2009) who call this trick question an *instructional manipulation check* (IMC). Additionally, we give respondents two hypothetical thought experiments where we ask them to imagine how they would behave under certain conditions. The conditions are identical except for a *subtle word change* that would only be apparent if the instructions were read carefully — hence, for close readers, there should be a greater difference in reported behavior as compared with people who were merely skimming.<sup>4</sup>

This paper presents initial evidence on alternative ways to present survey questions in order to reduce satisficing. We hypothesize that altering the cost-benefit analysis undertaken by survey respondents is the mechanism which reduces satisficing. The approach we present has the benefit of improving the quality of results

---

<sup>3</sup>The name was inspired from the “Captcha” Internet challenge-response test to ensure a human response (Von Ahn et al., 2003).

<sup>4</sup>This portion of our experiment replicates Study 1 in Oppenheimer et al. (2009). Their primary focus was to identify subsamples of higher quality data and to eliminate the “noisy data” (i.e., the participants who did not read the instructions carefully enough to pass the trick question). This enables researchers to increase the statistical power of their experiments.

without increasing monetary cost or convenience for the surveyor. We also examine the prevalence of satisficing and how it may vary across respondent demographics. Finally, we discuss ideas for further improving how to present survey questions.

Section 13.2 explains our experimental methods. Section 3.3 illustrates our most important results, section 13.5 concludes, and section 3.5 talks about future directions. Appendix 3.5 describes the TurkSurveyor open-source package for running experiments and Appendix 11.7 provides links to our source code and data so that others may replicate and verify our results.

## 3.2 Methods

### 3.2.1 Recruitment of Participants

We designed an MTurk HIT (our “task”) to appear as a nondescript survey task similar to many others that are now popular on MTurk. By making our survey appear like any other, we intended to recruit a population that is representative of the MTurk survey-taking population.

We entitled our task “Take a short 30-question survey — \$0.11USD” and its description was “Answer a few short questions for a survey”. The preview pane of the HIT only displayed, “Welcome to the short survey! In this survey you will answer 30 questions. You may only do one survey.” Our HIT was labeled with the keywords “survey”, “questionnaire”, “survey”, “poll”, “opinion”, “study”, and “experiment” so that people specifically looking for survey tasks could easily find our task. However, we also wanted to attract workers who were not specifically looking for survey tasks. Therefore, we posted batches of tasks at various times.

We recruited 784 MTurk workers from the United States.<sup>5</sup> Each worker was only

---

<sup>5</sup>This restriction reduces the influence of any confounding language-specific or cultural effects.

allowed to complete one survey task and took one of four different versions of the survey according to their randomized assignment. In total, 727 workers completed the entire survey and each was paid \$0.11USD.<sup>6</sup>

We posted in batches of 200 tasks at a time four times per day (at 10AM, 4PM, 10PM, and 4AM EST) as to not bias for early-riser or night-owl workers. Altogether, 18 bunches (of 200 HITs each) were posted between September 1 and September 5, 2010. All HITs expired six hours after creation as to not interfere with the subsequent batch. Note that if we had posted our tasks as one gargantuan batch and waited until completion (possibly a week or longer), we would have attracted a majority of workers who were *specifically* looking for survey tasks (most likely searching for them via keyword) rather than a more general sample of workers.<sup>7</sup> The workers were given a maximum of 45 minutes to complete the task.<sup>8</sup>

### 3.2.2 Treatments

To test the satisficing-reducing effect of the *Kapcha*, we randomized each participant into one of four treatments which are described below, summarized in table 3.1, and

---

Generalizing our results to other countries and languages may be a fruitful future research direction.

<sup>6</sup>The workers worked 81.3 hours at an average wage of \$0.98/hr and a total cash cost to the experimenters of \$87.97 (including Amazon’s fee of 10%). In this calculation, we ignore the time they spent on the feedback question and the bonuses we paid.

<sup>7</sup>This phenomenon is due to HITs rapidly losing prominence in the public listings and eventually being relegated to obscurity where they may only be found by those searching via keyword (see Chilton et al., 2010). Also note that we save the time each HIT was created at and expires at. We use this information to check at what point in the HIT listing life-cycle the worker accepted the HIT.

<sup>8</sup>Even though the survey was short, we wanted to give ample time to be able to collect data on task breaks. Note that few workers took advantage of the long time limit; 117 workers (16%) took more than 15 minutes and only 61 workers (8%) took more than 30 minutes.

pictured in figures 3.1a–d.<sup>9</sup>

Treatment	Increase perceived value of survey	Force slow down	Attract att- ention to ind- ividual words
<i>Control</i>			
<i>Exhortation</i>	✓		
<i>Timing control</i>		✓	
<i>Kapcha</i>		✓	✓

Table 3.1: Overview of treatments and how they improve data quality

The first treatment was the *Control* where the questions were displayed in a similar fashion as any other online survey.

Our *Exhortation* treatment presents survey questions in an identical way as the *Control* treatment except that we try to increase the survey taker’s motivation by reminding them in alarming red text at the bottom of each question page to “Please answer accurately. Your responses will be used for research.” Past research on survey design has shown that respondents are more likely to devote effort to completing surveys if they perceive them as valuable because they contribute to research (Krosnick, 1991).

Rather than using exhortation, our *Timing control* and *Kapcha* treatments induce more careful survey taking by changing the incentives of a respondent. In short, we lower the payoff to satisficing.<sup>10</sup> When respondents can breeze through a survey and click one answer after another without delay, they may be tempted to satisfice — i.e.,

---

<sup>9</sup>Videos illustrating the four treatments are available at <http://danachandler.com/kapchastudy.html>

<sup>10</sup>If the *Exhortation* treatment increased the rate at which people passed the trick question (which it did not), we might have worried that this framing could bias the way survey takers answer questions, particularly socially sensitive ones, since it reminds the respondent that they are under

click the first answer that seems correct or any answer at random. If, however, survey respondents are forced to wait before proceeding to the next question, we hypothesize that they will use this time to think more carefully about how to answer.

Our *Timing control* treatment is identical to the *Control* treatment except that the continue button is disabled and has a spinning graphic during a waiting period<sup>11</sup> after which the continue button is enabled.

The *Kapcha* treatment goes one step further and, in addition to slowing down the respondent for a time equal to the *Timing control* treatment, also draws additional attention to the instructions and answer choices by “fading-in” the survey’s words at 250 words per minute.

The delay time for the questions in the *Timing control* treatment were calibrated to be the same total fade-in time for the *Kapcha* participant’s question. By controlling for the timed delay, we were able to isolate the additional effect due to forcing the respondent to pay attention to the words in the *Kapcha*.

Although this is the first research to our knowledge that studies waiting periods and textual fade-ins, there is a long history of research on how various forms of survey implementation affect response. Two interesting examples include how self-administration lead respondents to answer sensitive questions more truthfully and how questions that are accompanied by audio do the same among people who might not understand the text (especially among low-literacy respondents). Recently, Couper et al. (2009) has helped separate the effect of the self-administration and the audio component.<sup>12</sup>

---

scrutiny. In social psychology, over-reporting “positive” behaviors is known as the “social desirability bias” (DeMaio, 1984).

<sup>11</sup>We peg the waiting period to the time it takes an average person to read the number of words in each question. Taylor (1965) finds that the average reading speed for college-level readers is 280 words per minute and 250 for twelfth-graders. We chose 250 words per minute.

<sup>12</sup>For an excellent, though slightly dated review of various survey presentation formats and the issues they try to overcome, see Chapter 10.1 of Tourangeau et al. (2000)

### 3.2.3 Custom Survey Task Design

As soon as the worker accepted the HIT, they were given a page with directions that explained the length of the survey and asked to begin when ready. Depending on the treatment, we also added an additional sentence or two to the instructions in order to explain the particularities associated with each treatment. For our *Exhortation* group, we emphasized the importance of giving accurate and honest answers. In our *Timing control* group, we told participants that the continue button would be disabled for a short time so they would have more time to read and answer each question. For our *Kapcha* group, we mentioned how words and answer choices would appear one at a time.

After reading the directions, the worker began the survey task which consisted of 30 questions plus two optional questions eliciting feedback. Each question was presented individually so that the respondent must click submit before moving onto the next question.<sup>13</sup>

Our first question, “question A”, is a hypothetical thought experiment (which we call the soda-pricing example) that “demonstrates how different expectations can change people’s willingness to pay for identical experiences” (Oppenheimer et al., 2009). The question text is shown below. The subtle text manipulation which induces an effect according to Thaler (1985) is shown in brackets and will be denoted as the “run-down” vs. “fancy” treatments:

You are on the beach on a hot day. For the last hour you have been thinking about how much you would enjoy an ice cold can of soda. Your companion needs to make a phone call and offers to bring back a soda from the only nearby place where drinks are sold, which happens to be a

---

<sup>13</sup>Note that most surveys on MTurk display all questions on one page. Presenting questions one at a time, as we do in our study, probably serves to reduce satisficing. A future study would allow us to determine how the number of questions on each page affects satisficing.

[run-down grocery store / fancy resort]. Your companion asks how much you are willing to pay for the soda and will only buy it if it is below the price you state. How much are you willing to pay?

It has been shown repeatedly in the literature that people are willing to pay more when the beverage comes from a fancy resort. If the workers were reading the instructions carefully, we expect them to pay a higher price in the “fancy” treatment.

The worker was then given “question B,” another hypothetical thought experiment (which we call the football attendance example), which demonstrates that people are susceptible to the sunk cost fallacy (for screenshots, see figure 3.1a–d). The question text is shown below. The subtle text manipulation which induces an effect according to Thaler (1985) is shown in brackets and will be denoted as the “paid” vs “free” treatments:

Imagine that your favorite football team is playing an important game. You have a ticket to the game that you have [paid handsomely for / received for free from a friend]. However, on the day of the game, it happens to be freezing cold. What do you do?

Their intention was gauged on a nine-point scale where 1 was labeled “definitely stay at home” and 9 was labeled “definitely go to the game”. It has been shown that people who read the treatment where they paid for the tickets are more likely to go to the game.

By randomizing the text changes independently of treatments, we were able compare the *strength* of these two well-established psychological effects across the four types of survey-presentation.

The participant then answered an “instructional manipulation check” (IMC) question. Once again, this is a trick question designed to gauge whether the participant reads and follows directions carefully. The instructions of the IMC question asks



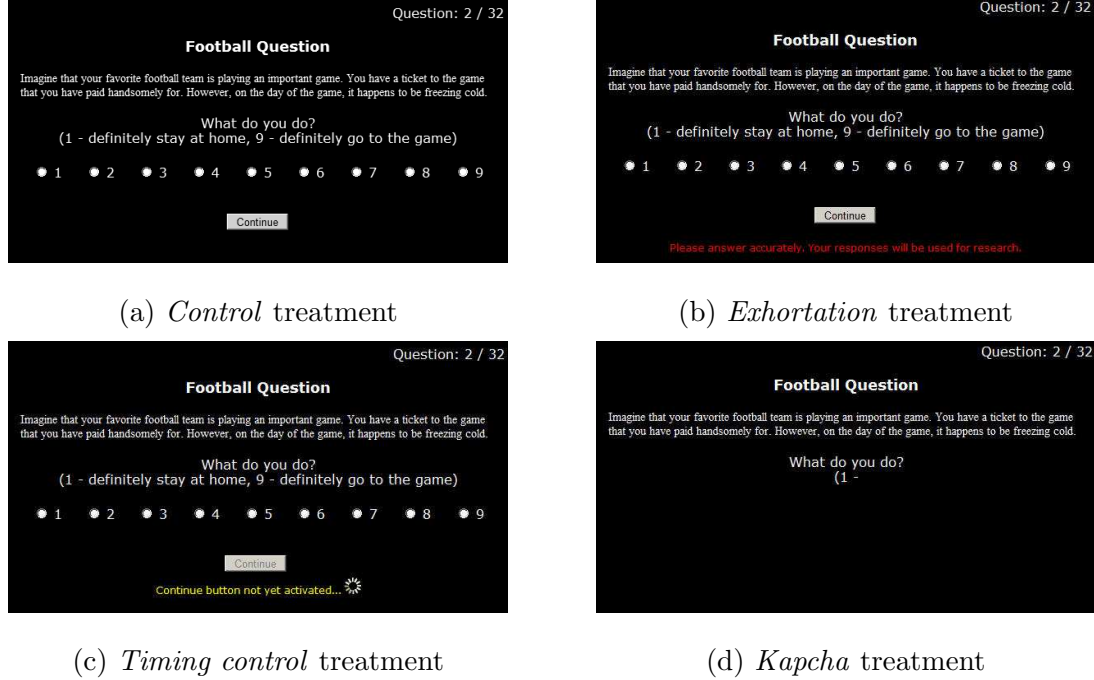


Figure 3.1: The participant’s screen during question B, the intent to go to a football game, shown for all four experimental treatments (in the “paid” treatment). The *Timing control* and *Kapcha* treatments are both shown at 10 seconds since page load.

the participant which sports they like to play and provides many options. However, within the question’s directions (i.e. the “fine print”), we tell the respondent to ignore the question prompt and instead click on the title above (see Fig 3.2). If the participant clicked the title, they “pass” the IMC; any other behavior is considered failure. We administer the IMC and consider it to be a proxy for satisficing behavior in general<sup>14</sup> which we were able to compare across the four treatments.

After this point, there are no longer any differences in how we present survey

<sup>14</sup>Oppenheimer et al. (2009) could not detect a significant difference between the “fancy resort” and the “run-down grocery store” conditions in question A using the full sample, *but* could detect a significant difference using data only from the participants who passed the IMC. This is an intuitive result; the psychological effect that occurs due to subtle word changes would only be detectable if a respondent carefully read the instructions.



Figure 3.2: A screenshot of the *instructional manipulation check* in the *Control* treatment.

questions across the treatments. We turn off the *Kapcha* fade-in, the *Timing control*, and the *Exhortation* message. This allows us to collect demographics and need for cognition measures in a way that is comparable and independent of treatment.<sup>15</sup>

The next eight questions collect demographic information. We ask for birth year, gender, and level of education. We then ask a few questions about their general work habits on MTurk: “Why do you complete tasks in Mechanical Turk?”, “How much do you earn per week on Mechanical Turk?”, “How much time do you spend per week on Mechanical Turk?”<sup>16</sup>, and “Do you generally multi-task while doing HITs?” To see if people who take surveys more often are any different, we also ask: “What percent of your time on MTurk do you spend answering surveys, polls, or questionnaires?”.

After the eight demographic questions, we administer an 18-question abbreviated version of the full “Need for Cognition” (NFC) scale (see Cacioppo et al., 1984). Respondents say how characteristic each of the statements are of their personality (e.g.

<sup>15</sup>Naturally, the influence of the treatment from the first three questions may linger.

<sup>16</sup>This question was used in Ipeirotis (2010), which presents result from a large MTurk survey

“I find satisfaction in deliberating hard and for long hours”) on a five-point scale where 1 indicates “extremely uncharacteristic” and 5 indicates “extremely characteristic”.<sup>17</sup> In short, the NFC scale assesses how much an individual has a need to think meticulously and abstractly. This is of interest to our study because an individual’s NFC may itself affect the probability of satisficing during a survey. People who have a higher NFC are also more likely to fill out the survey diligently and appear with high scores. Those with low NFC will have scores that are attenuated toward the center (i.e., 3) because they are more likely to haphazardly guess.

For the 30th question, we ask the participant to rate how motivated they were to take this survey on a nine-point scale.

We then give two *optional* feedback questions. On both questions, we indicated that responses would be given either a \$0.01, \$0.05 USD bonus, or no bonus (these three levels were randomized in order to test the effect of bonus level on feedback quality). The amount of feedback is also used as a measure of respondent engagement.

The first question inquired, “What did you like most about this survey? What did you like least about this survey? Is there anything you would recommend to make it better?” The second feedback prompt was only relevant for the *Kapcha* or *Timing control* treatments. We asked, “Certain respondents had to wait for each survey question to complete before filling in answers. We are especially interested in knowing how this affected the way you took the survey.”

### 3.2.4 Other Data Collected

During our recruitment period, we posted HIT bunches (see section 3.2.1 for details) with an equal number of tasks in each of the four experimental treatments.

In addition to the participants’ responses, we recorded how long survey respon-

---

<sup>17</sup>Approximately half of the questions are also reverse-coded so that noisy survey responses would cancel themselves out and tend toward the center.

dents spent on each part of the survey. This gave us some indication of how seriously people took our survey (i.e., read instructions, considered answers to questions). We also recorded when the participant’s task window was focused on our task or focused on another window.<sup>18</sup>

In the future, we hope to collect much more detailed information on the user’s activity including timestamps of exact mouse position locations, mouse clicks, and keystrokes. Ultimately, it would be an asset to researchers to be able to “playback” the task by watching the worker’s mouse movements in a short video in order to gain greater insight into how respondents answer surveys.<sup>19</sup> This would help identify satisficing behavior in a way that would go undetected using other rigid rules, but would be obvious from watching a video (e.g., instantaneously and haphazardly clicking on random answer choices).

### 3.3 Results

Table 3.2 shows the main results of the experiment. In short, we find that the *Kapcha* treatment increases the proportion of respondents that pass the instructional manipulation check (the *IMC pass rate*) relative to other treatments but causes more people to leave the task midway. We find the *Kapcha* treatment induces a highly significant effect on question A, but not on question B. Overall, we confirm both of Thaler (1985)’s economic effects if we combine all treatments.

---

<sup>18</sup>Unfortunately, this variable was not compatible with all Internet browsers and was too noisy to use in analysis.

<sup>19</sup>Everything mentioned here is possible by using <http://clicktale.com>’s premium service.

	Over- all	<i>Con- trol</i>	<i>Exhort- ation</i>	<i>Tim- ing</i>	<i>Kap- cha</i>
N	784	178	208	210	188
Attrition (%)	7.3	5.6	2.4	8.1	13.3
IMC Pass Rate (%)	81.7	77.4	76.4	83.9	90.2

**Question A price (\$)**

“fancy”	2.21	2.14	2.17	2.27	2.23
“rundown”	1.97	2.06	2.10	1.96	1.69
difference	0.24**	0.08	0.07	0.31	0.54***

**Question B intent**

“paid”	7.28	7.55	7.31	7.14	7.16
“free”	6.91	6.79	7.24	6.35	7.12
difference	0.37*	0.76	0.07	0.79*	0.04

\*p < 0.05, \*\*p < 0.01, \*\*\*p < 0.001

Table 3.2: Summary statistics by treatment

### 3.3.1 Timed Treatments Lead to More Attrition

Table 3.3 shows the observed attrition for each treatment as well as comparisons against the *Control*.<sup>20</sup> Our two timed treatments *timing control* and *Kapcha* led to higher attrition as compared with the *Control* treatment. This result is not surprising since by forcing some respondents to spend more time on our survey, we effectively are lowering their hourly wage and testing their patience. As we might expect, the *Exhortation* treatment, which reminds the respondent of the importance of our survey, slightly lowers attrition (though not significantly).<sup>21</sup>

Treatment	N	Attrition (%)	comparison with <i>Control</i> treatment (p-value)
<i>Control</i>	178	5.6	—
<i>Exhortation</i>	208	2.4	0.12
<i>Timing control</i>	210	8.1	0.46
<i>Kapcha</i>	188	13.3	0.02
All	784	7.3	—

Table 3.3: Attrition by treatment

Ordinarily, survey designers seek to minimize attrition (i.e., maximize completion rates of their surveys). However, in the MTurk environment, where the number of potential respondents is larger than the desired sample size, the researcher may want to restrict the sample to those who yield the highest quality data.<sup>22</sup> If the decision

<sup>20</sup>We employ two-sample two-sided z-tests for difference in proportion.

<sup>21</sup>Assuming that the true difference in the proportion of attrition between the *Control* and *Exhortation* treatment was 3.4%, we would need 620 observations in both treatments to have an 80% chance of detecting it.

<sup>22</sup>In many survey situations such as surveying current members of an organization, maximizing

to leave a survey midway through indicates that these people are “less serious”, we probably would not want them in our sample.<sup>23</sup>

Note that going forward, we only analyze tasks that were fully completed (i.e. workers who did not attrit).

### 3.3.2 Kapcha Alone is a Successful Mechanism for Reducing Satisficing

We investigate the IMC pass rate by experimental treatment and demographic controls.

Treatment	IMC Pass Rate (%)	comparisons (p-value)		
		<i>Exhortation</i>	<i>Timing</i>	<i>Kapcha</i>
<i>Control</i>	77.4	0.734	0.143	0.002**
<i>Exhortation</i>	76.4		0.057	< 0.001***
<i>Timing control</i>	83.9			0.076
<i>Kapcha</i>	90.2			

Table 3.4: IMC pass rate (%) by treatments with comparisons

Table 3.4 illustrates that the *Control* and *Exhortation* treatments differ significantly from the *Kapcha* treatment.<sup>24</sup> The difference between the *Timing Control* and *Kapcha* was almost significant ( $p = 0.076$ ). We suspect this difference is real but we most likely do not have enough data to detect it.<sup>25</sup>

Table 3.5 demonstrates that the only experimental treatment which significantly impacts the IMC pass rate is the *Kapcha* fade-in treatment, increasing the pass rate the response rate is a good strategy. However, in academic research, especially behavioral economics and psychology, weeding out non-serious respondents may be desirable.

<sup>23</sup>Oppenheimer et al. (2009) discusses how excluding data based on whether people fail the instructional manipulation check may lead to a non-representative population. This is a concern that should be considered by the researcher, but is unlikely to be relevant unless the non-representative

	without	with
( $N = 727$ )	controls b (se)	controls b (se)
<b>Treatment</b>		
<i>Exhortation</i>	-1.0 (4.4)	-1.0 (4.4)
<i>Timing Control</i>	6.6 (4.2)	7.3 (4.2)
<i>Kapcha</i>	12.8** (4.0)	13.0*** (3.9)
Gender (male)		-7.7* (3.1)
Age (26-35)		11.4** (4.0)
Age (36-45)		16.3*** (4.3)
Age (over 45)		17.1*** (4.6)
Completed college		4.2 (2.9)
Reported motivation		1.9 (1.1)
# words in feedback		0.3*** (0.1)
Need for cognition		3.8 (2.5)
Break for $\geq 2$ min		-13.1 (7.7)
Other covariates		✓
Intercept	77.4*** (3.2)	27.1 (14.3)
$R^2$	0.020	0.165

\*p < 0.05, \*\*p < 0.01, \*\*\*p < 0.001

Table 3.5: IMC pass rate (in %) explained by treatment and other covariates



by  $12.8 \pm 4.0\%$  ( $p < 0.01$ ). Controlling for demographics<sup>26</sup> makes the effect more statistically significant while leaving the estimate unchanged ( $p < 0.001$ ). As expected, controlling for demographics also significantly improves the overall fit of the model, as measured by  $R^2$ . Demographic factors that affect the IMC pass rate are discussed in section 3.3.4.

Are the higher IMC pass rates in those groups simply the result of “less serious” respondents removing themselves from our sample? We investigate whether the higher attrition rates in the *Kapcha* treatment can explain the differences in IMC pass rates we have been exploring.

Under the most conservative assumptions, we assume that all of the additional workers who left the *Kapcha* would have stayed and subsequently failed the IMC. More specifically, we assume that the differential attrition between the *Kapcha* and the *Control* treatments ( $13.3\% - 5.6\% = 7.7\%$ ) stayed and fail the IMC ( $188 \times 7.7\% = 14$  new failing workers). This lower IMC pass rate would become 83.05% which is greater than the 77.4% pass rate of the *Control* ( $p = 0.09$ , one-tailed two-sample z-test). Under these conservative assumptions, attrition can only explain 5.65% of the total 12.8% effect, or 44% of *Kapcha*’s success. It is reasonable to assume that the *Kapcha* adds an additional boost beyond merely annoying people until they leave.

---

subset of workers would bias the study.

<sup>24</sup>Via two-sample, two-tailed z-tests

<sup>25</sup>If the true proportions were equal to the means that we observed, we would need 492 observations in each treatment to have an 80% chance of detecting it.

<sup>26</sup>The other covariates in table 3.5 represent covariates that were not significant and jointly, barely significant. These include frequency of survey-taking on MTurk, hours per week on MTurk, earning per week on MTurk, task day of week, task hour of day, reported multitasking behavior, minutes since HIT was listed, and average time spent on first 30 questions.

### 3.3.3 Finding Larger Effects in the Economic Behavior Questions

Assuming that the subtle word changes from questions A and B cause real differences, we hypothesize that the largest effects in both questions will be found within the *Kapcha* treatment which is designed to force people to pay close attention to the words in the question text.

Tables 3.6 and 3.7 show the effect of subtle word changes on the price subjects pay for a soda when it comes from a “fancy resort” and the increased intent to go to a game whose tickets were “paid handsomely” for as opposed to received for free.

	Overall	<i>Control</i>	<i>Exhortation</i>	<i>Timing control</i>	<i>Kapcha</i>
<b>No controls</b>					
“fancy” b (se)	0.239** (0.086)	0.080 (0.173)	0.072 (0.176)	0.303 (0.176)	0.543*** (0.148)
$R^2$	0.011	0.001	0.001	0.015	0.077
<b>With controls<sup>a</sup></b>					
“fancy” b (se)	0.249** (0.089)	0.143 (0.180)	-0.021 (0.184)	0.319 (0.180)	0.533** (0.182)
$R^2$	0.083	0.210	0.185	0.254	0.205
$N^b$	714	163	201	190	160

\*p < 0.05, \*\*p < 0.01, \*\*\*p < 0.001

<sup>a</sup> Includes same controls as table 3.5

<sup>b</sup> We excluded 13 prices that were not numbers between \$0 and \$10

Table 3.6: Question A: Increase in willingness to pay for a soda due to subtle word changes involving whether source of soda was a fancy resort or run-down grocery store (with and without other controls)

Overall, workers would pay \$0.24 more for sodas bought at a “fancy resort” over a “run-down grocery store” ( $p < 0.01$ ). As expected, in our *Timing control* and *Kapcha* treatments, this effect is stronger than average and is largest and highly statistically significant in the *Kapcha* treatment. Controlling for demographics leaves this result unchanged, but substantially improves the overall fit of our model.

Overall, workers who paid for the football ticket were more likely to go than workers who received the ticket for free with a difference of 0.37 intention units on a nine-point scale ( $p < 0.05$ ). The effect was not robust when controlling for demographic variables.

The effect of questions B’s phrasal change is estimated to be between 0.29 and 0.37 depending on whether we control for demographics. In no individual treatment are there significant differences both with and without control variables. That said, the *Timing control* and the *Control* treatment appear to have a larger effect. However, these results are barely significant and vacillate upon the introduction or removal of the demographic controls. We consider these effects to be spurious and conclude that the standard errors in each treatment are too large (approximately 0.31 to 0.44 depending on the treatment) relative to the hypothesized effect size. Most likely, there was not enough data to detect the small effects given the considerable spread in responses.

The analysis of question A lends legitimacy to the *Kapcha* treatment’s power to reduce satisficing. However, we were unable to draw conclusions from the responses to question B. We again note that both investigations were underpowered. We hope to get more data in the future so we can use the effects found in questions A and B to proxy for satisficing behavior.

### 3.3.4 Observations on the MTurk Survey-taking Population

#### MTurkers Beat Stanford and NYU Students

We compare the IMC pass rates from Oppenheimer et al. (2009) with our data. In Oppenheimer et al. (2009), using  $n = 213$  New York University undergraduates, the IMC pass rate was 54%, which is lower than our *Control* group ( $n = 167$ ) with a pass rate of 77.3% ( $p < 0.001$ ). This *Control* treatment pass rate is similar to the 82.5% pass rate in Oppenheimer et al. (2009) during administration of a paper and pencil exam using  $n = 336$  Stanford university undergraduates who were believed to be “motivated” (because they were interested in either a major or minor in psychology).

#### Demographic and Behavioral Drivers

Which demographic groups paid the closest attention to our survey? In table 3.5, we find that women on average pass the IMC  $7.7 \pm 3.1\%$  more often than men ( $p < 0.05$ ). We find that older workers do better than younger workers; 26–35 year olds pass  $11.4 \pm 4.0\%$  more often ( $p < 0.01$ ); 36–45 year olds pass  $16.3 \pm 4.3\%$  more often ( $p < 0.001$ ); and workers over 45 years of age pass  $17.1 \pm 4.6\%$  more often ( $p < 0.001$ ). We also find that workers who completed college pass 4.2% more often (this result was nearly significant).

Two of our variables which measure respondent engagement, the NFC and self-reported motivation, are not significant when included together and with the number of words in feedback. However, these variables are both highly significant (when included only with the indicator variables for treatments).

Surprisingly, the worker’s average number of hours worked on MTurk per week, the average earnings on MTurk per week, the reported level of multitasking, nor the frequency of survey-related tasks were significant in predicting IMC pass rate.

## Feedback

The final significant relationship found was the number of words written as feedback (question #31). We find that for each word of additional feedback, the probability of passing increased by  $0.3 \pm 0.1\%$ . A one standard deviation change in the number of words of feedback (equal to 28.2 words) is associated with an 8.5% increase in the IMC pass rate even controlling for other measures of engagement. Therefore, the length of a free response can be used as a proxy for survey engagement, a result also reported by Bush and Parasuraman (1984).

We also did a small experiment studying how to incentivize feedback. We varied how much we offered respondents for providing feedback (offering either one, zero or five cents). The average feedback in the group without a bonus is 28.3 words. Compared with an unpaid bonus, paying a one cent reward garners 5.0 more words on average ( $p < 0.05$ ) and the five cent bonus garners 7.6 more words ( $p < 0.01$ ). Further, we could not reject the hypothesis that the two effects were equal ( $p = 0.364$ ). This indicates that paying a minimum bonus of one cent elicits almost as lengthy feedback as paying almost five times that much (and roughly half the value of the full HIT). Interestingly, although we expected the *Exhortation* group to provide more feedback since they were reminded that they were participating in the study, neither that group nor any other treatments received significantly more feedback (although the groups were jointly significant at the  $p < .05$  level).

Furthermore, the workers are eager to give feedback. Researchers can rapidly pilot their studies and get real-time feedback on how they are perceived by survey-takers.

Asking feedback also gave us a wealth of insight into how survey respondents perceived our various treatments including the *Kapcha*. One danger of setting the reading speed too slow for fast readers was illustrated by this worker from Colorado Springs, CO:<sup>27</sup>

---

<sup>27</sup>We recorded each worker's IP address which allowed us to determine their location.

“Text needs to be instantaneous. No apparent reason for it to appear slowly other than to aggravate the participant.”

However, this comment from another worker in Detroit, MI illustrates the intended purpose:

“I didn’t enjoy the way the words scrolled slowly, as I read fast, but in its defense the slow scrolling words lead me to pay closer attention to what I was reading and skim less.”

In addition, many workers are survey-savvy and eager to offer design suggestions such as this worker from San Bernardino, CA who is also familiar with Likert scales:

“I liked the situational question about the soda cost... I do not like the black background color, it hurts my eyes when contrasted with the white. Would prefer a 7 point likert scale if possible.”

### 3.4 Discussion

The main goal of our study is to investigate a survey platform that reduces satisficing across the board. We propose the idea of *Kapcha*, a method which involves slowing people down by fading-in the question text, thereby accentuating each word. We have found evidence that *Kapcha* has the potential to reduce satisficing in online surveys. We then open-source the platform (see Appendix 3.5) so that the surveyor can simply “plugin” the platform and be confident of obtaining more accurate results.

MTurk workers that participated in a survey task employing the *Kapcha* passed an *instructional manipulation check* about 13% more often than those who were given a standard survey and it is reasonable to assume that this pass rate can be used as

	Overall	<i>Control</i>	<i>Exhortation</i>	<i>Timing control</i>	<i>Kapcha</i>
<b>No controls</b>					
“paid”	0.370*	0.716	0.066	0.797*	0.039
b (se)	(0.176)	(0.365)	(0.314)	(0.376)	(0.381)
$R^2$	0.006	0.024	0.000	0.024	0.000
<b>With controls<sup>a</sup></b>					
“paid”	0.285	0.981*	0.150	0.726	-0.026
b (se)	(0.177)	(0.443)	(0.330)	(0.385)	(0.378)
$R^2$	0.066	0.203	0.162	0.256	0.322
$N$	727	168	203	193	163

\*p < 0.05, \*\*p < 0.01, \*\*\*p < 0.001

<sup>a</sup> Includes same controls as table 3.5

Table 3.7: Question B: Increase in intention to attend the football game due to subtle word changes involving whether the participant paid for or received a ticket for free (with and without other controls)

a proxy for general satisficing behavior. At most, only 44% of this effect can be explained by a higher proportion of people leaving the *Kapcha* survey task.

The treatment where we merely exhorted the participant to pay more attention had no significant effect on satisficing. The treatment that imposed a waiting period but did not accentuate the words, did better than the standard survey group, but the difference was not significant.

Upon analyzing demographic data, we find the segment of workers least likely to satisfice are females over the age of 26 who leave thoughtful feedback.

We must also emphasize that the trick question was very difficult and requires

carefully reading the fine print.<sup>28</sup> As a testament to the quality of work on MTurk, we find it absolutely incredible that even in the *Control* treatment, people pass the trick question with such high proportion.

### 3.5 Future directions

Our study indicates that using *Kapcha* can significantly increase the amount of attention respondents give to reading directions and answering questions. For future research, we would like to study how the *Kapcha* is affected by other variables such as levels of motivation or monetary incentives as well as among different populations. We would also like to conduct further experiment with how the *Kapcha* could be optimized using principles of psychology and perception so as to draw the attention of respondents. Finally, we would like to design a survey task that is deliberately designed to impose a high cognitive burden and cause respondents to satisfice. Testing the *Kapcha* under these circumstances will provide a clearer picture of its power.

#### **Kapcha may be moderated by other variables**

For one reason or another, the *Kapcha* may be more effective on certain populations. For instance, the degree to which a respondent pays closer attention when being forced to wait, or when text is faded in, may differ by language or culture. A study drawing participants from various countries may elucidate its differential effectiveness.

Apart from interactions with demographic variables, the effectiveness of the *Kapcha* may vary with monetary or non-monetary incentives.

For example, can you pay people to pay more attention? If people are paid higher monetary awards, does that reduce the advantages of using a *Kapcha*? It could be that incentives simply cannot induce people to pay more attention beyond a certain

---

<sup>28</sup>One of the authors gave it to colleagues in their department and each of them failed.



point and that the only way to increase attention to the highest levels is through attention-grabbing techniques.

Though not statistically significant, it appears that telling respondents that their answers will be used for research (our *Exhortation* treatment) motivates people to complete our survey at higher rates. This is most likely because reminding them about the survey’s research value imbues the survey with a sense of meaning (a similar result was found in Chandler and Kapelner, 2013 which is also Chapter 2 of this document). Therefore, it may be wise to insert an exhortative statement into the *Kapcha* to get an “added boost.”

### **The Kapcha appearance should be optimized**

Further, we would like to experiment with the particulars of the *Kapcha* presentation. In our experiment, we used white text on a black background and faded the words in at 250 words per minute. Does the choice of color schemes and text font matter?<sup>29</sup> What is the speed at which words should fade-in to optimize attention to our survey?

Research suggests that forcing a person to direct their gaze toward an area correlates highly with the attention they pay to that area (Hoffman and Subramaniam, 1995) and the psychology of perception is ripe with many other examples of how color, contrast, and movement could be used to draw attention.

We must admit that many of our faster reading respondents in the *Kapcha* group expressed that they did not like the unfamiliar method of fading-in survey questions (34 of 163 respondents left negative feedback compared with 7 positive feedbacks). Although it had its desired effect of increasing the attention people paid to the survey questions, we certainly would like to further calibrate the *Kapcha* so as to slow respondents down without annoying them.

---

<sup>29</sup>Many respondents complained that the white-on-black background was distracting which may have affected how well the *Kapcha* worked.

## **We propose a survey task that measures satisficing more generally**

The IMC and the behavioral questions are both noisy and incomplete measures of satisficing. Using only these response variables as proxies for satisficing is a weakness in our present study.

We propose to create a survey task that has several measures of satisficing in order to demonstrate that the *Kapcha* can reduce satisficing in the broadest context.

We will review the findings from the literature of survey response psychology (e.g. Krosnick et al., 1996, Krosnick, 1999, Tourangeau et al., 2000) which provide guidance for how to design surveys to minimize participant satisficing. Using these principles, we will reverse-engineer a survey that is *deliberately* constructed so that respondents are *likely* to satisfice. We will then see how well the *Kapcha* prevents satisficing even under the most difficult of circumstances.

To offer an example, Krosnick (2000) provides a framework for how respondents satisfice depending on the *structure of questions*, the survey’s *difficulty*, the *respondent’s ability*, and the *respondent’s motivation*. Three commonly cited examples of satisficing due to question structure are *response order effects* whereby people choose the first answer of surveys, *no opinion filters* whereby people who are lazy will sooner choose “no opinion” than take the time to think of what their opinion is, and *acquiescence bias*. where respondents are more likely to choose “agree” if the choices are “agree or disagree”. The difficulty of a survey can be related to the “readability” of the survey questions (higher readability implies shorter question length and basic vocabulary (Chall and Dale, 1995)). The respondent’s motivation may be related to how meaningful they perceive the task to be. Presumably, higher ability respondents and respondents who are motivated would also tend to satisfice less.

If the *Kapcha* is found to be effective here, we will be confident that the *Kapcha* method prevents satisficing under very general conditions.

## Data Sharing

We cross-validated some of the self-reported demographic information using data provided by Panos Ipeirotis from Ipeirotis (2010). 23 people who reported their age in our survey also reported their date of birth in Ipeirotis (2010)’s survey. In all but one case, the age and date of birth were consistent. This offers evidence that even over a time period of more than 6 months, time invariant demographic data can be reliably collected on separate occasions. Broadly speaking, MTurk workers seem to be honest in sharing their personal information.

Data sharing among academics using MTurk provides not only the possibility of validating data, but also of using demographics or other covariates from one study as controls in others. For example, in our present study, we evaluated respondent’s Need for Cognition which could be a useful control variable in other studies. In many cases, it may be highly useful to match demographic and other behavioral characteristics as a way to increase precision without using the limited time of respondents. Using data from other studies is especially beneficial in the case of natural field experiments where the researcher will not want insinuate that the task is an experiment.

We propose that researchers agree on a central, shared repository of data related to the MTurk workers and offer an API for easy access.

## TurkSurveyor: An Open-Source Experimental Platform

We would like to introduce “TurkSurveyor”, an open-source experimental system designed for running surveys (or survey-based experiments) on Amazon’s Mechanical Turk. TurkSurveyor is written in a mixture of Ruby (on Rails), HTML, CSS, and Javascript and is available under the MIT license and includes an instruction manual at <http://code.google.com/p/turksurveyor/>. The goal of its development is to have a simple push-button system which allows one, with a minimum of customiza-

tion, to use MTurk to collect data for a custom survey.

## **Replication**

At <http://danachandler.com/kapchastudy.html>, you can find the source code, the raw data, and the analysis used to run this study.

## **Acknowledgments**

The authors wish to thank Larry Brown, Persi Diaconis, David Gross, Susan Holmes, Daan Struyven, Nils Wernerfelt, and Adi Wyner for helpful discussion and comments. Both authors also acknowledge support from the National Science Foundation in the form of Graduate Research Fellowships.

## Estimating the Number of Objects in Images

By Susan Holmes and Adam Kapelner

### Abstract

We develop a method to estimate the count of objects or features in an image such as tallying the number of birds in a photograph or the the number of cells in a microscopic image. Our approach has two novel steps. We first develop software that records the labelings of many naive workers via Amazon’s Mechanical Turk. We then view each worker’s training as a “capture” in a set of capture-recapture experiments. We use statistical learning to eliminate falsely-trained objects and then take a Bayesian approach and use a Gibbs sampler to estimate the true number of objects.

### 4.1 Introduction

Counting and locating specific features in an image has many applications such as cell counting, harvest estimation, and crowd counting. Standard methods in counting (how many words did Shakespeare know? (Efron and Thisted, 1976) and animal abundance) use models where the frequencies of counts form sufficient statistics. How-

ever, with large amounts of noisy data, such models have to be refined. In this paper, we develop a two-stage solution to the counting and localization of objects in images. Some images feature objects that are easy to locate, thus the count is obvious; other images are difficult featuring ambiguous and obfuscated objects, this is where our solution can be applied.

In section 4.2 we give an introduction to object identification in the field of computer vision and our previous research that served as the inspiration for this project. In section 4.3 we talk about the engineering effort of our new tool, **DistributeEyes**, that uses crowd labor to label images. The use of this new tool has many applications, but we focus here on improving the estimate of the number of objects in an image. In section 4.4 we talk about statistical methods that use data from **DistributeEyes** to estimate the number of objects in images using Bayesian capture-recapture methods. In section 4.5 we run an experiment with a wide variety of images, and test our count estimation method on the experimental data. We are also able to present a quality control analysis and present observations about spatial and temporal patterns in workers’ training points. We conclude and talk about future directions in the final section 4.6.

## 4.2 Crowdsourcing Object Identification in Images

Many automated computer vision algorithms are available and have been tailored for specific types of images and object identification questions. Cell recognition techniques applied to microscopy images are transforming the field of pathology (Kapelner et al., 2007b), thermal/aerial photographs can be analyzed for car detection and counts used in traffic analysis.

Projects like LabelMe (Russell et al., 2008) provide annotation for objects in an image but have concentrated on images where there are less than 15 objects, most

are of different types. In contrast, we are interested in counting and locating many objects of similar types; in general, there will be more than 50 objects of each type which need to be counted. In our implementation, we ask the participants to identify the object by clicking on one central pixel, instead of providing a full outline of the object, or textual annotation.

Our previous research (Holmes et al., 2009; Kapelner et al., 2007b) yielded an interactive statistical learning software named “**GemIdent**” (Kapelner et al., 2007a) which was designed to locate cells in histological samples. The program can be adapted to any object recognition task on images with few colors and relative conformity in the shape and size of the sought objects. **GemIdent** was instrumental in the localization and identification of T-cells and cancer cells in immunohistologically stained microscope images enabling an architectural comparative analysis in tumor-draining lymph nodes from breast cancer patients and healthy lymph nodes (Setiadi et al., 2010). The software package has been open-sourced and available for download since May, 2007. We have received much feedback about its shortcomings. Overwhelmingly, users feel that the training step for the different object instances is too time-consuming. We were inspired to address this weakness by introducing a new possibility for incorporating training data — from a *crowd* of inexperienced trainers, located around the world, who assist in identifying objects of interest. “‘Crowd-sourcing’ is the act of outsourcing tasks, traditionally performed by an employee or contractor, to an undefined, large group of people or community (a crowd), through an open call”.<sup>1</sup>

Why charge anonymous laborers with such a task? General automatic object identification using statistical learning is difficult in cases where the contexts are heterogeneous but the objects are not. Manually marking the objects would yield the most accurate solution, but would be too time-consuming in real applications such

---

<sup>1</sup><http://en.wikipedia.org/wiki/Crowdsourcing>

as cell counting where there could be millions of cells in a single histological sample.

Our solution is to provide *training data* through a set of points collected via Internet-based crowdsourcing. We facilitate the distribution of the task of marking the objects. We thus extended our image segmentation software and names it “DistributeEyes”<sup>2</sup>. It is an HTML application that plugs into a crowdsourcing platform, currently Amazon’s Mechanical Turk<sup>3</sup> and communicates with a Gemident-enhanced client. We give the details of the integration of these platforms in the following section.

### 4.3 Engineering

The data acquisition solution has several different components which communicate with each other for an integrated solution (see figure 4.1).

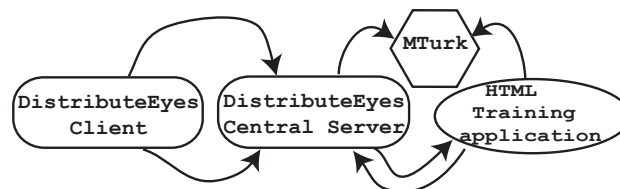


Figure 4.1: The DistributeEyes Components and their interactions

---

<sup>2</sup><http://www.distributeeyes.com>

<sup>3</sup><http://www.mturk.com>



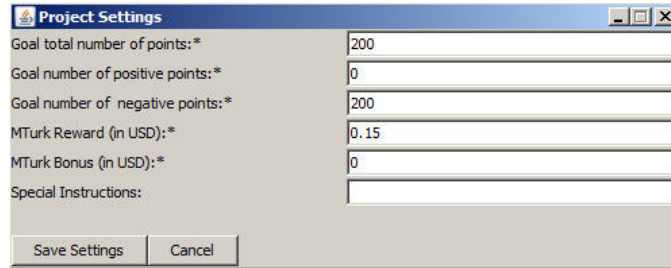


Figure 4.2: DistributeEyes Project Settings Dialog Window

### 4.3.1 The DistributeEyes client

A user of the client (the “user”) creates a new project and identifies the different objects of interest types (for historical reasons, we call these “phenotypes”) and then provides a few examples of each. The user then initializes the project on the central **DistributeEyes** server (the “server”). Then, images that we would like to extract training data from (“images”) are added to the project.

The “Distribute Images” panel (not shown) shows each image as a row in a table and allows users to select which images they want to distribute for online marking. The marking of one image is done as a “Human Intelligence Task” (“HIT”) on Amazon’s MTurk by a worker and is dubbed a “submission.”

A project settings panel allows customization of the following parameters: minimum number of total points, minimum number of positive points (objects) sought, minimum number of negative points (non-objects) sought, monetary reward for training, bonus reward for training, time-to-expire for the HIT, and special instructions.

After images are distributed, the client polls the server regularly to see if the HITs have been completed. Once a submission is detected, the client provides two means of assessing the quality of the work. The first is a window that shows the local area around each marked point which allows for quick deletions of simple mistakes (see figure 4.3). The second is the phenotype training window which displays the original

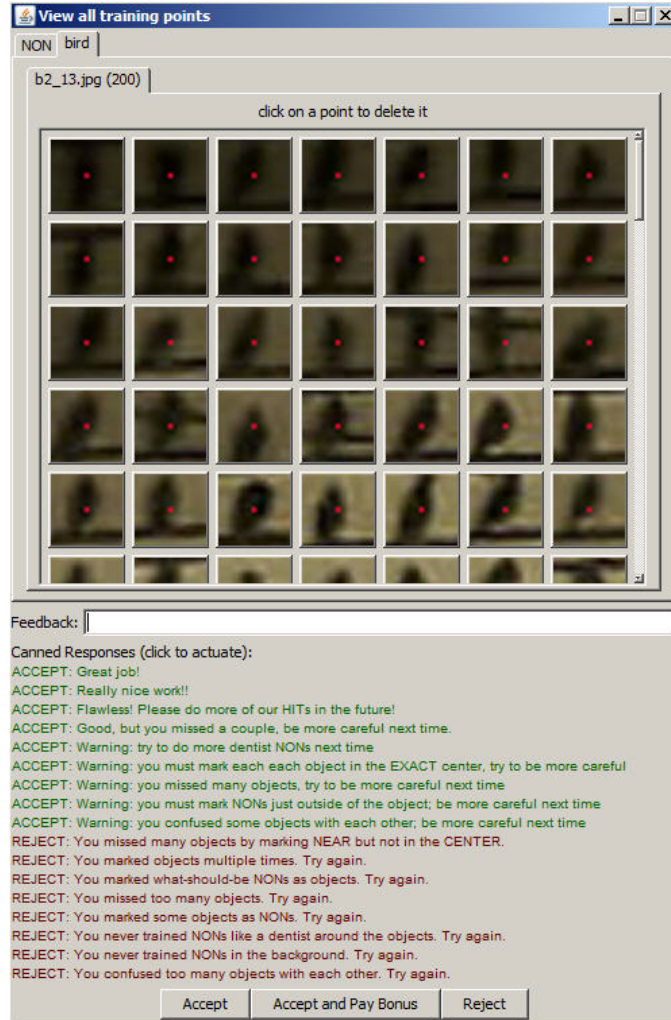


Figure 4.3: Worker submission check window

image with an overlay of the training points (not shown, see Holmes et al., 2009 Figure 8, p9).

Usually within 5-10 seconds the user can recognize if the worker did a satisfactory job and accept or reject accordingly. A rejection means zero compensation for the worker. The client also provides common canned feedback responses such as “Great job!” or “You missed too many objects. Try again.” If the worker went above and beyond, a bonus can be rewarded. As an additional tool, the worker’s historical record (acceptances, rejections, bonuses, and feedback) are displayed.

It is this speed-up of the training process which makes **DistributeEyes** a powerful

research tool with enormous potential.

### 4.3.2 The Central DistributeEyes Server

The function of the server is to be an intermediary for passing data between the client and MTurk’s workers, to render the `HTML` training application, and to serve the public homepage URL.<sup>4</sup> User accounts are created via the public portal and after email verification, the user can download the `DistributeEyes` client.

### 4.3.3 The `HTML` Training Application and the Raw Data

Each image can be rendered inside of a training application for marking by a worker. The training application comes complete with each example image of each phenotype, a magnified view of the image where the user marks / deletes points, magnification adjustments, a thumbnail, a scaled-to-fit view, and a help link.

Before the worker is allowed to use the application, he must “qualify” by watching a five-minute tutorial video which teaches training concepts and techniques, pass a six-question quiz (see figure 4.4), and agree that `DistributeEyes` can monitor his usage. Workers must pass the quiz in order to be able to work on a HIT. The worker is forced to watch the entire video before having the opportunity to answer the questions. Workers that pass the qualification quiz are recorded on the central server and no longer need to re-qualify to work on future HITs.

Then, the worker moves to the training application (see figure 4.5). The worker marks points via point-and-click until all goals set by the user are met and all special instructions are fulfilled. The “submit training” button becomes enabled, and the worker can upload his points to the central server. When the worker submits, he is led through a series of dialogs inquiring if he has made the common mistakes that

---

<sup>4</sup>The server was written in Ruby on Rails using MySQL for the database.



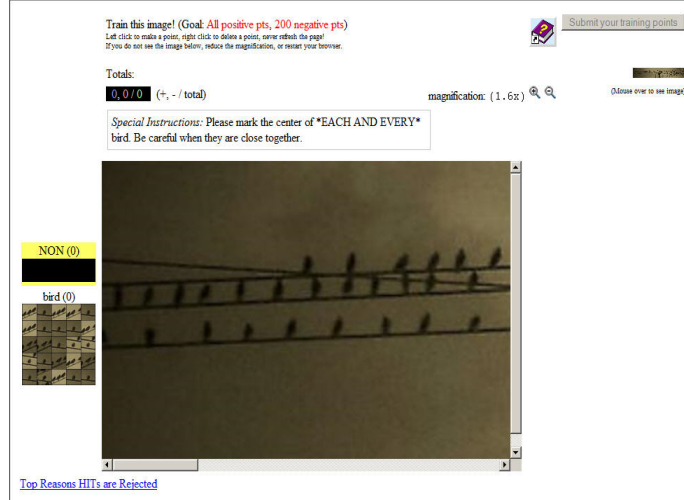


Figure 4.5: The HTML training application embedded into MTurk. The worker is training the project “birds” as part of our experiment (see section 4.5).

#### 4.3.4 Interfacing with MTurk

Amazon Mechanical Turk (MTurk) is a marketplace that coordinates the use of human intelligence to perform tasks which computers are unable to do. This labor paradigm has been coined “crowdsourcing” which Amazon whimsically refers to as “artificial, artificial intelligence.”

Since it is the largest such crowdsourcing marketplace on the Internet, and it provides a convenient Applications Programming Interface, it was chosen as the venue to host **DistributeEyes** tasks. To create an MTurk HIT, the HTML Training Application is rendered inside an IFrame which is wrapped inside of MTurk’s worker interface. From the perspective of the worker, everything occurs within MTurk; he is completely unaware of **DistributeEyes**.

After submission, the client will detect the completion within 10 minutes. Acceptances or rejections are relayed from the client to MTurk via the central server using the MTurk API. The server records each acceptance or rejection. The workers are warned upon rejections. If a worker receives many rejections, he may be banned from working on **DistributeEyes** HITs.

## 4.4 Statistical Model and Implementation

In many instances pooling together work done by many naive workers can have greater quality than an expert. Suppose we have one type of object (one phenotype) we would like to count (*e.g.* faces in a photograph of a crowd or cell nuclei in a microscopic field-of-view) and suppose we have  $W$  workers who submit training coordinates for where the objects are located. Our goal is to combine all  $W$  trainings together to create  $\hat{N}$ , an estimate for  $N$ , the total number of objects.<sup>5</sup>

To first approximation, we can view each of the  $W$  trainings as “captures” of different objects and all trainings together as a *capture-recapture* experiment in a closed population: that of the actual objects.

### 4.4.1 Capture-Recapture Models

Capture-Recapture experiments trace their history back to estimating the number of animals in a certain locality. The number of animals caught in a particular sample  $n$  may give little information about  $N$ , the total population size, considered to be static. Hence, many captures are done where all animals in a capture are marked via a unique identifier and released. In subsequent capture trials, the number of animals previously marked is noted. It is for this reason that this procedure is sometimes known as a discrete-time *capture-mark-recapture* experiment. For a more thorough introduction we refer the reader to the beginning of Otis et al. (1978) and Robert (2007, Chapter 5.1 and 5.2).

Methods for combining the data from the different captures to provide an estimate for  $N$  can vary based on the assumptions one makes about the heterogeneity of the animals and the way they are captured. Otis et al. (1978) notes three types of

---

<sup>5</sup>We also use their most likely locations in practical applications, but this is not addressed in this investigation.

heterogeneity:

- $t$  The probability of capturing an animal varies with the time of the capture, that is it varies trial-trial within the capture-recapture experiment.
- $b$  The probability of capture of an animal is dependent on its behavioral response to a previous capture.
- $h$  There is heterogeneity in each of the animal’s capture probability.

Each type of heterogeneity can be combined together for a total of eight possible models.

#### 4.4.2 Object Identification as a Capture-Recapture Model

The objects in the image of interest can be thought of analogously in the Capture-Recapture paradigm as animals. The number of objects are now the closed population size. Each of the  $W$  worker labelings are now viewed as capture trials. If two different workers mark the same object, it is counted as a “recapture”.

In an arbitrary image, some objects are more difficult to detect than others, therefore heterogeneity of type  $h$  *must* be accounted for. Since the objects are pixels unchanged in an image, there cannot be any heterogeneity due to previous captures, hence heterogeneity of type  $b$  does not need to be considered. Lastly, we assume that all workers have the same ability. This may not be fully justified, but we do this to avoid extra parameterization and to keep our model simple. There is also substantial evidence that all workers can be considered the same in ability for a variety of crowdsourced tasks (for example, see Kapelner et al., 2012 which is also Chapter 5 of this document). Therefore, capture probabilities are assumed not vary with the trial number, hence heterogeneity of type  $t$  does not apply. In conclusion, we believe that the  $M_h$  model (notation being standard in the literature) with  $W$  trials is a

reasonable model that we can apply to our data in order to estimate the number of objects in images.

There is one conceptual issue that does not perfectly adhere to the  $M_h$  model. Workers are charged with the task of clicking on the objects. Sometimes, they make mistakes by clicking on a point that is not actually an object, *i.e.* they label a “false positive” (these can be likened to traps capturing “ghost” animals), an intractable problem. We first present the model without this complication below and then we address this problem in section 4.5.5 using statistical learning.

#### 4.4.3 Estimating $N$ in an $M_h$ model

Let  $X_i$  denote the number of times object  $i$  has been labeled when all  $W$  labelings of one image of interest are aggregated. Let  $p_i$  denote the probability that object  $i$  is detected by any worker (we assumed worker ability is homogenous). It is clear that the number of objects found is binomial. We further assume that the  $p_i$ ’s are drawn from a beta distribution, an assumption that is justified in section 4.5.3.

$$\begin{aligned} X_1 \mid p_1 &\sim \text{Binomial}(W, p_1), \dots, X_N \mid p_N \sim \text{Binomial}(W, p_N) \\ p_1, \dots, p_N &\stackrel{iid}{\sim} \text{Beta}(\alpha, S\alpha) \end{aligned}$$

Following the work of George and Robert (1992) and Gilks et al. (1996), we choose to estimate  $N$  via Bayesian estimation. The Bayesian paradigm fits our purposes because we have built up sufficient prior information that can be incorporated into a realistic coherent model. We now outline the prior setup and Gibbs sampling procedure used in the hierarchical model found in Yip et al. (2005).

Let  $r$  denote the number of unique objects detected when all  $W$  labelings are pooled. Denote  $M = N - r$  as the number of objects left undetected. Let  $f_1$  denote the number of objects detected only by one worker, let  $f_2$  denote the number of



objected detected by only two workers, etc, and let  $f_W$  denote the number of objects detected by all workers. We now use the following flexible conjugate priors:

$$\alpha \sim \text{Gamma}(k, \lambda), \quad S \sim \text{Gamma}(\ell, \tau), \quad M \sim \text{Geometric}(\rho)$$

Integrating out  $p_1, \dots, p_N$ , we arrive at the following Gibbs conditional sampling scheme:<sup>6,7</sup>

$$\begin{aligned} M \mid \alpha, S, r, f_1, \dots, f_W, W, \rho &\sim \text{NegBin} \left( r + 1, 1 - \rho \frac{\Gamma(\alpha + S\alpha) \Gamma(S\alpha + W)}{\Gamma(S\alpha) \Gamma(\alpha + S\alpha + W)} \right) \\ \alpha \mid M, S, r, f_1, \dots, f_W, W, k, \lambda &\propto \left( \frac{\Gamma(\alpha + S\alpha)}{\Gamma(S\alpha) \Gamma(\alpha + S\alpha + W)} \right)^{r+M} \frac{\Gamma(S\alpha + W)^M}{\Gamma(\alpha)^r} \\ &\quad \times \alpha^{k-1} e^{-\lambda\alpha} \prod_{j=1}^W (\Gamma(\alpha + j) \Gamma(S\alpha + W - j))^{f_j} \\ S \mid \alpha, M, r, f_1, \dots, f_W, W, \ell, \lambda &\propto \left( \frac{\Gamma(\alpha + S\alpha)}{\Gamma(S\alpha + \alpha) \Gamma(\alpha + S\alpha + W)} \right)^{r+M} \Gamma(S\alpha + W)^M \\ &\quad \times S^{\ell-1} e^{-\lambda\alpha} \prod_{j=1}^W \Gamma(S\alpha + W - j)^{f_j} \end{aligned}$$

$N$  can now be sampled by using the posterior samples of  $M$ , the number of undetected objects, and adding back the number of detected objects,  $r$ .

#### 4.4.4 The Gibbs Sampler Implementation

For the prior distributions of  $\alpha$  and  $S$  we pick hyperparameters to have a target mean probability of detection of 66% with large spread which is reasonable for both easy and difficult images. We aim for a target mean  $\alpha = 4$  and mean  $S\alpha = 2$  but with

---

<sup>6</sup>Note that the distributions for  $\alpha$  and  $S$  are non-standard, and further, can only be obtained up to a norming constant, therefore they are sampled via a Metropolis-Hastings acceptance or rejection.

<sup>7</sup>Note that the parameterization of the beta distribution in the model definition as  $[\alpha, S\alpha]$  is done for computational reasons to reduce the correlation of the beta's two parameters during Gibbs sampling which is a common trick.

considerable variance in order to be relatively uninformative. Therefore we choose hyperparameters:

$$\alpha \sim \text{Gamma}(k = 2, \lambda = 0.5) \quad \text{and} \quad S \sim \text{Gamma}(\ell = 1, \tau = 0.25)$$

As for the prior distribution on  $M$ , the number of undetected objects, note that the geometric distribution has one parameter and thereby is difficult to control both the mean and variance. We choose rather arbitrarily a mean  $\mu_M = 5$  ( $\sigma_M^2 = 20$ ) which coerces  $\rho$  to be 0.8. Note that the results in this study were not sensitive to these choices.

For the Gibbs sampling, we do 100,000 sample iterations and we discard the first 10,000 as a burn-in and we thin by using only 10% of the remaining samples evenly spaced (after which autocorrelation is negligible for all three parameters), leaving us with empirical posterior distributions of 9,000 samples.

We use logs of the densities and proportional densities to avoid computational overflow when large values of the gamma function are required. For the Metropolis-Hastings steps, we follow Yip et al. (2005) and generate  $\alpha_1$  from the current  $\alpha_0$  using the lognormal distribution with scale parameter 1 (which we found moves us around the parameter space sufficiently quickly).

$$\alpha_1 \sim \log\mathcal{N}(\ln(\alpha_0), 1) \quad \text{and} \quad S_1 \sim \log\mathcal{N}(\ln(S_0), 1)$$

We then accept  $\alpha_1$  if:

$$\ln(U(0, 1)) < \ln\left(\frac{\mathbb{P}(\alpha_1 \mid M, S, r, f_1, \dots, f_W, W, k, \lambda)}{\mathbb{P}(\alpha_0 \mid M, S, r, f_1, \dots, f_W, W, k, \lambda)} \cdot \frac{\alpha_0}{\alpha_1}\right)$$

where  $U(0, 1)$  denotes a draw from a standard uniform distribution.  $S_1$  is sampled analogously.

## 4.5 Experiment and Results

To test the `DistributeEyes` system and our count estimation techniques, we allowed up to 50 workers on MTurk to label the locations of objects in twelve diverse images. We then used the labelings to estimate the total number of objects,  $N$ . Since we also have the baseline truth, we were able to build a reasonable model to account for the workers’ false positive mistakes. Finally, we were able to assess trends of worker trainings over time, bias, difficulty, spatial effects, and others.

### 4.5.1 Experimental Setup

We designed three types of object-recognition tasks increasing in complexity: (a) find all the objects of one phenotype (b) find all the objects of one phenotype and mark points that do not serve as examples of the object, the “NONs” (c) find all the objects of multiple phenotypes and mark points that do not serve as examples of any of the objects, the “NONs.” Complexity class (c) features projects with multiple phenotypes. For the analyses and estimation found in the following sections, these multiple phenotypes were collapsed into one. In each complexity class we tested four images for a total of 12 “projects”. We tried to represent a diverse range of phenotype types (see figure 4.6) across the projects.

For each of the projects, we created 50 identical HITs which was enough to investigate the distribution of quality in worker submissions. Workers were barred from marking the same image more than once. Our reward for a completed HIT was constant at \$0.15 USD. HITs were given a 60-minute time limit. The HITs were opened to the MTurk community at large without geographical restrictions. Worker submissions were checked within 10-20 seconds using the tools described in Section X.

If the submission was especially poor, or if a server error caused an incorrect

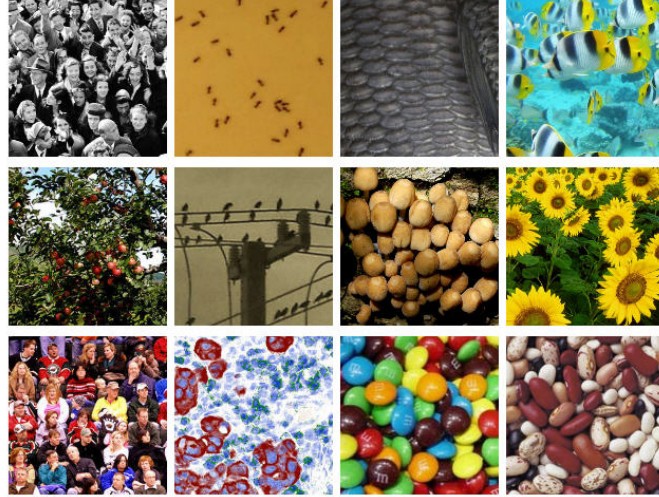


Figure 4.6: Cropped selections from the project images. From left to right then top to bottom: Faces, Ants, Scales, Fish, Tree, Birds, Mushrooms, Sunflowers, Fans, Cells, Candies, Beans

submission, it was rejected and the data was not analyzed. This is the reason that  $W$  varies between 23 and 47 in the next section.

#### 4.5.2 Data and Basic Results

Worker training data is output from the server into csv files that are then analyzed using R (R Development Core Team, 2005). A truth set was manually created for each of the twelve images. The data for each worker’s training was then compared to the truth set. If the worker’s object coordinate was within an average object radius from the truth point, it was marked as correct i.e. a “true positive”, otherwise, it was tallied as a “false positive.” Therefore, for each project we have a true positive binary matrix with workers on the rows and the true positives as the columns where a 1 represents worker  $j$  found object  $i$ . We also have a false positive binary matrix structured analogously. Overall, rejected trainings aside, the workers performed well. We measured “accuracy” for training a certain phenotype by the  $F1$ -score — a

harmonic average between *precision* (the percentage of worker’s training points that successfully corresponded to an object) and *recall* (the percentage of total objects found by the worker). The results for the positive phenotypes are summed up in table 4.1. The NONs (the negative phenotype) results were not analyzed, but anecdotally appear to be accurate as well. Broadly speaking, MTurk laborers seem to take this image-labeling task seriously.

Project	$F1\text{-score } (\bar{x} \pm SD)$	$W$
Faces	$0.96 \pm 0.03$	27
Ants	$0.96 \pm 0.02$	47
Scales	$0.89 \pm 0.05$	28
Fish	$0.90 \pm 0.04$	42
Tree	$0.73 \pm 0.16$	27
Birds	$0.97 \pm 0.02$	39
Mushrooms	$0.92 \pm 0.06$	36
Sunflowers	$0.88 \pm 0.07$	35
Fans	$0.96 \pm 0.04$	33
Cells	$0.79 \pm 0.06$	23
Candies	$0.92 \pm 0.04$	28
Beans	$0.90 \pm 0.06$	29

Table 4.1: Averages and standard deviations of  $F1$  scores for worker labelings. For the four images that had multiple phenotypes, we report the average  $F1$  score considering all points over all phenotypes. 50 image tasks were released for each project. The  $W$  is the number of trainings that were accepted and therefore the number of trainings that were analyzed.

### 4.5.3 Object Detection Probabilities are Distributed as a Mixture of Betas

Before we begin estimating  $N$ , we justify the assumption that the probability of detecting an object is beta-distributed.

We did not find compelling evidence that the probability of detecting the objects are beta-distributed for all 12 projects. However, we found strong evidence that the percent of objects detected by each worker (the true-positive rates) are beta distributed.

Because we know the truth, we know if each worker detected each true positive. We compute the proportion of true positives detected for each worker by taking the number found over the total number of true points. We then estimate the  $\alpha, \beta$  parameters of a putative beta distribution using maximum likelihood, and then test the appropriateness of the sample being drawn from the  $\text{Beta}(\alpha, \beta)$  using an Anderson-Darling test (table 4.2). For each project, the test passed at  $\alpha = 0.05$  implying that the true positives are beta-distributed (for an example, see figure 4.7).

Since the workers and the true positives are exchangeable, we have a jointly exchangeable binary array. We already know the probability model among workers is plausibly beta in all projects. Hence, by the Aldous-Hoover theorem, we know that the probability model across the true positives (the detection probabilities  $p_1, \dots, p_N$ ) must be a *mixture* of betas.

### 4.5.4 Estimating $N$ using the Experimental Data when the Truth is Known

We now apply the methods from section 4.4.3 to the data collected from the 12 images.

In the situation where the “truth is known”, we know the locations of the TP’s and can generate the statistics  $r$  and  $f_1, \dots, f_W$  using the binary true positive matrix (see section 4.5.2 for information about this construction). Note that we *ignore* all

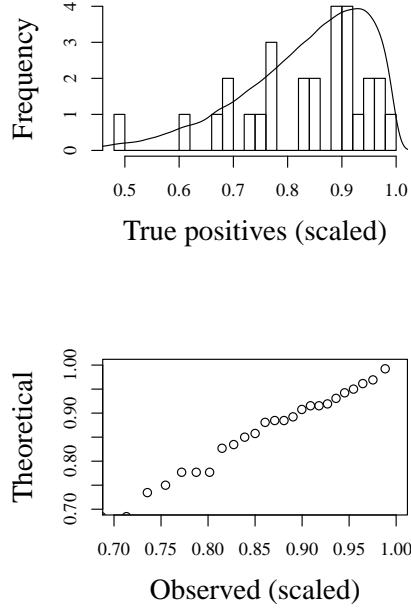


Figure 4.7: Top: Putative beta fit (found via maximum likelihood estimation) overlaid atop the true positive rate for workers in the “Fish” project. Bottom: a Q-Q plot for the putative beta distribution.

false positive points. We then proceed to use the Gibbs sampler explained in section 4.4.4.

The results for all projects are presented in figure 4.8. All of the 12 projects feature 95% posterior predictive intervals that capture the true number of TP’s. These results inform us that the model works very well when we have oracular knowledge.

#### 4.5.5 Estimating $N$ when the Truth is Unknown

In the realistic situation where “truth is unknown”, we only have the workers’ labelings. In order to estimate  $N$  without changing the model nor the Bayesian framework, we must estimate the sufficient statistics  $r$  and  $f_1, \dots, f_W$ . Recall in the  $M_h$  model that these sufficient statistics are computed *only* from true captures. Therefore, we

Project	Anderson-Darling $p$ -value
Faces	0.94
Ants	0.74
Scales	0.99
Fish	0.50
Tree	0.76
Birds	0.31
Mushrooms	0.13
Sunflowers	0.53
Fans	0.89
Cells	0.90
Candies	0.87
Beans	0.97
Apples	0.95

Table 4.2: The results of the Anderson-Darling test for the number of true positives of being Beta-distributed for all 12 projects.

proceed in three steps: (1) we group the worker training points into common objects via a clustering algorithm, (2) we estimate the likelihood that each cluster is a true object, and finally (3) we update the Gibbs sampler to account for this uncertainty.

#### 4.5.5.1 Grouping Training Points into Common Objects

We start with an unsophisticated clustering algorithm that resolves the workers' trainings to common objects. We construct the clusters through agglomeration. Starting with an initial point, we then agglomerate the closest point (measured by Euclidean distance) within a user-defined radius, then agglomerate the next closest point to the center of the initial two points, etc. We allow 10% of the workers to label a point



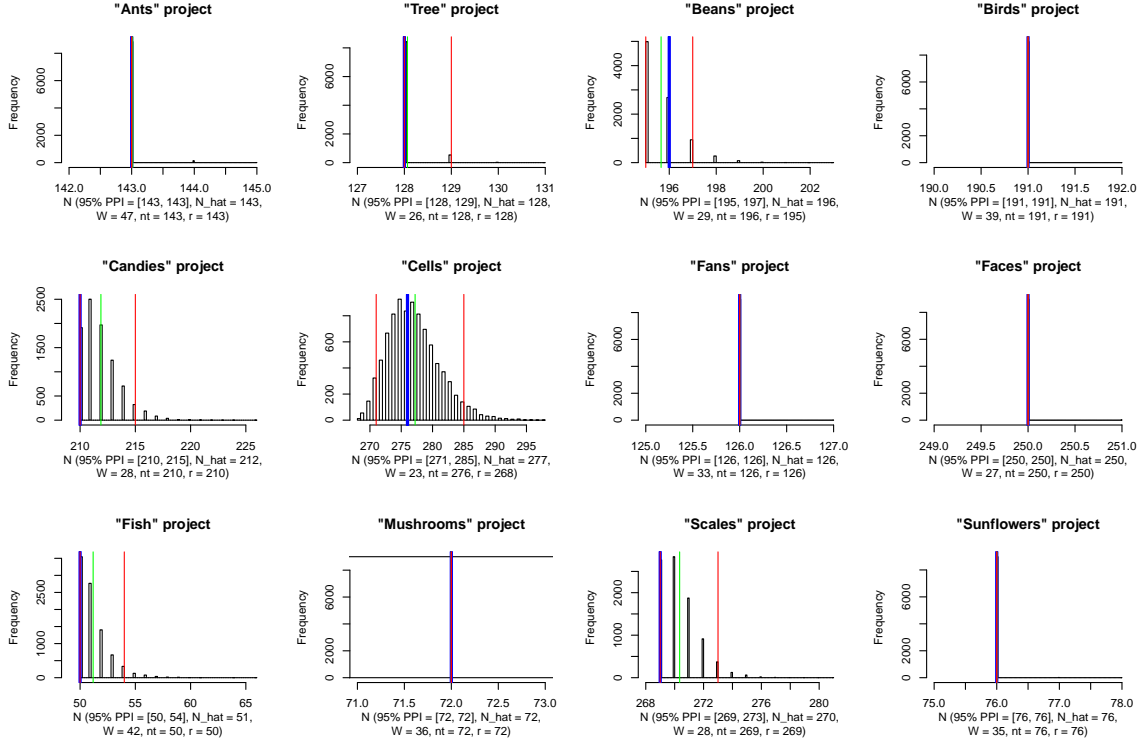


Figure 4.8: Histograms of the empirical posterior of  $N$  for each project considering the truth is known. The blue line is the number of true positives (labeled as “nt” below each histogram); the red lines are the bounds of a 95% posterior predictive interval (labeled as “95% PPI =  $[a, b]$ ”); the green line is the posterior mean (labeled as “N\_hat”). Note that 6 of the 12 projects were extremely easy, featuring large and unambiguous objects and hence there was no variation in the estimates.

twice as a duplicate.<sup>8</sup>

The total number of clusters represents the total number of objects found (both true objects and false objects) and the number of points in each cluster represents the number of workers that found that object.

Now we must estimate the probability that each cluster represents a false positive. We take a statistical learning approach.

<sup>8</sup>Our results were not sensitive to this choice.

#### 4.5.5.2 Classifying Clusters as True Captures or False Captures

The goal is to build a machine that accepts a set of points (a cluster of workers' coordinates), and outputs the probability that the cluster is a true positive (or a false positive).

We build a design matrix  $X$  where each row represents a cluster and the column entries are features of the cluster. The raw features are found in table 4.3.<sup>9</sup> Note that the radius of the objects of interest are *not* used during the computations of any of the features.

We use the statistical learning algorithm Classification and Regression Trees (CART, Breiman et al., 1984)<sup>10</sup>. We built the classifier in-sample and achieved an average error rate of 3.1%. Examining the tree split rules, we found that the features that were the most important in the discrimination were  $n$ ,  $A$ , and  $SSE$ .

#### 4.5.5.3 Updating the Capture-Recapture Gibbs Sampler

We use CART to output the probability of being a true positive.<sup>11</sup> We use these estimates to augment the  $M$ ,  $\alpha$ , and  $S$  Gibbs sampler (from section 4.4.4) by adding a sampling step for which simulates whether or not each cluster is a true positive based on the probability estimate from CART.<sup>12</sup>

$$M \mid \alpha, S, r, f_1, \dots, f_W, W, \rho \sim \text{same as before}$$

$$\alpha \mid M, S, r, f_1, \dots, f_W, W, k, \lambda \propto \text{same as before}$$

---

<sup>9</sup>The minimum volume enclosing ellipse (MVEE) was computed using an R-implementation of the Khachiyan Algorithm (Gacs and Lovász, 1981). The  $\alpha$ -hull was computed using the `alphahull` package.

<sup>10</sup>The implementation used was the R package `rpart` with standard options

<sup>11</sup>This is estimated by examining the empirical distribution in each of the leaves of the pruned classification tree.

<sup>12</sup>If the outputted probability is zero or one, we draw from a  $\text{Beta}(1, r_o + 1)$  or  $\text{Beta}(r_o + 1, 1)$  respectively where  $r_o$  is the number of *total* clusters found across all workers for the given project.

Number	Name	Description
1	$n$	The number of points in the cluster
2	$\varepsilon$	The eccentricity of the minimum volume enclosing ellipse (MVEE)
3	$foc$	The focus of the MVEE
4	$a\ell$	The length of the semi-major axis of the MVEE
5	$b\ell$	The length of the semi-minor axis of the MVEE
6	$SSE$	The sum squared Euclidean distances of each point from the cluster centroid
7	$A$	The area of the $\alpha$ -hull of the points (where $\alpha \gg 0$ )
8	$x_c, y_c$	The centroid (coordinates) of the cluster
9	$d_1, d_2, d_3, d_4$	The centroid - centroid Euclidean distance of the first closest cluster, second closest, etc

Table 4.3: The raw features used to discriminate a cluster between being a true positive and a false positive. We also add many derived features as  $A/n$ ,  $a_\ell/n$ , etc. The total number of features used was 14.

$$S \mid \alpha, M, r, f_1, \dots, f_W, W, \ell, \lambda \propto \text{same as before}$$

$$\text{cluster}_1 \sim \text{Bern}(\mathbb{P}(\text{cluster 1 is a TP according to CART}))$$

$$\text{cluster}_2 \sim \text{Bern}(\mathbb{P}(\text{cluster 2 is a TP according to CART}))$$

$$\vdots$$

$$f_1 = \text{The number of clusters with 1 training point}$$

$$\vdots$$

$$f_W = \text{The number of clusters with } W \text{ training points}$$

$$r = \sum_{i=1}^W f_i$$

To compute  $f_1$ , we count the the number of clusters that have one training point; to compute  $f_2$ , we count the number of clusters that have two training points, etc.  $r$  represents the total number of clusters found, which is equivalent to summing all the  $f_i$ 's.

Figure 4.9 shows our promising results. The scales and beans project features very ambiguous or hidden objects. We would argue that this type of object estimateion is not possible with our system due to the difficulty in distinguishing FP's from TP's.

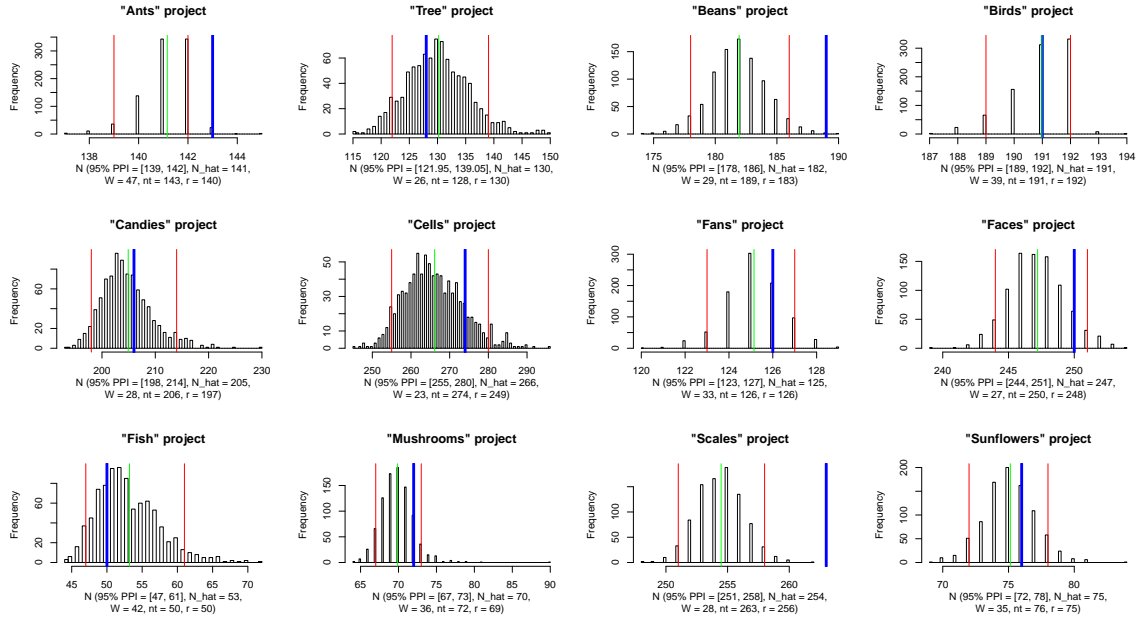


Figure 4.9: Histograms of the empirical posterior of  $N$  for each project when truth is unknown.  $f_1, \dots, f_w, r$  are sampled via a three-step procedure. Layout follows figure 4.8.

#### 4.5.6 Spatial and Temporal Findings

In addition to estimating counts using the Bayesian capture-recapture model, we carried out other investigations on the worker labelings. Below are the most important of our findings.

#### 4.5.6.1 Jumping Around Improves Accuracy

We find weak evidence that workers who “jump around” the image during training tend to have more accurate trainings. This is expected, since workers that go back and check their work do indeed jump around (see figure 4.10a). The effect was weak; on average over all projects, a one standard deviation increase in training distance yields a two-tenths standard deviation increase in  $F1$  score ( $p = 0.0001$ ).

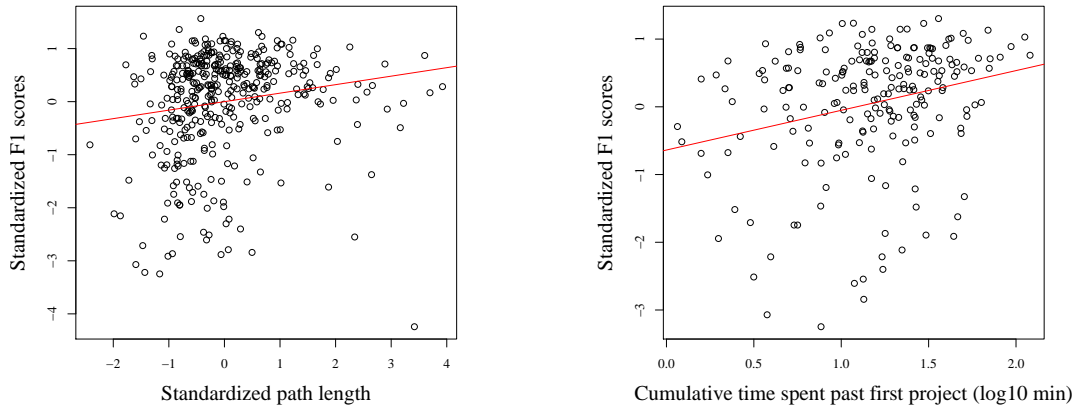


Figure 4.10: Left: Standardized accuracy regressed on standardized training distance (across all projects). Right: Standardized accuracy regressed on log 10 cumulative time training in minutes beyond first image (across all projects and workers who completed more than three images)

#### 4.5.6.2 Workers Improve with Time

We found evidence that workers overall become more accurate in training with more experience. Investigating workers that finished 3 or more projects (see figure 4.10b), we found on average that beyond the first training, for each order of magnitude increase in cumulative time training, the  $F1$  score increases by half of a standard deviation ( $p < 0.001$ ). Also expected, we find a similar effect when regressing standardized  $F1$  scores on total number of projects instead of cumulative time training.

#### 4.5.6.3 Longer Training does not Yield More Accurate Results

Surprisingly, spending more time training *within a single image* does not correspond to more accurate results. We regressed  $F1$  scores on time training for all twelve projects individually. Not one of the twelve projects features a significant correlation at a Bonferroni-corrected level.

#### 4.5.6.4 Some Images are More Difficult than Others

We ordered each project by average  $F1$  score and then compared the  $F1$  score distributions using Wilcoxon-rank tests. Anecdotally, the *easy* projects seem to have easily identifiable objects all in the spatial foreground; the *intermediate* projects either had ambiguous objects in either the foreground background; the *difficult* projects had very ambiguous objects in the foreground and/or background. The best results from `DistributeEyes` seem to come from images that have features that are both easily-identifiable and in a single spatial perspective.

We find that our capture-recapture expresses image difficulty via the width of the posterior distribution that it outputted. These widths more or less correspond to the results we present here.

#### 4.5.6.5 Trainings Can be Biased

We investigated whether workers on average tend to train above, below, or exactly the number of true objects. We find in projects where objects are ambiguous (i.e. the *intermediate* difficulty level described in section 4.5.6.4), the training is *unbiased*. We hypothesize that lazy trainers and over-ambitious trainers balance each other out. We also find that on projects where objects are well-defined (i.e. the *easy* difficulty level), the trainings are generally *biased downwards*. Since there was a clear maximum, the assiduous trainers could never balance the lazy trainers. We find that our capture-

recapture approach is excellent for mitigating these effects.

## 4.6 Conclusion

As large amounts of data are accumulated over many images and image-contexts, we can build data-based prior distributions for recognizing situations where many small objects could be located and counted. Eventually, we plan to construct a generalized “salience” classifier. A lofty goal would be to perform *unsupervised* machine learning to find objects of interest given an arbitrary context.

Another potential application is that **DistributeEyes**-trained data can lead to more efficient object tracking in videos and 3D extensions. One of the applications of the training/segmentation/counting paradigm presented here is to the tracking of animals leading to behavioral analysis (the “Ants” project). An example of passing from the static image analysis to video tracking has been developed recently (Pinter-Wollman et al., 2011).

**DistributeEyes** provides a platform for combining human generated data of high quality with computer based segmentation and feature detection output. This bridges the gap between areas when only human based systems are available (cell counting) and those where automated systems provide adequate data (image segmentation).

We can now use this on future studies both to determine the number of workers needed to acquire a reasonable estimate of a crowd count and for the collection of training data of a necessary accuracy level to be fed into automated learning algorithms.

## Acknowledgements

Thanks to Larry Brown for helpful discussions.

## Collecting labels for Word Sense Disambiguation

Adam Kapelner, Krishna Kaliannan, Lyle Ungar and Dean Foster

### Abstract

We use crowdsourcing to disambiguate 1000 words from among coarse-grained senses, the most extensive investigation to date. Ten unique participants disambiguate each example, and, using regression, we find surprising features which drive differential WSD accuracy: (a) the number of rephrasings within a sense definition is associated with higher accuracy; (b) as word frequency increases, accuracy decreases even if the number of senses is kept constant; and (c) spending more time is associated with a decrease in accuracy. We also observe that all participants are about equal in ability, practice (without feedback) does not seem to lead to improvement, and that having many participants label the same example provides a partial substitute for more expensive annotation.

### 5.1 Introduction

Word sense disambiguation (WSD) is the process of identifying the meaning, or “sense,” of a word in a written context (Ide and Véronis, 1998). In his compre-



hensive survey, Navigli (2009) considers WSD an AI-complete problem — a task which is at least as hard as the most difficult problems in artificial intelligence. Why is WSD difficult and what is driving its difficulty? This study examines human WSD performance and tries to identify drivers of accuracy. We hope that our findings can be incorporated into future WSD systems.

To examine human WSD performance, we tap pools of anonymous untrained human labor; this is known as “crowdsourcing.” A thriving pool of crowdsourced labor is Amazon’s Mechanical Turk (MTurk), an Internet-based microtask marketplace where the workers (called “Turkers”) do simple, one-off tasks (called “human intelligence tasks” or “HITs”), for small payments. See Snow et al. (2008); Callison-Burch (2010); and Akkaya et al. (2010) for MTurk’s use in NLP, and Chandler and Kapelner (2013) which is also Chapter 2 of this document and Mason and Suri (2011) for further reading on MTurk as a research platform.

We performed the first extensive look at coarse-grained WSD on MTurk. We studied a large and variegated set of words: 1,000 contextual examples of 89 distinct words annotated by 10 unique Turkers each. In the closest related literature, Snow et al. (2008) found high Turker annotation accuracy but only annotated a single word, while Passonneau et al. (2011) focused on only a few words and annotated fine-grained senses. The extensive size of our study lends itself to the discovery of new factors affecting annotator accuracy.

Our contribution is three-fold. First, we use regression to identify a variety of factors that drive accuracy such as (a) the number of rephrasings within a sense definition is associated with higher accuracy; (b) as word frequency increases, accuracy decreases even if the number of senses is kept constant; and (c) time-spent on an annotation is associated with lower accuracy.

Second, we echo previous findings, mostly from non-WSD experiments, demonstrating that Turkers are respectably accurate (Snow et al., 2008), they’re approx-

imately equal in ability (Parent, 2010; Passonneau et al., 2011), spam is virtually non-existent (Akkaya et al., 2010), responses from multiple Turkers can be pooled to achieve high quality results (Snow et al., 2008; Akkaya et al., 2010), and that workers do not improve with experience (Akkaya et al., 2010). Third, we present a system of crowdsourcing WSD boasting a throughput of about 5,000 disambiguations per day at \$0.011 per annotation.

## 5.2 Methods and data collection

We selected a subset of the OntoNotes data (Hovy et al., 2006), the SemEval-2007 coarse-grained English Lexical Sample WSD task training data (Pradhan et al., 2007). The coarse-grained senses in OntoNotes address a concern that nuanced differences in sense inventories drives disagreement among annotators (Brown et al., 2010). We picked 1,000 contextual examples at random from the full set of 22,281.<sup>1</sup> Our sample is detailed in table 1. It consisted of 590 nouns and 410 verb examples that had between 2-15 senses each (nouns:  $5.7 \pm 3.0$  senses, verbs:  $4.7 \pm 3.3$  senses). For each snippet, ten annotations were completed by ten *unique* Turkers.

### 5.2.1 The WSD HIT

We designed a simple WSD task that was rendered inside an MTurk HIT.<sup>2</sup> The Turker read one example in context with the target word emboldened, and then picked the best choice from among a set of coarse-grained senses (see Figure 5.1). We gave a

---

<sup>1</sup>We later disqualified 9 of the 1,000 because they had words with only one sense.

<sup>2</sup>The HIT was entitled “Tell us the best meaning of a word... do many and earn a lot! Really Easy!”, the wage was \$0.01, the time limit for each task was seven minutes, and the HITs expired after one hour. We posted batches of 750 new HITs to MTurk hourly upon expiration of the previous batch. Thus, the task was found readily on the homepage which drove the rapid completion.

text box for soliciting optional feedback and there was a submit button below. We term a completed HIT an “annotation.”

target word	# inst	# senses	target word	# inst	# senses	target word	# inst	# senses
affect-v	1	3	end-v	8	4	policy-n	10	3
allow-v	8	2	enjoy-v	3	2	position-n	13	7
announce-v	4	3	examine-v	3	3	power-n	12	4
approve-v	3	2	exchange-n	17	6	president-n	34	3
area-n	15	5	exist-v	1	2	produce-v	6	3
ask-v	16	6	explain-v	2	2	promise-v	3	2
attempt-v	2	2	express-v	3	3	propose-v	1	3
authority-n	3	6	feel-v	24	3	prove-v	2	6
avoid-v	2	2	find-v	7	6	raise-v	6	9
base-n	5	12	fix-v	2	6	rate-n	49	2
begin-v	7	4	future-n	16	4	recall-v	2	4
believe-v	9	2	go-v	12	14	receive-v	4	2
bill-n	18	9	hold-v	5	10	regard-v	1	3
build-v	1	4	hour-n	9	4	remember-v	8	6
buy-v	7	7	job-n	6	10	remove-v	4	2
capital-n	15	5	join-v	2	4	report-v	7	4
care-v	6	3	keep-v	11	8	rush-v	1	4
carrier-n	4	13	kill-v	5	9	say-v	104	5
chance-n	5	4	lead-v	9	7	see-v	9	10
claim-v	4	4	maintain-v	3	4	set-v	10	12
come-v	7	11	management-n	13	2	share-n	103	3
complain-v	2	2	move-n	19	4	source-n	10	6
complete-v	1	3	need-v	11	2	space-n	2	8
condition-n	7	4	network-n	9	4	start-v	8	7
defense-n	8	8	occur-v	2	4	state-n	33	4
development-n	8	3	order-n	11	9	system-n	15	7
disclose-v	5	2	part-n	14	7	turn-v	19	15
do-v	4	6	people-n	38	6	value-n	16	5
drug-n	7	3	plant-n	12	3	work-v	4	9
effect-n	7	5	point-n	27	14			

Table 5.1: The 89 words of the sample of 1000 OntoNotes snippets used in this study. “# inst” is the number of instances in the 1,000 with the corresponding target word. “# senses” is the number of sense choices provided by OntoNotes.

We employed anti-spam and survey bias minimizing techniques to obtain better data. We faded in each word in the context and the sense choices one-by-one at 300 words/min.<sup>3</sup> Additionally, we randomized the display order of the sense choices.

<sup>3</sup>As Kapelner and Chandler (2010) found (also Chapter 3 of this document), this accomplishes

This reduces “first response alternative bias” as explained in Krosnick (1991), but may decrease accuracy when compared to displaying the senses in descending frequency order as observed by Fellbaum et al. (1997). We also limited participation to US Turkers to encourage fluency in English.

## Word Meaning Task

Read the following snippet which will fade in slowly:

Apple shares fell 75 cents in over-the-counter trading to close at \$48 a share. Fiscal fourth-quarter sales grew about 18% to \$1.38 billion from \$1.17 billion a year earlier. Without the Adobe gain, Apple's full-year operating profit edged up 1.5% to \$406 million, or \$3.16 a **share**, from \$400.3 million, or \$3.08 a share. Including the Adobe gain, full-year net was \$454 million, or \$3.53 a share. Sales for the year rose nearly 30% to \$5.28 billion from \$4.07 billion a year earlier.

Please pick the meaning of the word **share** which best fits the context of the paragraph above:

- ☐ capital stock in a corporation
- ☐ a tool for tilling soil
- ☐ a portion or percentage of a whole

Submit my definition of "share" (and whatever optional feedback I left below)

My feedback:

We also welcome and give bonuses to feedback, comments, and bug reports:

Figure 5.1: An example of the WSD task that appears inside an MTurk HIT. This was displayed piecewise as each word in the example (“snippet”) and senses faded-in slowly.

Upon completion, the Turker was given an option to do another of our WSD tasks three things: (1) Turkers who plan on cheating will be more likely to leave our task, (2) Turkers will spend more time on the task and, most importantly, (3) Turkers will more carefully read and concentrate on the meaning of the text.

(this is the MTurk default). A Turker was not limited in the number of annotations they could do.<sup>4</sup> The entire study took 51 hours and cost \$110. The code, raw data, and analysis scripts are available under GPL2 at [github.com/kapelner/wordturk\\_pilot](https://github.com/kapelner/wordturk_pilot).

## 5.3 Results and data analysis

We were interested in investigating (1) which features in the target word, the context, and sense definition text affect Turker accuracy, (2) which characteristics in the Turker’s engagement of the task affect accuracy, (3) heterogeneity in worker performance, and (4) the combination of Turker responses to boost accuracy.

We recruited 595 Turkers to work on our tasks, yielding an average accuracy of 73.4%.

We measured inter-tagger agreement (ITA) using the alpha-reliability coefficient (Krippendorff, 1970) to be 0.66 (0.70 for nouns and 0.60 for verbs) which comports with Chklovski and Mihalcea (2003)’s *Open Mind Word Expert* system.

However, OntoNotes was specially designed by Hovy et al. (2006) to have 90% ITA by experts. Our measure is significantly less. Untrained Turkers should not be expected to be experts.

### 5.3.1 Performance and language characteristics

What makes WSD difficult for untrained Turkers? Are there too many senses to choose from? Is the example difficult to read? With 10,000 instances from 600 workers, we can attempt to answer these questions.

---

<sup>4</sup>The actual upper limit was all 1,000 examples but in practice, not one Turker came close to completing all of them. The most productive Turker completed 405 annotations while the median completed was 4.

We first construct the features of interest:

- target word part-of-speech (*target word is noun?*)
- target word length in characters (*# chars in target word*)
- target word frequency (*log target word frequency*)  
log of frequency in the contemporary corpus of American English (Davies, 2008).
- number of senses to choose from (*# senses to disambiguate*)
- number of characters in the correct sense definition (*# chars in definition*)
- number of rephrasings in definition text (*# rephrasings in definition*)  
For example, the word “allot” has a sense with definition text “let, make possible, give permission” which would be counted as three rephrasings.
- number of characters in context (*# chars in context*)

We add a fixed intercept for each Turker to account for correlation among tasks completed by the same worker. The result of an ordinary least squares (OLS) regression of correct (as binary) on the variables above is presented in table 2.<sup>5</sup>

We found that, controlling for all other variables, nouns have 8% higher disambiguation accuracy. This difference between noun and verb accuracy is also reflected in automatic system performance on the SemEval-2007 task Pradhan et al. (2007), and often attributed to the idea that nouns “commonly denote concrete, imagible referents” (Fellbaum et al., 1997). For each extra sense, accuracy suffers 3% which also is expected since the Turkers have more choices. We show accuracy by number of senses and part of speech in figure 5.2. We also found the longer the target word, the more difficult the task, reflecting the fact that longer words are often more complex. Similarly, the longer the context or length of definitions decreased accuracy but the effect was quite small.

---

<sup>5</sup>We also ran a variety of fixed and random effects linear and logit models, all of which gave the same significance results. We chose to present the OLS output because of its familiarity and interpretability.

	estimate	t
<i>target word is noun?</i>	8.4%	7.5 ***
<i># chars in target word</i>	-1.0%	3.6 ***
<i>log target word frequency</i>	-3.7%	7.6 ***
<i># senses to disambiguate</i>	-2.9%	19.8 ***
<i># chars in definition</i>	-0.063%	2.6 **
<i># rephrasings in definition</i>	3.4%	5.4 ***
<i># chars in context</i>	-0.0062%	2.6 **

Table 5.2: OLS regression of instance correctness on features of the target word, context, and senses. Fixed effects for each of the 595 Turkers are not shown. \*\* indicates significance at the  $< .01$  level, \*\*\* indicates significance at the  $< 0.001$  level.

Surprisingly, with each extra rephrasing of the definition of the correct sense there is a gain of 3.5%. This suggests untrained annotators benefit from receiving a variety of sense descriptions, or that more rephrasings suggests a more coarse-grained sense which is easier for annotators to understand.

Finally, as the word becomes more common in the English language (controlling for all other variables, including length of word and number of senses) accuracy still suffers. Possibly the more prevalent the word in our language, the more likely it will have senses that overlap conceptually.

### 5.3.2 Performance and Turker characteristics

Are there any characteristics about the Turker’s engagement with our task that impacts accuracy? We create the following features: time spent on task, the number of words in their optional feedback message, and the number of annotations that worker completed prior to the response being examined. To control for the difficulty of each task, we added 1,000 fixed intercepts — one for each unique task; and to control

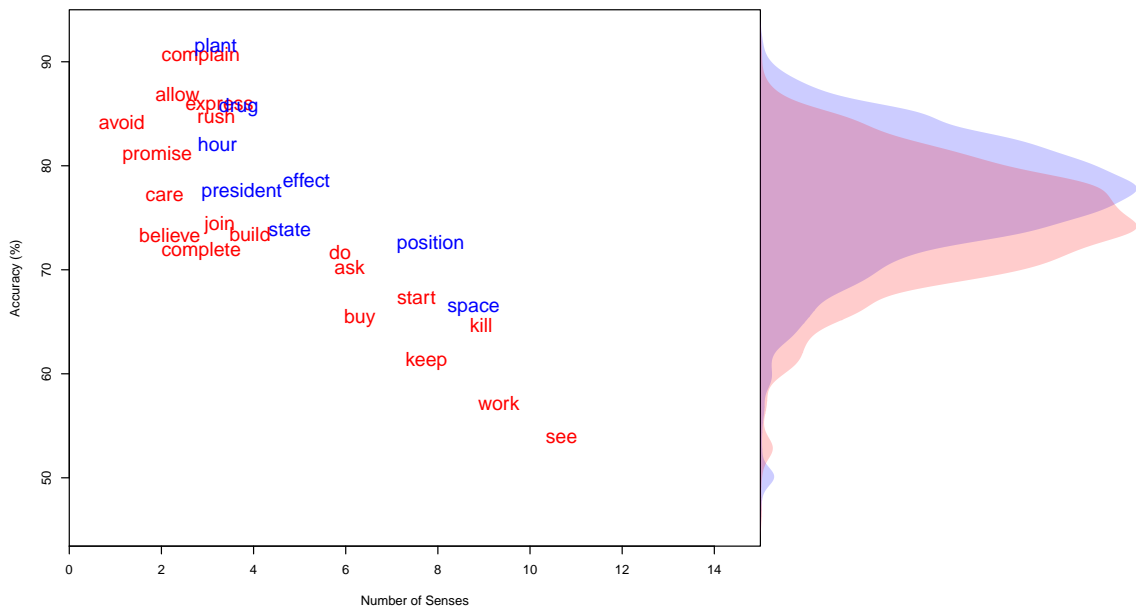


Figure 5.2: Predicted accuracy vs. number of senses for a sample of the words in our study. Nouns are blue; verbs are red. The densities are smoothed histograms of the noun and verb predicted accuracies. Note that the word display is jittered; there are at least two senses for each word.

for correlation among the workers, we added a fixed intercept for each worker. An ordinary least squares regression of the WSD task being correct (as binary) on the variables above<sup>5</sup> was run. We found that for each additional second spent on the task, accuracy drops by 0.06% ( $p < 0.001$ ). We found that, contrary to Kapelner and Chandler (2010) (also Chapter 3 of this document), leaving comments does not correspond to higher accuracy, and, in agreement with Akkaya et al. (2010), the number of tasks completed prior does not impact accuracy. This may imply that a learning effect does not exist; practice (without feedback) does *not* make perfect.

Surprisingly, spending more time on the disambiguation task associates with a significant *reduction* in accuracy ( $p < 0.001$ ).<sup>6</sup> Note that this is *after* we non-

<sup>6</sup>We validated this linear approximation by regressing time spent as a polynomial and found the effect to be monotonically decreasing with a flat stretch in the middle.



parametrically control for instance difficulty and worker ability. For every additional minute spent, a Turker is 3.6% less likely to answer correctly. We posit three theories: (1) taking breaks leads to loss of concentration (2) the “knee-jerk” response is best (rumination should be discouraged), and (3) although we control for instance difficulty, an instance may only be difficult for particular workers as evidenced by their taking longer.

### 5.3.3 Turker equality

In order to replicate previous work, we investigate Turker equality and the presence of spammers and superstars via plotting the number of annotations correct by the number of annotations completed in figure 5.3. To test the null hypothesis that all workers are equal (and thus, average), each worker’s *total contributions* are assumed to be drawn from independent Binomial random variables with probability of success  $p = 73.4\%$  (the experimental average). Does the worker’s confidence interval (CI) contain  $p$ ? Figure 5.3 reveals that every worker has approximately the same capacity for performing coarse-grained WSD except for two above-average superstars and two below-average.

To test for spammers, we test against the null hypothesis of random answering,  $p = 25.5\%$  (determined by simulation). Among workers who did a significant number of tasks,<sup>7</sup> we find only one worker who may be a spammer. We echo Akkaya et al. (2010), Snow et al. (2008), and Singh et al. (2002) and conclude there is minimal spammer contribution. Once again, we do not observe a change in accuracy by quantity of tasks completed, an observation confirmed using regression (table 3).

---

<sup>7</sup>We do not have significant power to claim a worker has accuracy of even 50% until about  $n = 79$  at the Bonferroni-corrected  $\alpha$  level.

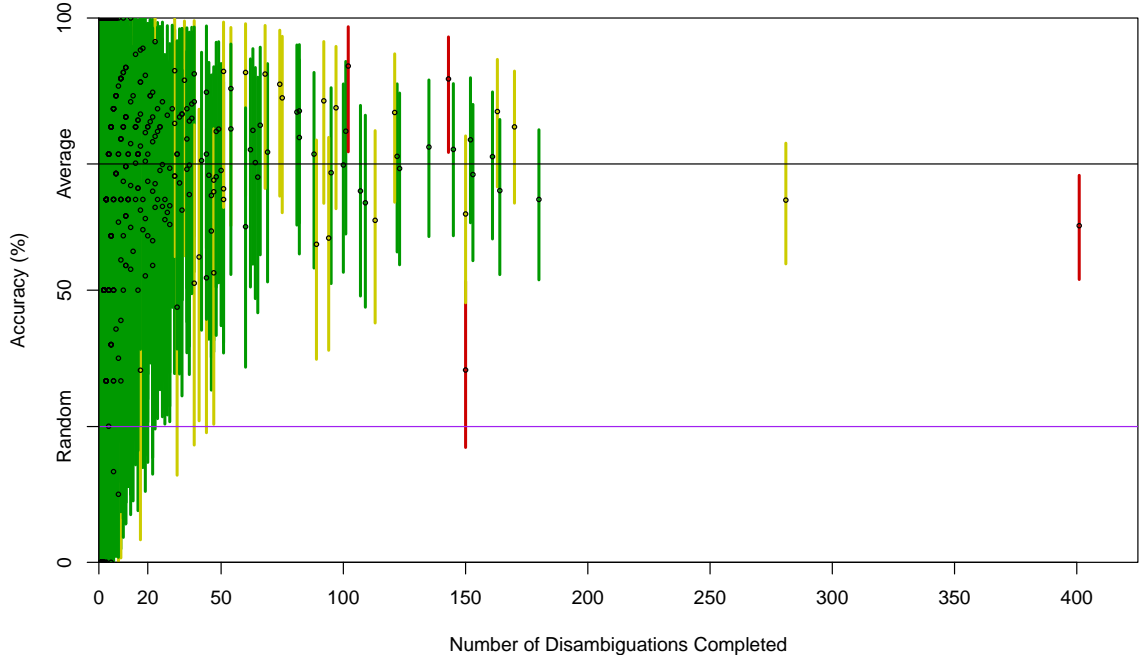


Figure 5.3: Accuracy of all 595 Turkers. The black line is the average accuracy ( $p = 73.4\%$ ) and the purple line represents random sense choice accuracy ( $25.5\%$ ). We plot the Bonferroni-corrected Binomial proportion confidence intervals in green if they include  $p$ , yellow if the non-Bonferroni-corrected confidence intervals do not include  $p$ , and red if neither include  $p$ .

### 5.3.4 Combining responses to optimize prediction

We can combine the 10 unique disambiguation responses for each of the 1000 examples to yield higher accuracy. Our algorithm is naive — we take the plurality vote and arbitrate ties randomly. Snow et al. (2008) found such an approach results in higher accuracy for disambiguating ‘president’. We wondered if the same is true for our more extensive dataset and annotations.

Table 4 illustrates our results. There is an overall accuracy of 85.7% when annotations from all workers are aggregated. This is in the ballpark of the best supervised statistical learning techniques which boast almost 90% (Pradhan et al., 2007).<sup>8</sup> We

<sup>8</sup>Note that this is not a fair comparison. These supervised algorithms were given all the training

# of Annotations	2	3	4	5	6	7	8	9	10	2.4 (1st plurality)
Accuracy	.734	.795	.808	.824	.830	.837	.840	.843	.857	.811

Table 5.3: Accuracy of the WSD task using plurality voting for different numbers of Turkers. The last column is the accuracy of the variable algorithm: starting with two workers and adding an additional worker until plurality.

determined the marginal accuracy of each added Turker by simulating random subsets of two Turkers, three Turkers, etc and employed the same plurality vote.

With techniques such as discarding results from annotators who often disagree, and giving the annotators sense choices in order of sense frequency, we could likely achieve higher accuracy.

Given MTurk annotation costs, we believe this system can be extended to accurately disambiguate a million words a year at 80% accuracy for about \$25,000. This demonstrates the system’s potential for mass annotation, but we reiterate that the main goal of this current work was to gain insight into drivers of WSD accuracy.

## 5.4 Conclusion

We performed the first extensive study of crowdsourced coarse-grained word sense disambiguation in order to gain insight into the behavioral and linguistic features that affect accuracy of the untrained annotations. As expected, we found results improved when there were less sense choices or when the target word was a noun, and that untrained workers did not improve with experience. However, we also discovered data while Turkers were *not* given any previous examples. They also arbitrated based on the senses’ frequencies while we randomized the order that the senses appeared in. Finally, they were not limited to polysemous words as we were.

ered surprising insights: (1) the number of rephrasings in the correct sense definition corresponded with improved annotator accuracy, (2) frequency of target word corresponded with lower accuracy, and (3) time-spent on an individual annotation corresponded with lower accuracy. It also seems that time pressure may increase accuracy. Future experiments that prove these relationships causally may be fruitful. Lastly, we looked at Turker ability and found that they are all roughly equal in ability, and although individually not as accurate as experts, many Turkers may be pooled to improve accuracy.

## **Acknowledgments**

We thank Mark Liberman and Lynn Selhat for helpful comments and discussions. Adam Kapelner thanks the National Science Foundation for the Graduate Research Fellowship that made this work possible.

## Well-Being Prediction

Hansen Andrew Schwartz, Johannes Eichstaedt, Eduardo Blanco, Megha Agrawal, Lukasz Dziurzynski, Margaret Kern, Adam Kapelner, Greg Park, Sneha Jha, David Stillwell, Michael Kosinski and Lyle Ungar

### Abstract

We present the task of predicting well-being, as measured by the *satisfaction with life* scale, through the language people use in social media. Well-being, encompassing much more than emotion or mood, is attributed to people rather than text. We create a predictive model over *user-level* well-being scores obtained through Facebook. However, with individual *messages* themselves being the units that communicate well-being, we explore a cascaded message-to-user model which utilizes message-level models to inform user-level predictions. Our models perform over and above a baseline of lexicon-based happiness measurements, with the cascaded approach being most accurate by a significant margin. Finally, we suggest that analyses of language and well-being can go beyond prediction by identifying the language that characterizes well-being.

## 6.1 Introduction

As human beings, we desire “the good life”. When the British Broadcasting Corporation (BBC) asked 1,001 Britons what the prime objective of their government should be – “greatest happiness” or “greatest wealth” – 81% answered with happiness (Easton, 2006). In other studies around the globe, 69% of people on average rate well-being as more important than any other life outcome (Diener, 2000).

Well-being is more than simply positive emotion. Psychologists, organizations, and governments measuring well-being are now turning to comprehensive constructs of multiple factors such as meaning in life, engagement in activities, and the state of one’s relationships in addition to positive emotion (Seligman, 2011). While some language analyses have explored “happiness” simply based on emotion or mood (Alm et al., 2005; Mihalcea and Liu, 2006; Kramer, 2010; Dodds et al., 2011), modeling the broader construct of well-being is a relatively unexplored task.

In this paper, we present the task of predicting well-being based on natural language use. We develop a system that predicts the *satisfaction with life* (a measure of well-being) of Facebook *users* based on simple lexical and topical features. However, since individual *messages* themselves are the units of expression, we investigate the use of message-level models to improve user-level predictions. The message data, which was more easy to come by through Amazon’s Mechanical Turk, allowed us both to supplement our *satisfaction with life* data, as well as explore another construct of well-being, *PERMA* (discussed below). Lastly, for a human-level attribute like well-being, insights toward greater understanding is potentially just as important as prediction; Toward this end, we identify topics that correlate most with well-being as clues to achieving “the good life.”

## 6.2 Background

### 6.2.1 Well-Being

Despite a desire for “the good life”,<sup>1</sup> well-being has traditionally been measured *indirectly* as a lack of problems (e.g., lack of depression and psychological disorder, low crime/disease/poverty rates) or by economic prosperity (i.e., gross domestic product). (Peterson and Park, 2003) aptly note: “if our interest is in the good life, we must look explicitly at indices of human thriving” (p. 144). Government agencies around the world are beginning to shift their attention toward *directly* measuring well-being (Stiglitz et al., 2009). In 2011, the U.K. Office of National Statistics piloted four well-being questions in their annual national survey. Similar efforts are now underway in France, Canada, Bhutan, South Africa, and Australia. These initiatives follow a long history of academics who have attempted to rethink the notion of progress, deemphasizing the sole reliance on economic indicators and arguing instead that the welfare of a nation must be understood more holistically, with consideration of aspects such as social belonging, meaning, and optimism in the population (Abramovitz et al., 1973; Layard, 2005; Seligman, 2011).

**Satisfaction with Life.** “Satisfaction with Life” (*SWL*) is a well-established construct in the psychological literature to directly measure overall well-being. Measures of life satisfaction have been used reliably for several decades, and are increasingly being utilized by governments and organizations around the world as informative social indicators for policy decisions (Diener et al., 2012). It is assessed by asking people to indicate to what extent they agree with statements such as “In most ways my life is

---

<sup>1</sup>We consider “the good life” and *well-being* synonymous within this work – measured through the *satisfaction with life scale* or *PERMA*. *Happiness* can refer to either well-being or simply emotion / mood; we only use *happiness* when referencing other works that use it.

close to my ideal” (Diener et al., 1985). *SWL* draws on the subjects’ evaluative judgments and seems to be highly comparative both within nations and between nations (Diener et al., 2010). Furthermore, although it is conceptualized as a single dimension, it is influenced by multiple domains, such as meaning in life or relationships in addition to emotional states.

**PERMA.** Other psychologists have suggested that well-being measures should distinguish multiple domains of life (Ryff and Keyes, 1995; Seligman, 2011). In such “dashboard approaches” just like the “state” of an airplane is not given by a single indicator but instead by a variety of different indicators (altitude, speed, heading, fuel consumption), well-being can be understood as a construct that spans multiple dimensions (Forgeard et al., 2011; Frey & Stutzer, 2010; Seligman, 2011; Stiglitz, Sen, & Fitoussi, 2009). In his Well-Being Theory, Seligman (2011) suggests five major pillars: *Positive Emotions*, *Engagement*, *Relationships*, *Meaning*, and *Accomplishment* (*PERMA*). Although greater specificity could be delineated, we chose the *PERMA* constructs since they capture fairly explicit and often foregrounded ends people pursue (things discussed in social media). Other “dashboard approaches” seek to capture subtle psychological notions such as “autonomy” or “self-acceptance” (Ryff and Keyes, 1995).

*Positive Emotion.* Positive emotion is one of the most studied aspects of well-being. It includes positively valenced emotions such as joy, contentment, and excitement. Although positive emotion is an important part of well-being, the positive psychology field has demonstrated an increasing emphasis on other components of well-being rather than emotion alone (Forgeard et al., 2011).

*Engagement.* Engagement includes involvement / participation in groups or activities; enthusiasm and interested in activities; commitment / dedication to work, activities, or groups; concentration and persistence through challenges (Appleton et al.,



2008; Csikszentmihalyi, 1997). It represents a passion and involvement in life, rather than apathy and boredom.

*Relationships.* Positive relationships includes trusting others, perceiving others as being there if needed, receiving social support, and giving to others (Taylor, 2011), which considerable evidence identifies as important predictors of health and life outcomes (Tay et al., 2012).

*Meaning.* Meaning in life captures having a sense of purpose, significance, and understanding in life (Steger et al., 2008), and its pursuit has often been considered an important component of psychological well-being (Frankl, 1997; Steger et al., 2008). It involves transcending the self, feeling a sense of connection to a higher power or purpose, and provides goals or a course of direction to follow (Schueller and Seligman, 2010).

*Accomplishment.* Accomplishment is the most objective component of well-being, often defined in terms of awards, and other outward achievement (Ericsson, 2002; Forgeard et al., 2011). It can also represent a sense of mastery, perceived competence, and goal attainment (Pearlin and Schooler, 1978).

### 6.2.2 Related Tasks

Among others, predicting emotion, mood, personality, and classic sentiment analyses are related to our task. In this section, we provide a cursory review with a focus on social media and related approaches. To our knowledge, well-being prediction as more than positive emotion is a novel task, and considering *message*-level predictions to improve a *user*-level model has not been explored.

Although sentiment analysis usually takes place at the single document or sentence level (Turney, 2002; Pang et al., 2002), some have explored multi-level approaches. For example, (Pang and Lee, 2004) utilize a cascaded model where sentence level subjectivity classifiers are used to determine whether a sentence should be included for

document-level analysis. (McDonald et al., 2007) expanded the idea by incorporating a joint-model of sentence and document level sentiment annotations. We find this sentence-to-document multi-level prediction analogous to our multi-level approach, though the efforts aim at predicting the sentiment of text; they do not model any feature at the *user* level (human attributes). Our average user writes 123 messages, far more than the number of sentences in the average movie review document of a standard sentiment analysis task.

Some sentiment tasks have moved closer to human-level attributes. For example, signals used in distant supervision (Read, 2005; Davidov et al., 2010), where heuristics replace manual annotations (e.g., “:(” = negative polarity), somewhat captures people’s emotional states. (Tang and Chen, 2011) attempts to differentiate between the emotions of the writer and the reader of content on a microblogging platform with social network characteristics. More directly, some have looked at predicting emotion of text (Alm et al., 2005) or more specifically learned the language of happy and sad blogposts based on self-annotated moods in LiveJournal (Mihalcea and Liu, 2006). (Aman and Szpakowicz, 2007) identified thesaurus-based topics related to the emotional expressions from an English blog corpus. (Strapparava and Mihalcea, 2008) constructed models from a large corpora of world knowledge to identify the affective tone across six basic emotions in texts.

While mood and sentiment analysis aim at text annotations, there have been a few tasks looking at modeling human-level conditions by the language one uses. Work on personality follows the widely accepted Big-Five personality traits (McCrae and John, 1992). (Bachrach et al., 2012) analyzes correlations with basic Facebook features (number of friends, photos, tags, likes, etc.) and (Yarkoni, 2010) with LIWC (Pennebaker et al., 2001) word categories (pronouns, cognitive process, etc.). Personality prediction efforts use LIWC categories coupled with other shallow features (Mairesse et al., 2007; Gill et al., 2009) and n-grams (Nowson, 2007). Recently,

a competition was held to predict the personality of twitter users based on use of LIWC categorical features (Sumner et al., 2012), and things have moved somewhat beyond personality; for example, finding indicators of psychological health (Saleem et al., 2012). While these works predict personality at the user level, they do not incorporate message-level features.

Most recently, (Kramer, 2010) used a premade lexicon of positive and negative emotion words to measure “gross national happiness” and (Dodds et al., 2011) used MTurk ratings of individual words for positive or negative valence. Dodds’ MTurk application served as a model for our MTurk data collection, but rather than asking workers to annotated individual words, we had whole messages annotated, thus putting words into context. Both Dodds and Kramer demonstrated face validity (i.e. people are happier on the weekends or sad after celebrities die), though they lacked an empirical evaluation of the degree to which they accurately measure happiness. By using these approaches as baselines for *SWL* prediction, we provide an empirical evaluation here. Furthermore, in this paper we model well-being as more than emotion or mood.

## 6.3 Method

Here we describe predictive models of well-being, as measured through the *satisfaction with life* (*SWL*) scale and the *PERMA* construct. We describe a user-level model, a message-level model, and then a cascaded model whereby message predictions inform the user-level predictions.

As the first work attempting the prediction of user-level *SWL*, we explore a limited and consistent feature space for both user-level and message-level models:

**ngrams.** (unigrams and bigrams) We use an emoticon-aware tokenizer built on top of Pott’s “happyfuntokenizer”<sup>2</sup> to extract all unigrams and bigrams.

**topics.** Topics were created by running *latent dirichlet allocation* (*LDA*) over a set of 18 million Facebook status updates from MyPersonality (Kosinski and Stillwell, 2012). Below, we use 2,000 topics, which seems based on inspection to offer a good compromise between topic coherence and topic overlap. As shown in results, the topics both look good and improve predictive accuracy when used as features. A user,  $u$ ’s, usage of a topic,  $t$ , was calculated as:

$$\mathbb{P}(t \mid u) = \sum_{w \in words_u} \mathbb{P}(t \mid w) \mathbb{P}(w \mid u)$$

where  $\mathbb{P}(t \mid w)$  is the probability of a topic given a word (a value provided by the generated topic model) and  $\mathbb{P}(w \mid u)$  is a user’s probability of mentioning word  $w$ . Additionally, beyond simply prediction utility, topics provide insight into the latent categories of language that characterize well-being.

**lexica.** We also included the manually developed categories of words from Linguistic Inquiry and Word Count (LIWC) (Pennebaker et al., 2007) as well as the weighted lexica from Dodd’s Hedonometer (Dodds et al., 2011). While LDA topics provide a data-driven set of categorical features, these lexica provide features grounded in psychological and linguistic theory and human judgment. LIWC, in particular, was developed over decades with many iterations (Pennebaker et al., 2007). The *positive emotion* and *negative emotion* categories of LIWC are widely used, including being used by (Kramer, 2010) in his measure of “gross national happiness” (*GNH*) and by (Golder and Macy, 2011) to track diurnal mood variation. *GNH* along with Dodd’s

---

<sup>2</sup><http://sentiment.christopherpotts.net/code-data/happyfuntokenizing.py>

*Hedonometer* also function as baselines for our predictive models.

Each feature is encoded as both binary (1 if mentioned at least once, 0 otherwise) as well as frequency normalized by the total number of words by the user or message being observed ( $\frac{freq(feature)}{\sum_{w \in Words} freq(w)}$ ). We filtered out infrequent features defined as those used by less than 10% of users or 0.1% of messages.

### 6.3.1 User-Level Models.

Our basic user-level model fits ridge regression (Hoerl and Kennard, 1970) of the features described above to *SWL* scores. Since unigram and bigram features produce a large sparse input matrix, we use Randomized Principal Component Analysis (*PCA*) (Halko et al., 2011) to transform the space to a more manageable size for ridge. Specifically, we reduce the feature matrix to  $\frac{1}{4} \times \text{train\_size}$  components for all models utilizing more than that many features. The projection matrix from *PCA* is stored as part of the model such that prediction / test data is transformed based on a projection matrix fit to training data. We also tried Lasso (L1 penalized) regression (Tibshirani, 1996), which works well with sparse data, but we got better results using the *PCA* and Ridge combination.

Making the task slightly easier, we also tested classification by splitting the *SWL* data into high / low classes at the mean. Such an approach is common for other works dealing with human-level real-valued variables such as personality prediction (Mairesse et al., 2007). We use a linear support-vector machine (Fan et al., 2008), and to keep consistency with regression, apply *PCA* to the feature space before fitting.

### 6.3.2 Message-level Models.

Our message-level well-being models serve two purposes. First, we integrate them into a cascaded message-to-user level model (described next) in order to improve user-level

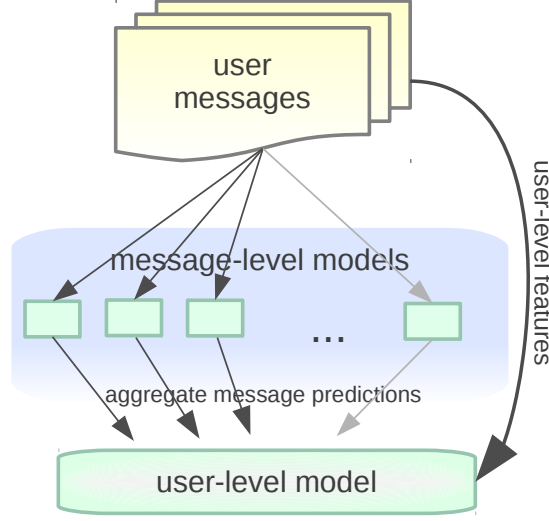


Figure 6.1: Basic depiction of our cascaded message-to-user level prediction model.

predictions. Second, they allow us to explore a “dashboard” construct of well-being, *PERMA*, containing many factors. Since a *SWL* score represents a comprehensive well-being concept (as described in the background), one might expect it to benefit from information about specific domains of well-being such as the positive relationship or accomplishment in *PERMA*

The features at the message level mirrored those at the user-level, except that they are aggregated per message. We again used the two-stage process of randomized *PCA* transformation of our features before fitting and predicting with either ridge regression or support-vector machines for classification. However, unlike the user level where we built a model for only one outcome, here we also build a model for each of the positive and negative polarities of *PERMA*, resulting in 11 total for each of classification and regression. As discussed in section 6.4.1, our message-level *PERMA* annotations either indicated “no expression” (rating of 0) or the extent of expression (rating of 1 - 6). Therefore, we setup our classifiers to indicate whether a message expressed (i.e.  $rating \geq 1$ ) the given component of *PERMA*. For *SWL*, which was scored as a continuum (from 0 to 3(neutral) to 6), the classifier simply indicated whether the message was not neutral (i.e.  $2 < rating < 4$ ).

Status	P <sup>+</sup>	P <sup>-</sup>	E <sup>+</sup>	E <sup>-</sup>	R <sup>+</sup>	R <sup>-</sup>	M <sup>+</sup>	M <sup>-</sup>	A <sup>+</sup>	A <sup>-</sup>
Celebrating this amazing day.. lmao.. first of many	6.0	0.0	3.0	0.0	4.5	0.0	3.5	0.5	5.5	0.0
I wanna thank GOD for letting me see another BDAY...LOVE YA BIG MAMA I KNO U SMILING DOWN ON ME!!!!	3.0	1.7	4.0	0.0	5.0	0.0	5.5	0.0	3.0	0.0
Goin to laundry mat got hella laundry to do uuuhhhh.....just did a major clean up take him out...take him out of the game already..	0.0	5.5	3.0	0.0	4.5	1.7	0.0	0.0	0.0	0.0
I have such an amazing bf he took good care of me at the hospital.which he always takes good care of me.Im so blessed to have him.	5.0	1.0	1.7	0.0	6.0	1.0	2.0	0.5	0.0	0.0
Status							Cont. Category			
I hate wen you watch a movie and the ending is sooo predictable :/							E <sup>+</sup> : 3.0		E <sup>-</sup> : 2.7	
Just when I thought my whole world had crumbled into a million pieces, you came along and brought me crazy glue and band-aids.							R <sup>+</sup> : 2.7		R <sup>-</sup> : 2.0	
Another FUN bright photoshoot coming up in my future! Cant WAIT! :)							M <sup>+</sup> : 2.5		M <sup>-</sup> : 1.3	

Table 6.1: Top: Examples of statuses with contrastive values for each PERMA category; Bottom: Examples expressing both polarities for PERMA categories (engagement, relationships, and meaning).

### 6.3.3 Cascaded Message-to-User Level Well-Being Prediction

Although well-being is attributed to people, we believe it might be possible to capture expressions of it at the message level (e.g. using MTurker ratings of messages) and to pass this information along to the user level. Referring to Figures 6.1, our cascaded model aggregates predictions across all of each user’s messages and incorporates them as the mean prediction across a user’s messages. This mean in turn becomes a feature supplied to the user-level model. We used both the classification and regression

predictions from the message-level to inform user-level predictions. For example, if we train message-level predictors for both polarities for each of the five domains of PERMA, that results in 20 features within the cascaded model: 5 domains \* 2 polarities \* 2 types of predictors (classification and regression).

## 6.4 Evaluation

We evaluated our user-level, message-level, and cascaded message-to-user predictive models over a corpora of Facebook status updates. The first is a user-level corpus in which volunteers from Facebook took the *SWL* questionnaire and shared their status updates. The second corpus contains messages from Facebook which were annotated using Amazon’s MTurk. Each corpus was divided randomly into 80% training/development instances and 20% test. We then test whether our models do better than a baseline of happy/hedonic lexica (Kramer, 2010; Dodds et al., 2011), and whether a cascaded message-to-user level model improves upon the pure user-level model.

### 6.4.1 Data Sets

**User-Level Data.** Our user-level data was acquired through the MyPersonality Facebook App (Kosinski and Stillwell, 2012). In this case, we focused on the users who took Deiner’s (1985) *SWL* questionnaire and who agreed to share their status updates for research purposes. Past research using this scale has demonstrated its high internal consistency (reported alphas range from .79 to .89), and moderate temporal stability (reported test-retest correlations over two month intervals range from .50-.82).

Mean *SWL* in our sample was 4.3 (on a 7-point scale), consistent with the mean *SWL* reported in North American college and adult samples (typically 4.8). A subset



of our sample ( $n = 157$ ) retook the *SWL* scale six months later, and the resulting test-retest correlation ( $r = .62$ ) was similar to those reported in past studies (Pavot and Diener, 1993). Test-retest correlation forms an upper-bound on the predictive accuracy we should expect to get from our models.

We further refined the data to just include status updates made in the 6 months prior to their taking the *SWL* questionnaire and only from those participants who wrote at least 500 words in that time. This resulted in a total of 2,198 individuals writing a total 260,840 messages.

**Message-Level Data.** We used Amazon’s Mechanical Turk (MTurk) to acquire *PERMA* and *SWL* annotations for 5,100 public Facebook status updates. The status updates were randomly selected from among 230 million public Facebook messages, which contained at least 50% English words according to the ASpell Official English dictionary.<sup>3,4</sup>

On MTurk, the largest online task-based labor market, tasks are completed by an on-demand labor force composed of “Turkers” (Chandler and Kapelner, 2013). We designed an MTurk application to render one annotation per task which paid a wage of \$0.01. Upon entry, turkers completed a research consent and were shown a video that explained the well-being category they were rating. Upon passing a brief quiz, they were qualified to annotate the particular component of well-being for which they were trained. If they wished to work on a different component, then they were required to qualify with a separate training video and quiz.

By using a slider with rating scale that ranged from none (0) to very strong (6), turkers indicated the “extent to which [a message] expresses” the particular component, both positive and negative, of *PERMA*. For *SWL*, which is conceived uni-dimensionally, they indicated how much they agreed with a statement that the

---

<sup>3</sup>[http://misc.aspell.net/wiki/English\\_Dictionaries](http://misc.aspell.net/wiki/English_Dictionaries)

<sup>4</sup>We plan to make this data available upon IRB approval for public sharing.

message indicated satisfaction with life on a scale ranging from strongly disagree (0) to neutral (3) to strongly agree (6).

We opted for more messages rather than more ratings per messages, utilizing two ratings for most messages and bringing in a third rating for difficult instances (defined as those within 1 standard deviation, 2 points, of each other). When considering the reliability between the initial 2 ratings, intra-class correlations were in the “moderate” to “substantial” agreement range .4 to .8 (Shrout and Fleiss, 1979) for all but two domains; positive engagement and negative accomplishment were in the “fair” agreement range: .2 to .4 (range labels provided by (Landis and Koch, 1977)) suggesting they were more difficult to annotate and promoting the idea of getting a third rating for more accurate scores (less accurate ratings only make our task of improving user-level predictions more difficult). In the end we used the mean rating for each message as the gold-standard.

Refer to Table 6.1 for examples of annotated messages. We found the PERMA categories each contain different but related information. Correlations within the five positive PERMA categories and within the five negative categories ranged from Pearson  $r = 0.36$  to  $0.68$ . Consistent with the psychological literature (Diener and Seligman, 2002), the highest correlations were between relationships and positive emotion. The correlations between the positive and negative dimensions were negative but weak (we show some examples where messages express both polarities in table 6.1), ranging from  $r = -0.04$  to  $-0.39$ , suggesting the separation of polarities into separate scale was appropriate. Given these correlations, each PERMA category should still contribute independent information for predicting well being.

The messages in this dataset and those from the user-level dataset are completely disjoint. Thus, when creating the cascaded model, the message-level predictors will have never seen any of the users’ messages before, avoiding overfit issues. Though this data-set is smaller in the number of messages (5,100 versus 260,000 for the MyPer-

<b>baselines</b>	<i>r</i>	<i>acc</i>
(mean   random-class)	.000	.50
<i>lexica: GNH</i>	.210	-
<i>lexica: Hedonometer</i>	.108	-
<b>user-level models</b>	<i>r</i>	<i>acc</i>
<i>ngrams</i>	.262	.60
<i>topics</i>	.254	.58
<i>lexica</i>	.198	.59
<i>ngrams + topics</i>	.299	.58
<i>ngrams + lexica</i>	.269	.60
<i>topics + lexica</i>	.252	.60
<i>ngrams + topics + lexica</i>	.301	.59
<b>cascaded models</b>	<i>r</i>	<i>acc</i>
<i>msg-reg</i>	.236	.63
<i>msg-class</i>	.237	.62
<i>msg-reg + msg-class</i>	.247	.63
<b><i>msg-reg + msg-class + usr feats</i></b>	<b>.333</b>	.64

Table 6.2: User-level prediction scores as the Pearson correlation coefficient ( $r$ ) or classification accuracy ( $acc$ ). Refer to previous section for feature definitions. *msg-reg*: message-level regression feeding cascaded model; *msg-class*: message-level classification feeds cascaded model; *usr feats*: all user level language features (*ngrams + topics + lexica*). bold: significant ( $p < .05$ ) improvement over user-level features alone.

sonality set), our hope is that the fact that it has more labels (5,100 messages versus 2,200 users) can supplement to user-level data to improve accuracy. Furthermore, the benefit of a cascaded model might become even greater if fewer labeled users are

available.

### 6.4.2 Results

Results are reported using the Pearson correlation coefficient ( $r$ ). The data was divided such that 80% (1758 users) was used for training and development while 20% was held out for testing (440 users). The scores represent the comparison between predictions over the test data and the user’s scores from the *SWL* survey. Since we’re studying well-being, the Pearson correlation allows us to frame the results with a metric used widely in social sciences. Human psychological variables have a “correlational upper-bound” in the range of  $r = 0.3$  to  $r = 0.4$  with human behaviors such as language use (Meyer et al., 2001).

Table 6.2 shows the user-level predictions for all combinations of features as well as the cascaded models. We see that an ngram model out-performs baselines of methods used previously for happiness prediction (Dodds et al., 2011; Kramer, 2010), while our best user-level results come from a combination of *ngrams*, *topics*, and the *lexica* (though not significantly better than *ngrams* and *topics* alone).

We see that the cascaded models gave a significant increase in performance, from .301 from the user-level language features alone to .333 with cascaded models. This is quite surprising considering the message-level models were only based on 5,100 messages, but there were over 200,000 messages across all the users. A similar improvement is seen for the classification task, though there is no improvement above *ngrams* until adding the cascaded models.

We looked at the impact of individual message-level predictors for each component of PERMA and determined all message-level predictors are playing a part. With out-of-sample correlations ranging from  $r = .15$  to  $r = .247$ , all domains of *PERMA*, as modeled through the language of annotated messages, had an impact on user-level *SWL*.

features	P <sup>+</sup>	P <sup>-</sup>	E <sup>+</sup>	E <sup>-</sup>	R <sup>+</sup>	R <sup>-</sup>	M <sup>+</sup>	M <sup>-</sup>	A <sup>+</sup>	A <sup>-</sup>	SWL
<i>ngrams</i>	0.563	0.412	0.349	0.347	0.604	0.365	0.405	0.279	0.363	0.285	0.479
<i>topics</i>	0.500	0.430	0.252	0.405	0.500	0.417	0.400	0.286	0.260	0.322	0.495
<i>lexica</i>	0.413	0.442	0.205	0.292	0.445	0.376	0.217	0.205	0.147	0.292	0.430
<i>ngrams+</i>	0.598	0.492	0.369	0.421	0.641	0.401	<b>0.451</b>	0.314	0.380	<b>0.398</b>	0.550
<i>topics</i>											
<i>ngrams+</i>	0.613	0.526	0.346	0.375	0.621	0.446	0.435	<b>0.331</b>	<b>0.414</b>	0.361	0.543
<i>lexica</i>											
<i>topics+</i>	0.509	0.464	0.247	0.376	0.525	0.423	0.372	0.26	0.307	0.268	0.505
<i>lexica</i>											
<b><i>ngrams+</i></b>	<b>0.617</b>	<b>0.504</b>	<b>0.374</b>	<b>0.422</b>	<b>0.655</b>	<b>0.427</b>	0.441	0.311	0.402	0.352	<b>0.566</b>
<b><i>topics+</i></b>											
<b><i>lexica</i></b>											

Table 6.3: Message-level prediction scores as the Pearson correlation coefficient ( $r$ ) across the PERMA and SWL categories.

Predicting the annotations at the message-level was easier than user-level well-being prediction. Refer to Table 6.3 for the accuracies when testing over the message-level corpus itself (again divided into 80% train, 20% test). Interestingly, for some categories *topics* were better than *ngrams* while for others *ngrams* were better than *topics*. We also noticed positive relationships seems to be the easiest of all to predict, higher than both positive emotion and *SWL*.

## 6.5 Discussion: Well-Being Insights

The LDA topics most highly correlated with user-level SWL shed light on the mechanisms which may contribute to a person’s satisfaction with their life in a way that seem valid in light of the psychological literature. Figure 6.2 shows four of the top ten positively correlated topics, and the two leading negative ones. Several of the topics



Figure 6.2: The top 4 topics positively correlated and 2 topics negatively correlated with *SWL*.

tap various aspects of engagement. Engagement is a multidimensional construct that can be considered an important predictor or an indicator of well-being (Appleton et al., 2008; Fredricks et al., 2011). Engagement is considered a key part of healthy aging (Rowe and Kahn, 1987) and is included as a core component of well-being in some theories (Seligman, 2011). The top two topics arguably tap aspects of engagement. *excited*, *super*, *tomorrow* references affective and psychological states which suggest that people are happily engrossed in activities of life (Schaufeli et al., 2006). *meeting*, *conference*, *staff* hints at communal engagement, at work or at school, which is connected to the literature on communal engagement with its overlap with involvement, dedication, and organizational citizenship behavior (Appleton et al., 2006). The *bored*, *bore*, *text* topic is part of the same story: tapping people's disengagement from life, it emerges as one of the strongest negative predictors of SWL.

The *skills*, *management*, *business* topic tells a story about which workplaces are conducive to SWL. Previous research about county level SWL has revealed moderate

positive correlations between SWL and employment in the “professional” occupation sector (Lawless and Lucas, 2011), and from a theoretical psychological perspective (Ryan and Deci, 2000) it makes sense that SWL is high in high value-creation occupations in which continuous learning, roles of responsibility and skill development are valued. The set of curated lexica of Linguistic Inquiry and Word Count (LIWC) has a *work* lexicon, which emerges as the LIWC category with the highest correlation with SWL, hinting at a similar association.

The *family, friends, wonderful* topic suggests that a communal, pro-social orientation is correlated with the presence of SWL, which is a prediction made by the well-being literature. Psychologists have long believed that relationships are one of the strongest predictors of well-being (Tay et al., 2012), a construct sometimes alternatively referred to as “relatedness” (Ryan and Deci, 2000).

The *fucking, fuck, shit* topic emerges as the single strongest (negative) topic predictor of SWL. Such expressions of high negative affective arousal being strong negative predictors of SWL and well-being is a finding that has emerged robustly in our work across methodologies and datasets. The LIWC dictionaries tell a similar story, the strongest negative predictors of SWL are *negative emotions, anger* and *swear words*.

## 6.6 Conclusions

We presented the task of predicting well-being, a comprehensive construct of more than positive emotion, based on natural language use. We developed predictive models of well-being, as measured through the *satisfaction with life*(SWL) scale, over Facebook volunteers. Our models significantly out-predict baselines of happy/hedonic lexica (Kramer, 2010; Dodds et al., 2011). A cascaded model in which message-level predictions inform a user-level model gave the best performance.

We believe well-being prediction is an interesting and worthwhile task for the

computational linguistics community. Measurement of well-being is gaining interest as people and governments start to recognize that economic measures alone do not capture the welfare of societies. Methodologically there is much to explore, such as more sophisticated joint models of user and message-level information or the use of syntactic structure as features. Additionally, we suggest that language-based analyses of well-being need not end with *prediction*. Links between *SWL* and the everyday language in social-media enriches our *understanding* of Well-Being and its determinants, indicators and consequences.



## Bayesian Additive Regression Trees Implementation

By Adam Kapelner and Justin Bleich

### Abstract

We present a new package in R implementing Bayesian Additive Regression Trees (BART). The package introduces many new features for data analysis using BART such as variable selection, interaction detection, model diagnostic plots, incorporation of missing data and the ability to save trees for future prediction. It is significantly faster than the current R implementation, parallelized, and capable of handling both large sample sizes and high-dimensional data.

### 7.1 Introduction

Ensemble-of-trees methods have become popular choices for forecasting in both regression and classification problems. Algorithms such as Random Forests (Breiman, 2001b) and stochastic gradient boosting (Friedman, 2002) are two well-established and widely employed procedures. Recent advances in ensemble methods include Dynamic Trees (Taddy et al., 2011) and Bayesian Additive Regression Trees (BART, Chipman et al., 2010), which depart from predecessors in that they rely on an underlying

Bayesian probability model rather than a pure algorithm. **BART** has demonstrated substantial promise in a wide variety of simulations and real world applications such as predicting avalanches on mountain roads (Blattenberger and Fowles, 2014), predicting how transcription factors interact with DNA (Zhou and Liu, 2008) and predicting movie box office revenues (Eliashberg, 2010). This paper introduces **bartMachine**, a new R package that significantly expands the capabilities of using **BART** for data analysis.

Currently, there exists a single implementation of **BART** on CRAN: **BayesTree**, the package developed by the algorithm’s original authors. One of the major drawbacks of this implementation is its lack of a **predict** function. Test data must be provided as an argument during the training phase of the model. Hence it is impossible to generate forecasts on future data without re-fitting with the entire model. Since the run time is not trivial, forecasting becomes an arduous exercise. A significantly faster implementation of **BART** that contains master-slave parallelization exists as Pratola et al. (2013), but this is only available as standalone C++ source code and not integrated with R.

The goal of **bartMachine** is to provide a fast, easy-to-use, visualization-rich machine learning package for R users. Our implementation of **BART** is in **Java** and is integrated into R via **rJava** (Urbanek, 2011). From a runtime perspective, our algorithm is significantly faster and is parallelized, allowing computation on as many cores as desired. Not only is the model construction itself parallelized, but the additional features such as prediction, variable selection, and many others can be divided across cores as well.

Additionally, we include a variety of expanded and new features. We implement the ability to save trees in memory and provide convenience functions for prediction on test data. We also include plotting functions for both credible and predictive intervals and plots for visually inspecting convergence of **BART**’s Gibbs sampler. We

expand variable importance exploration to include permutation tests and interaction detection. We implement recently developed features for **BART** including a principled approach to variable selection and the ability to incorporate in prior information for covariates (Chapter 9 of this document, Bleich et al., 2013). We also implement the strategy found in Chapter 8 of this document (Kapelner and Bleich, 2013) to incorporate missing data during training and handle missingness during prediction.

In Section 11.2, we provide an overview of **BART** with a special emphasis on the features that have been extended. In Section 7.3 we provide a general introduction to the package, highlighting the novel features. Section 7.4 provides step-by-step examples of the regression capabilities and Section 7.5 introduces additional step-by-step examples of features unique to classification problems. We conclude in Section 13.5. Appendix A.2.1 discusses the details of our implementation and how it differs from **BayesTree**. Appendix A.2.5 offers predictive performance comparisons.

## 7.2 Overview of BART

**BART** provides a unique approach to nonparametric function estimation using regression trees. Regression trees rely on recursive binary partitioning of predictor space into a set of hyperrectangles in order to approximate some unknown function  $f$ . Such trees have received praise for their ability to flexibly fit interactions and nonlinearities. Models composed of sums of regression trees are able to capture additive effects in  $f$  better than single trees.

**BART** can be considered a sum-of-trees ensemble, with a novel estimation approach which relies on a fully Bayesian probability model. Specifically, the **BART** model can be expressed as:

$$\mathbf{Y} = f(\mathbf{X}) + \boldsymbol{\varepsilon} \approx \mathfrak{T}_1^{\text{leaf}}(\mathbf{X}) + \mathfrak{T}_2^{\text{leaf}}(\mathbf{X}) + \dots + \mathfrak{T}_m^{\text{leaf}}(\mathbf{X}) + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}_n(\mathbf{0}, \sigma^2 \mathbf{I}_n) \quad (7.1)$$

Here we have  $m$  distinct regression trees, each composed of a tree structure, denoted by  $\mathfrak{T}$ , and the parameters at the terminal nodes (also called leaves), denoted by  $\mathcal{L}$ . The two together, denoted as  $\mathfrak{T}^{\mathcal{L}}$  represents an entire tree with both its structure and set of leaf parameters.

The structure of a given tree  $\mathfrak{T}_t$  includes information on how any observation recurses down the tree. For each nonterminal (internal) node of the tree, there is a “splitting rule” taking the form  $\mathbf{x}_j < c$  consisting of the “splitting variable”  $\mathbf{x}_j$  and the “splitting value”  $c$ . An observation moves to the left daughter node if the condition set by the splitting rule is satisfied and to the right daughter node otherwise. The process continues until a terminal node is reached. Then, the observation receives the leaf value of the terminal node as its predicted value. We denote the set of tree’s leaf parameters as  $\mathcal{L}_t = \{\mu_{t,1}, \mu_{t,2}, \dots, \mu_{t,b_t}\}$  where  $b_t$  is the number of terminal nodes for a given tree.

BART can be distinguished from other ensemble-of-trees models due to its underlying probability model. As a Bayesian model, BART consists of a set of priors for the structure and the leaf parameters and a likelihood for data in the terminal nodes. The aim of the priors is to provide regularization, preventing any single regression tree from dominating the total fit.

We provide an overview of the BART priors and likelihood and then discuss how draws from the posterior distribution are made. A more complete exposition can be found in Chipman et al. (2010).

### 7.2.1 Priors and Likelihood

There are three priors for the BART model: a prior on the tree structure itself, a prior on the leaf parameters, and a prior on the error variance  $\sigma^2$ . The prior on  $\sigma^2$  is independent from the other two and each tree is independent, yielding:

$$\begin{aligned}
\mathbb{P}(\mathfrak{z}_1^{\text{leaf}}, \dots, \mathfrak{z}_m^{\text{leaf}}, \sigma^2) &= \left[ \prod_t \mathbb{P}(\mathfrak{z}_t^{\text{leaf}}) \right] \mathbb{P}(\sigma^2) = \left[ \prod_t \mathbb{P}(\mathcal{L}_t \mid \mathfrak{z}_t) \mathbb{P}(\mathfrak{z}_t) \right] \mathbb{P}(\sigma^2) \\
&= \left[ \prod_t \prod_{\ell} \mathbb{P}(\mu_{t,\ell} \mid \mathfrak{z}_t) \mathbb{P}(\mathfrak{z}_t) \right] \mathbb{P}(\sigma^2)
\end{aligned}$$

where the last line follows from an additional assumption of conditional independence of the leaf parameters given the tree’s structure.

The first prior is on the locations of nodes within the tree. Nodes at depth  $d$  are nonterminal with probability  $\alpha(1+d)^{-\beta}$  where  $\alpha \in (0, 1)$  and  $\beta \in [0, \infty]$ . This prior keeps the trees shallow, limiting complexity of any single tree. Default values for these hyperparameters of  $\alpha = 0.95$  and  $\beta = 2$  are recommended by Chipman et al. (2010).

For nonterminal nodes, splitting rules have the following prior. First, a predictor is randomly selected to serve as the splitting variable. In the original formulation, each available predictor is equally likely to be chosen, but this is relaxed in our implementation to allow an arbitrary discrete distribution (see Section 7.4.11). Then, the splitting value is selected by randomly choosing a value of the selected predictor with equal probability.

The third prior is on the leaf parameters. Given a tree with a set of terminal nodes, each terminal node (or leaf) has a continuous parameter (the leaf parameter) representing the “best guess” of the response in this partition of predictor space. This parameter is the fitted value assigned to any observation that lands in that node. The prior on each of the leaf parameters is given as:  $\mu_{\ell} \stackrel{iid}{\sim} \mathcal{N}(\mu_{\mu}, \sigma_{\mu}^2)$ . The expectation,  $\mu_{\mu}$ , is picked to be the range center,  $(y_{\min} + y_{\max})/2$ . The variance is empirically chosen so that the range center plus or minus  $k = 2$  variances cover 95% of the provided response values in the training set (by default). The aim of this prior is to provide model regularization by shrinking the leaf parameters towards the center of

the distribution of the response.

The final prior is on the error variance and is chosen to be  $\sigma^2 \sim \text{InvGamma}(\nu/2, \nu\lambda/2)$ .  $\lambda$  is determined from the data so that there is a  $q = 90\%$  a priori chance (by default) that the BART model will improve upon the RMSE from an ordinary least squares regression. Therefore, the majority of the prior probability mass lies below the RMSE from least squares regression. Additionally, this prior limits the probability mass placed on small values of  $\sigma^2$  to prevent overfitting.

Note that the adjustable hyperparameters are  $\alpha$ ,  $\beta$ ,  $k$ ,  $\nu$  and  $q$ . Default values generally provide good performance, but optimal tuning can be achieved via cross-validation, an automatic feature implemented and described in Section 7.4.2.

Along with a set of priors, BART specifies the likelihood of responses in the terminal nodes. They are assumed a priori Normal with the mean being the “best guess” in the leaf at the current moment (in the Gibbs sampler) and variance being the best guess of the variance at the moment i.e.  $\mathbf{y}_\ell \sim \mathcal{N}(\mu_\ell, \sigma^2/m)$ . Note that  $\sigma^2$  is scaled by the number of trees in order that the sum of the variances across the  $m$  trees is  $\sigma^2$ .

## 7.2.2 Posterior Distribution and Prediction

A Gibbs sampler (Geman and Geman, 1984) is employed to generate draws from the posterior distribution of  $\mathbb{P}(\mathfrak{F}_1^{\otimes}, \dots, \mathfrak{F}_m^{\otimes}, \sigma^2 \mid \mathbf{y})$ . A key feature of the Gibbs sampler for BART is to employ a form of “Bayesian backfitting” (Hastie and Tibshirani, 2000) where the  $j$ th tree is fit iteratively, holding all other  $m - 1$  trees constant by exposing only the residual response that remains unfitted:

$$\mathbf{R}_j := \mathbf{y} - \sum_{t \neq j} \mathfrak{F}_t^{\otimes}(\mathbf{X}). \quad (7.2)$$

The Gibbs sampler,

$$\begin{aligned}
1 : & \quad \mathfrak{T}_1 \mid \mathbf{R}_{-1}, \sigma^2 \\
2 : & \quad \varnothing_1 \mid \mathfrak{T}_1, \mathbf{R}_{-1}, \sigma^2 \\
3 : & \quad \mathfrak{T}_2 \mid \mathbf{R}_{-2}, \sigma^2 \\
4 : & \quad \varnothing_2 \mid \mathfrak{T}_2, \mathbf{R}_{-2}, \sigma^2 \\
& \quad \vdots \\
2m-1 : & \quad \mathfrak{T}_m \mid \mathbf{R}_{-m}, \sigma^2 \\
2m : & \quad \varnothing_m \mid \mathfrak{T}_m, \mathbf{R}_{-m}, \sigma^2 \\
2m+1 : & \quad \sigma^2 \mid \mathfrak{T}_1, \varnothing_1, \dots, \mathfrak{T}_m, \varnothing_m, \mathbf{E},
\end{aligned} \tag{7.3}$$

proceeds by first proposing a change to the first tree's structure  $\mathfrak{T}$  which are accepted or rejected via a Metropolis-Hastings step (Hastings, 1970). Note that sampling from the posterior of the tree structure does not depend on the leaf parameters, as they can be analytically margined out of the computation (see Appendix A.8.3.1). Given the tree structure, samples from the posterior of the  $b$  leaf parameters  $\varnothing_1 := \{\mu_1, \dots, \mu_b\}$  are then drawn. This procedure proceeds iteratively for each tree, using the updated set of partial residuals  $\mathbf{R}_j$ . Finally, conditional on the updated set of tree structures and leaf parameters, a draw from the posterior of  $\sigma^2$  is made based on the full model residuals  $\mathbf{E} := \mathbf{y} - \sum_{t=1}^m \mathfrak{T}_t^{\varnothing}(\mathbf{X})$ .

Within a given terminal node, since both the prior and likelihood are normally distributed, the posterior of each of the leaf parameters in  $\varnothing$  is conjugate normal with its mean being a weighted combination of the likelihood and prior parameters (lines 2, 4,  $\dots$ ,  $2m$  in equation set A.14). Due to the normal-inverse-gamma conjugacy, the posterior of  $\sigma^2$  is inverse gamma as well (line  $2m+1$  in equation set A.14). The complete expressions for these posteriors can be found in Gelman et al. (2004).

Lines 1, 3,  $\dots$ ,  $2m-1$  in equation set A.14 rely on Metropolis-Hastings draws from the posterior of the tree distributions. These involve introducing small perturbations to the tree structure: growing a terminal node by adding two daughter nodes, pruning two daughter nodes (rendering their parent node terminal), or changing a split rule. We denote these possible alterations as: GROW, PRUNE, and CHANGE.<sup>1</sup> The mathematics associated with the Metropolis-Hastings step is simple but is tedious, and we refer the interested reader to Appendix A.2.1 for the explicit calculations. Once again, over many Gibbs samples, trees can dynamically morph their structure in an effort to capture the fit left currently unexplained.

Pratola et al. (2013) argue that a CHANGE step is unnecessary for sufficient mixing of the Gibbs sampler. While we too observed this to be true for estimates of the posterior means, we found that omitting CHANGE can negatively impact the variable inclusion proportions (the feature introduced in Section 7.4.6). As a result, we implement a modified CHANGE where we only propose new splits for nodes that are singly internal: both children nodes are terminal nodes (details are given in Appendix A.8.3.3).

All  $2m + 1$  steps represent a *single* Gibbs iteration. We have observed that generally no more than 1,000 iterations are needed as “burn-in” (see Section 7.4.4 for convergence diagnostics). An additional 1,000 iterations is usually sufficient to serve as draws from the posterior for  $f(\mathbf{x})$ . A single predicted value  $\hat{f}(\mathbf{x})$  can be obtained by taking the average of the posterior values and a quantile estimate can be obtained by computing the appropriate quantile of the posterior values. Additional features of the posterior distribution will be discussed in Section 7.4.

---

<sup>1</sup>In the original formulation, Chipman et al. (2010) include an additional alteration called SWAP. Due to the complexity of bookkeeping associated with this alteration, we do not implement it.



### 7.2.3 BART for Classification

BART can easily be modified to handle classification problems for categorical response variables. In Chipman et al. (2010), only binary outcomes were explored but recent work has extended BART to the multiclass problem (Kindo et al., 2013). Our implementation handles binary classification and we plan to implement multiclass outcomes in a future release.

For the binary classification problem (coded with outcomes “0” and “1”), we assume a probit model,

$$\mathbb{P}(Y = 1 \mid \mathbf{X}) = \Phi \left( \mathfrak{F}_1^{\text{leaf}}(\mathbf{X}) + \mathfrak{F}_2^{\text{leaf}}(\mathbf{X}) + \dots + \mathfrak{F}_m^{\text{leaf}}(\mathbf{X}) \right),$$

where  $\Phi$  denotes the cumulative density function of the standard normal distribution. In this formulation, the sum-of-trees model serves as an estimate of the conditional probit at  $\mathbf{x}$  which can be easily transformed into a conditional probability estimate of  $Y = 1$ .

In the classification setting, the prior on  $\sigma^2$  is not needed as the model assumes  $\sigma^2 = 1$ . The prior on the tree structure remains the same as in the regression setting and a few minor modifications are required for the prior on the leaf parameters.

Sampling from the posterior distribution is again obtained via Gibbs sampling with a Metropolis-Hastings step outlined in Section 7.2.2. Following the data augmentation approach of (Albert and Chib, 1993) an additional vector of latent variables  $\mathbf{Z}$  is introduced into the Gibbs sampler. Then, a new step is created in the Gibbs sampler where draws of  $\mathbf{Z} \mid \mathbf{y}$  are obtained by conditioning on the sum-of-trees model:

$$Z_i \mid y_i = 1 \sim \max N \left( \sum_t \mathfrak{F}_t^{\text{leaf}}(\mathbf{X}), 1 \right), 0 \text{ and}$$

$$Z_i \mid y_i = 0 \sim \min \left\{ N \left( \sum_t \mathfrak{F}_t^{\otimes}(\mathbf{X}), 1 \right), 0 \right\}.$$

Next,  $\mathbf{Z}$  is used as the response vector instead of  $\mathbf{y}$  in all steps of Equation A.14.

Upon obtaining a sufficient number of samples from the posterior, inferences can be made using the the posterior distribution of conditional probabilities and classification can be undertaken by applying a threshold to the to the means (or another summary) of these posterior probabilities. The relevant classification features of `bartMachine` are discussed in Section 7.5.

## 7.3 The `bartMachine` package

The package `bartMachine` provides a novel implementation of Bayesian Additive Regression Trees in R. The algorithm is substantially faster than the current R package `BayesTree` and our implementation is parallelized at the Gibbs sample level during prediction. Additionally, the interface with `rJava` allows for the entire posterior distribution of tree ensembles to persist throughout the R session, allowing for prediction and other calls to the trees without having to re-run the Gibbs sampler (a limitation in the current implementation). The model object cannot persist across sessions (using R's `save` command for instance) and we view the addition of this feature as future work. Since our implementation is different from `BayesTree`, we provide a predictive accuracy bakeoff on different datasets in Appendix A.2.5 which illustrates that the two are about equal.

### 7.3.1 Speed Improvements and Parallelization

We make a number of significant speed improvements over the original implementation.

First, **bartMachine** is fully parallelized (with the number of cores customizable) during model creation, prediction, and many of the other features. During model creation, we chose to parallelize by creating one independent Gibbs chain per core. Thus, if we have 500 burn-in samples and 1,000 post burn-in samples and four cores, each core would sample 750 samples: 500 for a burn-in and 250 post burn-in samples. The final model will aggregate the 250 post burn-in samples for the four cores yielding the desired 1,000 total post burn-in samples. This has the drawback of effectively running the burn-in serially, but has the added benefit of reducing auto-correlation of the sum-of-trees samples in the posterior samples since the chains are independent which may provide greater predictive performance. Parallelization at the level of likelihood calculations is left for a future release. Parallelization for prediction and other features scale linearly in the number of cores.

Additionally, we take advantage of a number of additional computational short-cuts:

1. Computing the unfitted responses for each tree (Equation 7.2) can be accomplished by keeping a running vector and making entry-wise updates as the Gibbs sampler (Equation A.14) progresses from step 1 to  $2m$ . Additionally, during the  $\sigma^2$  step  $2m + 1$ , the residuals do not have to be computed by dropping the data down all the trees.
2. Each node caches its acceptable variables for split rules and the acceptable unique split values so they do not need to be calculated at each tree sampling step. This speed enhancement, which we call *memcache* comes at the expense of memory and may cause issues for large data sets. We include a toggle in our implementation defaulted to “on.”
3. Careful calculations in Appendix A.2.1 eliminate many unnecessary computations. For instance, the likelihood ratios are only functions of the squared sum of re-

sponses and no longer require computing the sum of the responses squared.

Figure 7.1 displays model creation speeds for different values of  $n$  on a linear model with  $p = 20$ , normally distributed covariates,  $\beta_1, \dots, \beta_{20} \stackrel{iid}{\sim} U(-1, 1)$ , and standard normal noise. Note that we do not vary  $p$  as it was already shown in Chipman et al. (2010) that **BART**'s computation time is largely unaffected by the dimensionality of the problem (relative to the influence of sample size). We include results for **BART** using **BayesTree** (Chipman et al., 2010), **bartMachine** with one and four cores, the *memcache* option on and off, as well as four cores, *memcache* off and computation of in-sample statistics off (all with  $m = 50$  trees). We also include Random Forests via the package **randomForest** (Liaw and Wiener, 2002) with its default settings.

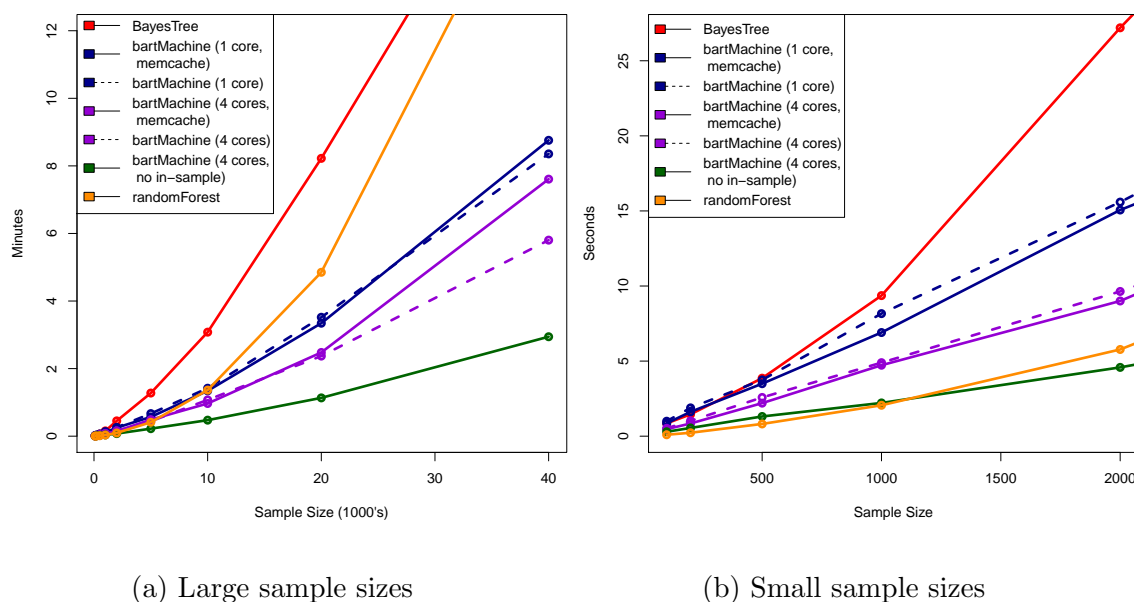


Figure 7.1: Model creation times as a function of sample size for a number of settings of **bartMachine**, **BayesTree** and **RandomForests**. Simulations were run on a quad-core 3.4GHz Intel i5 desktop with 24GB of RAM running the Windows 7 64bit operating system.

We first note that Figure 7.1a demonstrates that the **bartMachine** model creation

runtime is approximately linear in  $n$ . There is about a 30% speed-up when using four cores instead of one. The *memcache* enhancement should be turned off only with sample sizes larger than  $n = 20,000$ . Noteworthy is the 50% reduction in time of constructing the model when not computing in-sample statistics. In-sample statistics are computed by default because the user generally wishes to see them. Also, for the purposes of this comparison, **BayesTree** models compute the in-sample statistics by necessity since the trees are not saved. The **randomForest** implementation becomes slower just after  $n = 1,000$  due to its reliance on a greedy exhaustive search at each node.

Figure 7.1b displays results for smaller sample sizes ( $n \leq 2,000$ ) that are often encountered in practice. We observe the *memcache* enhancement provides about a 10% speed improvement. Thus, if memory is an issue, it can be turned off with little performance degradation.

### 7.3.2 Missing Data in BART

**bartMachine** implements a native method for incorporating missing data into both model creation and future prediction with test data. The details are given in Chapter 8 of this document (Kapelner and Bleich, 2013) but we provide a brief summary here.

There are a number of ways to incorporate missingness into tree-based methods (see Ding and Simonoff, 2010 for a review). The method implemented here is known as “Missing Incorporated in Attributes” (MIA, Twala et al., 2008, section 2) which natively incorporates missingness by augmenting the nodes’ splitting rules to (a) also handle sorting the missing data to the left or right and (b) use missingness *itself* as a variable to be considered in a splitting rule. Algorithm 2 summarizes these new splitting rules as they are implemented within the package.

Implementing MIA into the BART procedure is straightforward. These new split-

ting rules are sampled uniformly during the GROW or CHANGE steps. For example, a splitting rule might be “ $\mathbf{x}_j < c$  or  $\mathbf{x}_j$  is missing.” To account for splitting on missingness itself, we create dummy vectors of length  $n$  for each of the  $p$  attributes, denoted  $\mathbf{M}_1, \dots, \mathbf{M}_p$ , which assume the value 1 when the entry is missing and 0 when the entry is present. The original training matrix is then augmented with these dummies, giving the opportunity to select missingness *itself* when choosing a new splitting rule during the grow or change steps. Note that this can increase the number of predictors by up to a factor of 2. We illustrate the building a BART model with missing data in Section 7.4.9. As described in Chipman et al. (2010, Section 6), BART’s runtime increases negligibly in the number of covariates and this has been our experience using the augmented training matrix.

---

**Algorithm 1** The MIA choices for all attributes  $j \in \{1, \dots, p\}$  and all split points  $x_{ij}^*$  where  $i \in \{1, \dots, n\}$  during a GROW or CHANGE step in `bartMachine`.

---

- 1: If  $x_{ij}$  is missing, send it  $\leftarrow$ ; if it is present and  $x_{ij} \leq x_{ij}^*$ , send it  $\leftarrow$ , otherwise  $\rightarrow$ .
  - 2: If  $x_{ij}$  is missing, send it  $\rightarrow$ ; if it is present and  $x_{ij} \leq x_{ij}^*$ , send it  $\leftarrow$ , otherwise  $\rightarrow$ .
  - 3: If  $x_{ij}$  is missing, send it  $\leftarrow$ ; if it is present, send it  $\rightarrow$ .
- 

### 7.3.3 Principled Variable Selection

Our package also implements the variable selection procedures developed in Chapter 9 of this document (Bleich et al., 2013), which is best applied to data problems where the number of covariates influencing the response is small relative to the total number of covariates. We give a brief summary of the procedures here.

In order to select variables, we make use of the “variable inclusion proportions,” the proportion of times each predictor is chosen as a splitting rule divided by the total

number of splitting rules appearing in the model (see Section 7.4.6 for more details). The variable selection procedure can be outlined as follows:

1. Compute the model’s variable inclusion proportions.
2. Permute the response vector, thereby breaking the relationship between the covariates and the response. Rebuild the model and compute the “null” variable inclusion proportions. Repeat this a number of times to create a null permutation distribution.
3. Three selection rules are can be used depending on the desired stringency of selection:
  - (a) Local Threshold: Include a predictor  $\mathbf{x}_k$  if its variable inclusion proportion exceeds the  $1 - \alpha$  quantile of its own null distribution.
  - (b) Global Max Threshold: Include a predictor  $\mathbf{x}_k$  if its variable inclusion proportion exceeds the  $1 - \alpha$  quantile of the distribution of the maximum of the null variable inclusion proportions from each permutation of the response.
  - (c) Global SE Threshold: Select  $\mathbf{x}_k$  if its variable inclusion proportion exceeds a threshold based from its own null distribution mean and SD with a global multiplier shared by all predictors.

The Local procedure is the least stringent in terms of selection and the Global Max procedure the most. The Global SE procedure is a compromise, but behaves more similarly to the Global Max. Chapter 9 of this document (Bleich et al., 2013) demonstrates that the best procedure depends on the underlying sparsity of the problem, which is often unknown. Therefore, the authors include an additional procedure that chooses the best of these thresholds via cross-validation and this method is also implemented in `bartMachine`. Examples of these procedures for variable selection are provided in Section 7.4.10.

## 7.4 Regression Features

We illustrate the package features by using both real and simulated data, focusing first on regression problems.

### 7.4.1 Computing parameters

We first set some computing parameters. We allow up to 5GB of RAM for the Java heap (although we never used more than 1GB during this paper’s exploration)<sup>2</sup> and we set the number of computing cores available for use to 4.

```
> library(bartMachine)
> set_bart_machine_num_cores(4)
> init_java_for_bart_machine_with_mem_in_mb(5000)
```

The following Sections 7.4.2 – 7.4.10 use a dataset obtained from UCI (Bache and Lichman, 2013). The  $n = 201$  observations are automobiles and the goal is to predict each automobile’s price from 25 features (15 continuous and 10 nominal), first explored by Kibler et al. (1989).<sup>3</sup> This dataset also contains missing data.

### 7.4.2 Model Building

We are now ready to construct a **bartMachine** model. The default hyperparameters generally follow the recommendations of Chipman et al. (2010) and provide a ready-to-use algorithm for many data problems. Our hyperparameter settings are

---

<sup>2</sup>Note that the maximum amount of memory can be set only *once* at the beginning of the R session (a limitation of **rJava** since only one Java Virtual Machine can be initiated per session), but the number of cores can be respecified at any time.

<sup>3</sup>We first preprocess the data. We first drop one of the nominal predictors (car company) due to too many categories (22). We then coerce two of the of the nominal predictors to be continuous. Further, the response variable, price, was logged to reduce right skew in its distribution.



$m = 50$ ,<sup>4</sup>  $\alpha = 0.95$ ,  $\beta = 2$ ,  $k = 2$ ,  $q = 0.9$ ,  $\nu = 3$ , probabilities of the GROW / PRUNE / CHANGE steps is 39% / 39% / 44%. We set the number of burn-in Gibbs samples to 250 and number of post-burn-in samples to 1,000. We default the missing data feature to be off. We default the covariates to be equally important *a priori*. Other parameters and their defaults can be found in the package’s online manual. Below is a default `bartMachine` model. Here,  $\mathbf{X}$  denotes automobile attributes and  $\mathbf{y}$  denotes the log price of the automobile.

```
> bart_machine = build_bart_machine(X, y)
Building BART for regression ... evaluating in sample data...done
```

If one wishes to see the iterations of the Gibbs sampler of Equation A.14, the flag `verbose` can be set to “TRUE”. One can see more debug information from the Java program by setting the flag `debug_log` to true and the program will print to `unnamed.log` in the current working directory. We now inspect the model object to query its in-sample performance and to be reminded of the input data and model hyperparameters.

Since the response was considered continuous, we employ BART for regression. The dimensions of the design matrix are given. Note that we dropped 45 observations that contained missing data (which we will retain in Section 7.4.9). We then have a recording of the MSE for the OLS model and our average estimate of  $\sigma_e^2$ . We are then given in-sample statistics on error. Pseudo- $R^2$  is calculated via  $1 - SSE/SST$ . Also provided are outputs from tests of the error distribution being mean centered and normal. In this case, we cannot conclude normality of the residuals using the Shapiro-Wilk test.

---

<sup>4</sup>In contrast to Chipman et al. (2010), we recommend this default as a good starting point rather than  $m = 200$  due to our experience experimenting with the “RMSE by number of trees” feature found in later in this section. Performance is often similar and computational time and memory requirements are dramatically reduced.

```

> bart_machine
Bart Machine v1.0b for regression

training data n = 160 and p = 46
built in 1 secs on 4 cores, 50 trees, 250 burn in and 1000 post. samples

sigsq est for y beforehand: 0.014
avg sigsq estimate after burn-in: 0.00886

in-sample statistics:
  L1 = 9.03
  L2 = 0.8
  rmse = 0.07
  Pseudo-Rsq = 0.9741
p-val for shapiro-wilk test of normality of residuals: 0.01785
p-val for zero-mean noise: 0.97692

```

Figure 7.2: The summary for the default `bartMachine` model built with the automobile data

We can also obtain out-of-sample statistics to assess level of overfitting by using k-fold cross validation. Using 10 folds we find:

```

> k_fold_cv(X, y, k_folds = 10)

$L1_err      $L2_err      $rmse      $PseudoRsq
[1] 22.63155    [1] 5.202831    [1] 0.1803266  [1] 0.831917

```

The Pseudo- $R^2$  being lower out-of-sample versus in-sample suggests evidence that BART is slightly overfitting (note that the training sample during cross-validation is 10% smaller).

It may also be of interest how the number of trees  $m$  affects performance. One can examine how out-of-sample predictions vary by the number of trees via

```

> rmse_by_num_trees(bart_machine, num_replicates = 20)

```

and the output is shown in Figure 7.3.

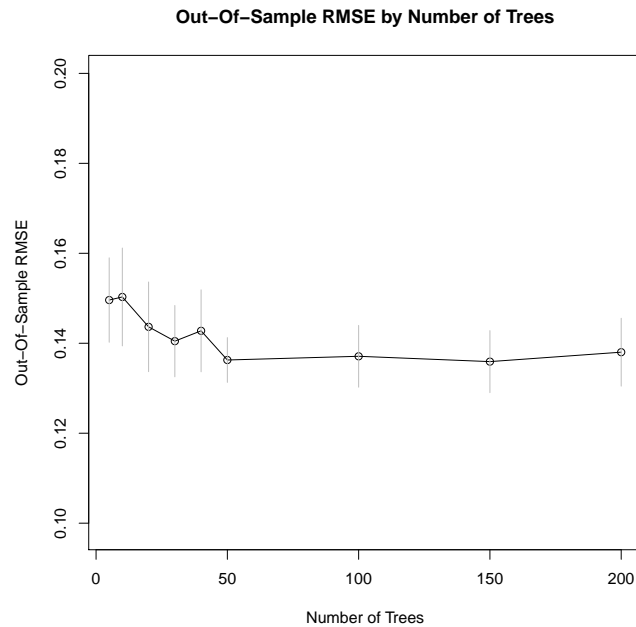


Figure 7.3: Out-of-sample predictive performance by number of trees

It seems that increasing  $m > 50$  does not result in any substantial increase in performance. We can now try to build a better `bartMachine` by grid-searching over a set of hyperparameter combinations, including  $m$  (for more details, see BART-cv in Chipman et al., 2010). The default grid search is small and it can be customized by the user.

```
> bart_machine_cv = build_bart_machine_cv(X, y)
```

```
...
```

```
BART CV win: k: 2 nu, q: 10, 0.75 m: 200
```

This function returns the “winning” model, which is the one with lowest out-of-sample RMSE over a 5-fold cross-validation. Here, the cross-validated `bartMachine` model has slightly better in-sample performance ( $L1 = 8.18$ ,  $L2 = 0.68$  and Pseudo- $R^2 = 0.978$ ) as well as slightly better out-of-sample performance:

```
> k_fold_cv(X, y, k_folds = 10, k = 2, nu = 3, q = 0.9, num_trees = 200)
```

<code>\$L1_err</code>	<code>\$L2_err</code>	<code>\$rmse</code>	<code>\$PseudoRsq</code>
<code>[1] 21.21557</code>	<code>[1] 4.517916</code>	<code>[1] 0.1680386</code>	<code>[1] 0.8540439</code>

Predictions are handled with the `predict` function:

```
predict(bart_machine_cv, X[1 : 14, ])
      9.479963  9.775766  9.799110 10.050041  9.659138  9.697902  9.873622
      9.931972  8.550436  8.688874  8.794351  8.647986  8.690300  9.029066
```

We also include a convenience method `bart_predict_for_test_data` that will predict and return out-of-sample error metrics when the test outcomes are known.

### 7.4.3 Model Destroying

As noted in the introduction to Section 7.3, `bartMachine` objects persist in **Java** for the entirety of an **R** session. These model objects can use a substantial amount of RAM and it is prudent to release this memory when it is no longer needed. Simply removing a `bartMachine` in **R** does *not* destroy the **Java** object and release the RAM (resulting in a memory leak).

Therefore, we provide a utility function `destroy_bart_machine` that cleans up the **Java** object. This function should be called when a `bartMachine` object is no longer needed and *before* removing or overwriting the **R** variable. Since we no longer are using the original `bart_machine` object, we now release its memory via:

```
destroy_bart_machine(bart_machine)
```

### 7.4.4 Assumption Checking

The package includes features that assess the plausibility of the **BART** model assumptions. Checking the mean-centeredness of the noise is addressed in the summary

output of Figure 7.2 and is simply a one-sample  $t$ -test of the average residual value against a null hypothesis of true mean zero. We assess both normality and heteroskedasticity via:

```
> check_bart_error_assumptions(bart_machine_cv)
```

This will display a window similar to Figure 7.4 which contains a QQ-plot (to assess normality) as well as a residual-by-predicted plot (to assess homoskedasticity). It appears that the errors are most likely normal and homoskedastic.

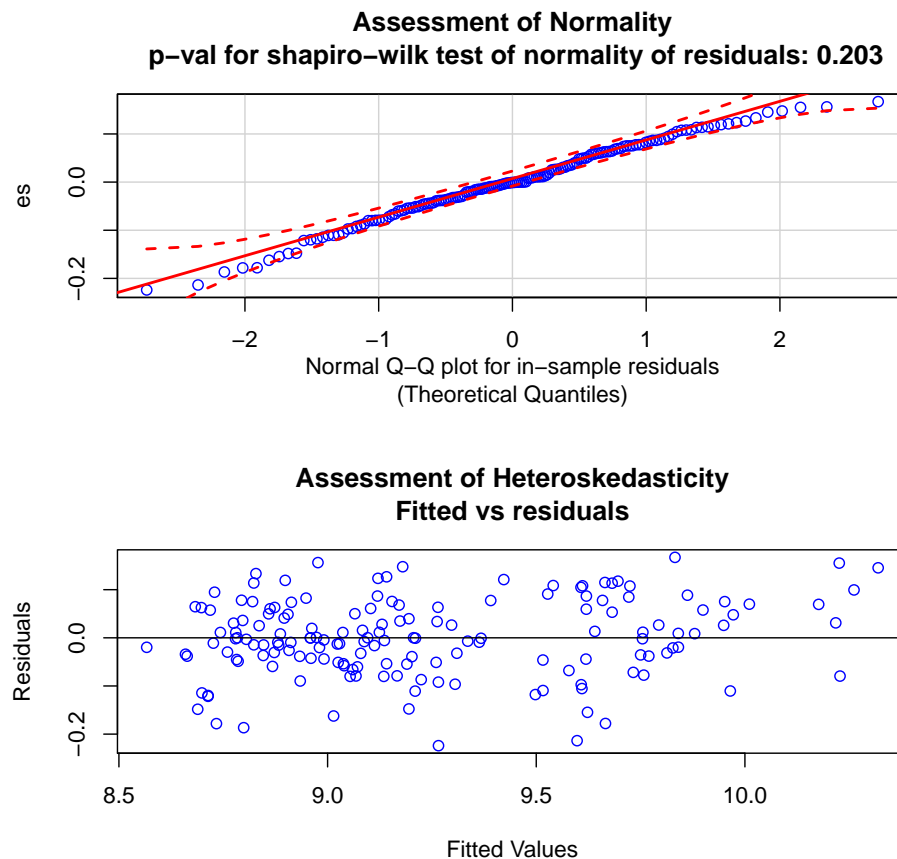


Figure 7.4: Test of normality of errors using QQ-plot and the Shapiro-Wilk test (top), residual plot to assess heteroskedasticity (bottom).

In addition to the model assumptions, BART requires convergence of its Gibbs sampler. Figure 7.5 displays four types of convergence diagnostics.

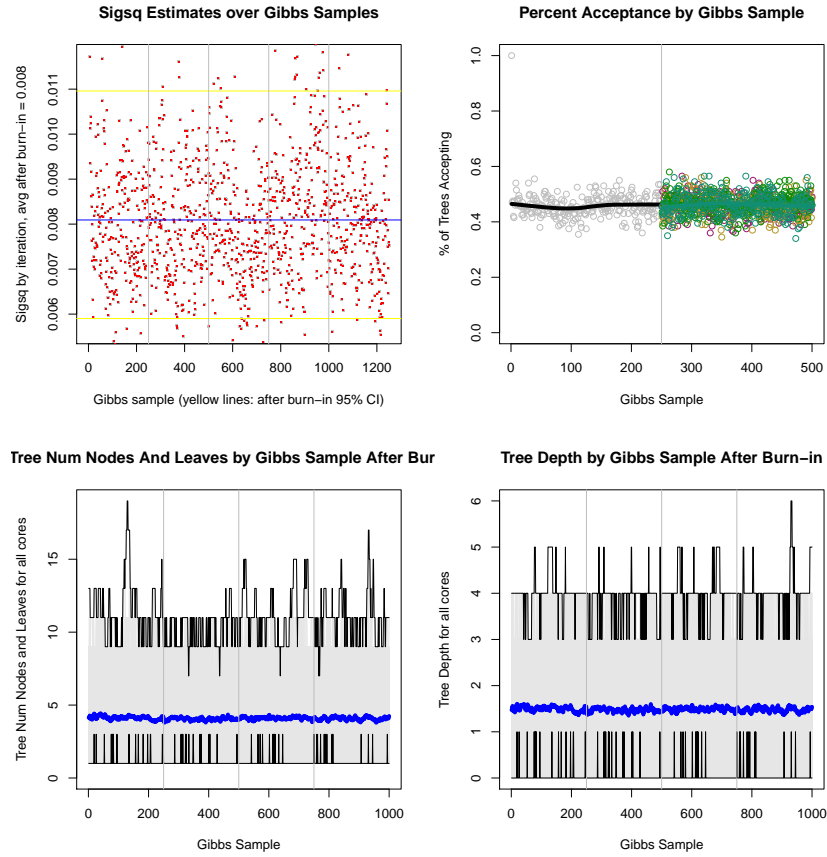


Figure 7.5: Convergence diagnostics for the cross-validated `bartMachine` model. Top left:  $\sigma^2$  by Gibbs sample. Samples to the left of the first vertical grey line are burn-in from the first computing core's Gibbs sample chain. The four subsequent plots separated by grey lines are the post-burn-in samples from each of the four computing cores employed during model construction. Top right: percent acceptance of Metropolis-Hastings proposals across the  $m$  trees where each point plots one iteration. Points before the grey vertical line illustrate burn-in samples and points after illustrate post burn-in. Each computing core is colored differently. Bottom left: average number of leaves across the  $m$  trees by sample (post burn-in only where computing cores separated by vertical grey lines). Bottom right: average tree depth across the  $m$  trees by sample (post burn-in only where computing cores separated by vertical grey lines).

### 7.4.5 Credible Intervals and Prediction Intervals

An advantage of **BART** is that if we believe the priors and model assumptions, the Bayesian probability model and corresponding burned-in Gibbs samples provide the approximate posterior distribution of  $f(\mathbf{x})$ . Thus, one can compute uncertainty estimates via quantiles of the posterior samples. These provide Bayesian “credible intervals” which are intervals for the conditional expectation function,  $\mathbb{E}[\mathbf{y} \mid \mathbf{X}]$ .

Another useful uncertainty interval can be computed for individual predictions by combining uncertainty from the conditional expectation function with the systematic, homoskedastic normal noise produced by  $\mathcal{E}$ . Since we have draws from the posterior of the conditional expectation distribution and concomitant draws from the posterior of the variance distribution, we can simulate the distribution of the response by drawing many realizations from  $\mathcal{N}(f(\mathbf{x})_n, \sigma_n^2)$  for each the post-burn-in samples and aggregating them. The prediction interval is then provided by the appropriate quantiles of the posterior samples.

Below is an example of how both types of intervals are computed in the package (for the 100th observation of the training data):

```
> calc_credible_intervals(bart_machine_cv, X[100, ], ci_conf = 0.95)
      ci_lower_bd ci_upper_bd
[1,]      8.725202      8.971687
> calc_prediction_intervals(bart_machine_cv, X[100, ], pi_conf = 0.95)
      pi_lower_bd pi_upper_bd
[1,]      8.631243      9.06353
```

Note that the prediction intervals are wider than the credible intervals because they reflect the uncertainty from the error term.

We can then plot these intervals in sample:

```
> plot_y_vs_yhat(bart_machine_cv, credible_intervals = TRUE)
```

```
> plot_y_vs_yhat(bart_machine_cv, prediction_intervals = TRUE)
```

Figure 7.6a shows how our prediction fared against the original response (in-sample) with 95% credible intervals. Figure 7.6b shows the same prediction versus the original response plot now with 95% prediction intervals.

## 7.4.6 Variable Importance

After a BART model is built, it is natural to ask the question: which variables are most important? This is assessed by examining the splitting rules in the  $m$  trees across the post burn-in gibbs samples which are known as “inclusion proportions” (Chipman et al., 2010). The inclusion proportion for any given predictor represents the proportion of times that variable is chosen as a splitting rule out of all splitting rules among the posterior draws of the sum-of-trees model. Figure 7.7 illustrates the inclusion proportions for all variables obtained via:

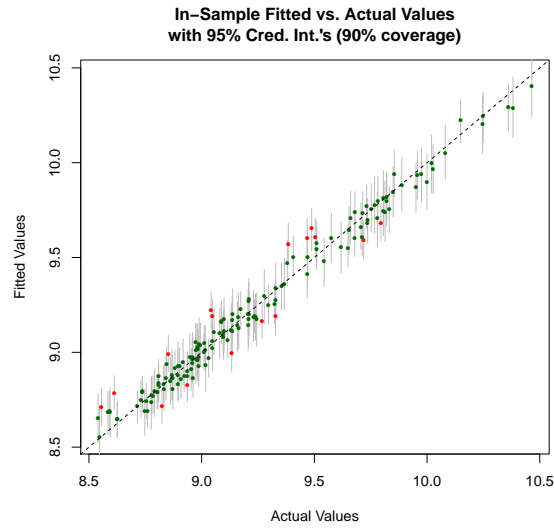
```
> investigate_var_importance(bart_machine_cv,  
  num_replicates_for_avg = 20)
```

Actual selection of variables *significantly* affecting the response is addressed conceptually in Section 7.3.3 and examples are provided in Section 7.4.10.

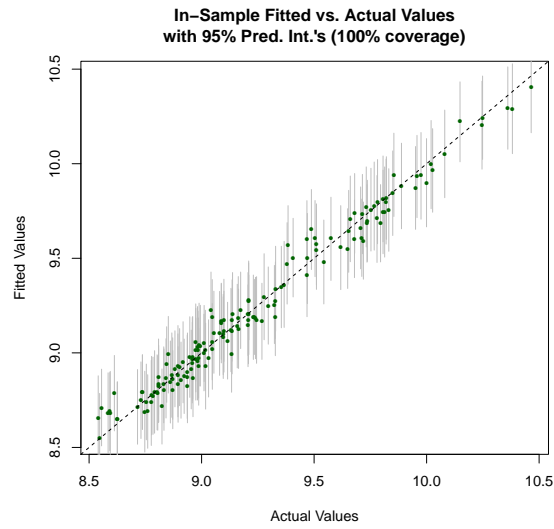
## 7.4.7 Variable Effects

It is also natural to ask: does  $\mathbf{x}_j$  affect the response, controlling for other variables in the model? This is roughly analogous to the  $t$ -test in ordinary least squares regression of no linear effect of  $\mathbf{x}_j$  on  $\mathbf{y}$  while controlling for  $\mathbf{x}_{-j}$ . The null hypothesis here is the same but the linearity constraint is relaxed. To test this, we employ a permutation approach where we record the observed Pseudo- $R^2$  from the `bartMachine` model built with the original data. Then we permute the  $\mathbf{x}_j$ th column, thereby destroying any





(a) Segments illustrate credible intervals



(b) Segments illustrate prediction intervals

Figure 7.6: Fitted versus actual response values for the automobile dataset. Segments are 95% credible intervals (a) or 95% prediction intervals (b). Green dots indicate the true response is within the stated interval and red dots indicate otherwise. Note that the percent coverage in (a) is not expected to be 95% because the response includes a noise term.

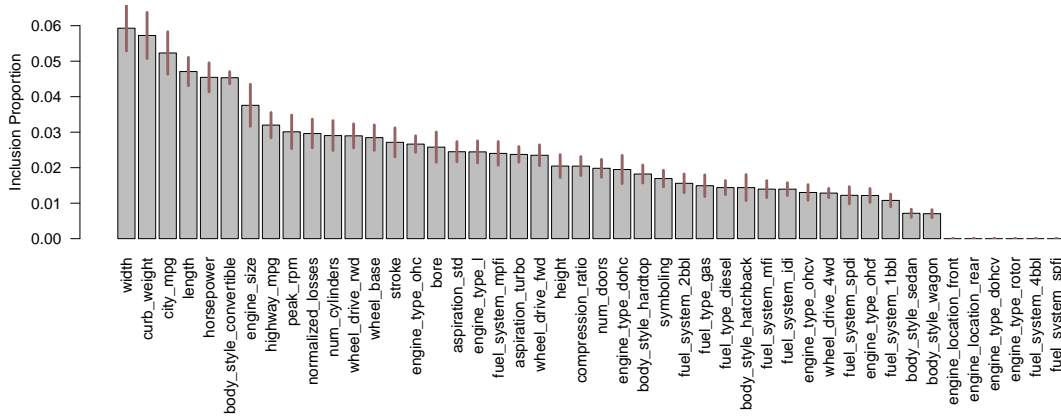


Figure 7.7: Average variable inclusion proportions in the cross-validated `bartMachine` model for the automobile data averaged over 100 model constructions to obtain stable estimates across many posterior modes in the sum-of-trees distribution (as recommended in Chapter 9 of this document, Bleich et al., 2013). The segments atop the bars represent 95% confidence intervals. The eight predictors with inclusion proportions of zero feature identically one value (after missing data was dropped).

relationship between  $\mathbf{x}_j$  and  $\mathbf{y}$ , construct a new duplicate `bartMachine` model from this permuted design matrix and record a “null” Pseudo- $R^2$ . We then repeat this process to obtain a null distribution of Pseudo- $R^2$ ’s. Since the alternative hypothesis is that  $\mathbf{x}_j$  has an effect on  $\mathbf{y}$  in terms of predictive power, our  $p_{\text{val}}$  is the proportion of null Pseudo- $R^2$ ’s greater than the observed Pseudo- $R^2$ , making our procedure a natural one-sided test. Note, however, that this test is conditional on the `BART` model and its selected priors being true, similar to the assumptions of the linear model.

If we wish to test if a set of covariates  $A \subset \{\mathbf{x}_1, \dots, \mathbf{x}_p\}$  affect the response after controlling for other variables, we repeat the procedure outlined in the above paragraph by permuting the columns of  $A$  in every null sample. This is roughly analogous to the partial  $F$ -test in ordinary least squares regression.

If we wish to test if *any* of the covariates matter in predicting  $\mathbf{y}$ , roughly analogous

to the omnibus  $F$ -test in ordinary least squares regression, we simply permute  $\mathbf{y}$  during the null sampling. This procedure breaks the relationship between the response and the predictors but does not alter the existing associations between predictors.

At  $\alpha = 0.05$ , Figure 7.8a demonstrates an insignificant effect of the variable `width` of car on price. Even though `width` is putatively the “most important” variable as measured by proportions of splits in the posterior sum-of-trees model (Figure 7.7), note that this is largely an easy prediction problem with many collinear predictors. Figure 7.8b shows the results of a test of the putatively most important categorical variable, `body style` (which involves permuting the categories, then dummifying the levels to preserve the structure of the variable). We find a marginally significant effect ( $p = 0.0495$ ). A test of the top ten most important variables is convincingly significant (Figure 7.8c). For the omnibus test, Figure 7.8d illustrates an extremely statistically significant result, as would be expected. The code to run these tests is shown below (output suppressed).

```
> cov_importance_test(bart_machine_cv, covariates = c("width"))
> cov_importance_test(bart_machine_cv, covariates = c("body_style"))
> cov_importance_test(bart_machine_cv, covariates = c("width",
  "curb_weight", "city_mpg", "length", "horsepower", "body_style",
  "engine_size", "highway_mpg", "peak_rpm", "normalized_losses"))
> cov_importance_test(bart_machine_cv)
```

## 7.4.8 Partial Dependence

A data analyst may also be interested in understanding how  $\mathbf{x}_j$  affects the response on average, after controlling for other predictors. This can be examined using Friedman (2001)’s Partial Dependence Function (PDP),

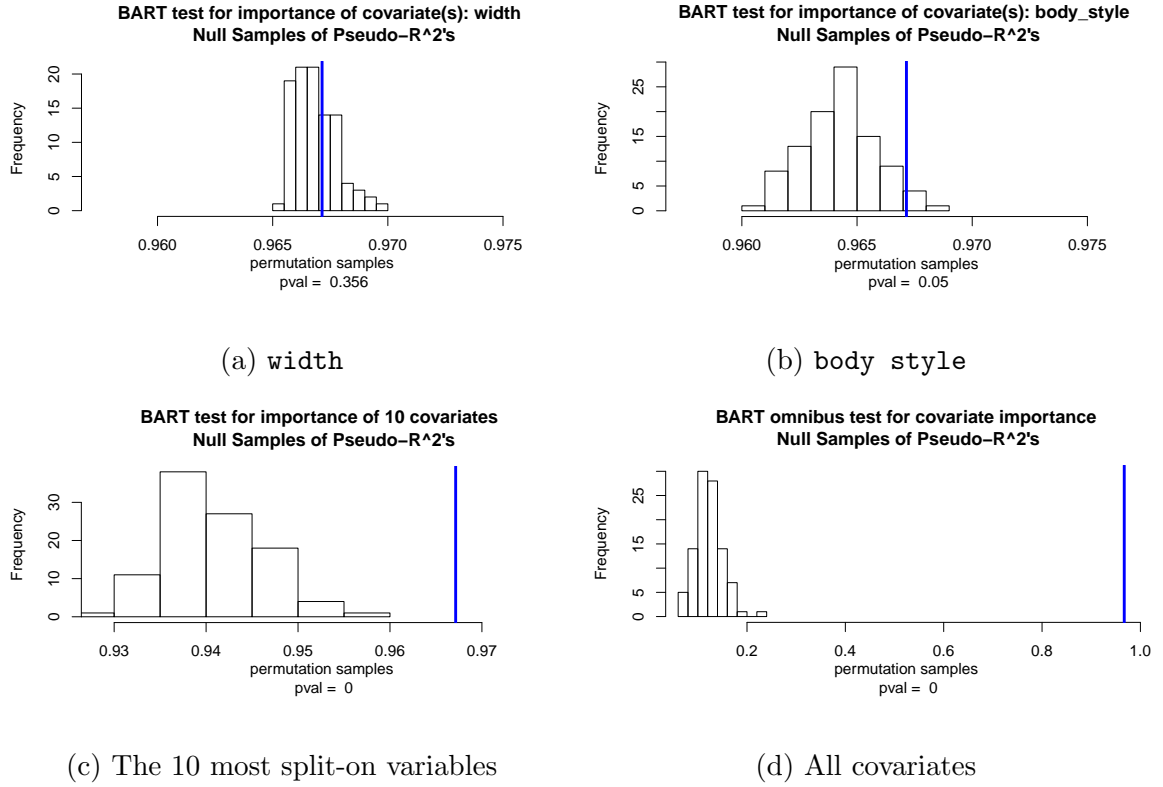


Figure 7.8: Tests of covariate importance conditional on the cross-validated `bartMachine` model. All tests performed with 100 null samples.

$$f_j(\mathbf{x}_j) = \mathbb{E}_{\mathbf{x}_{-j}} [f(\mathbf{x}_j, \mathbf{x}_{-j})] = \int f(\mathbf{x}_j, \mathbf{x}_{-j}) dP(\mathbf{x}_{-j}), \quad (7.4)$$

where  $\mathbf{x}_{-j}$  denotes all variables except  $\mathbf{x}_j$ . The PDP of predictor  $\mathbf{x}_j$  gives the average value of  $f$  when  $\mathbf{x}_j$  is fixed and  $\mathbf{x}_{-j}$  varies over its marginal distribution,  $dP(\mathbf{x}_{-j})$ . As neither the true model  $f$  nor the distribution of the predictors  $dP(\mathbf{x}_{-j})$  are known, we estimate Equation 11.1 by computing

$$\hat{f}_j(\mathbf{x}_j) = \frac{1}{n} \sum_{i=1}^n \hat{f}(\mathbf{x}_j, \mathbf{x}_{-j,i}) \quad (7.5)$$

where  $n$  is the number of observations in the training data and  $\hat{f}$  denotes the `bartMachine` model. Since `BART` provides an estimated posterior distribution, we can plot credible bands for the PDP function. In Equation 7.5, the  $\hat{f}$  can be replaced with a function that calculates the  $q$ th quantile of the post-burned-in Gibbs samples for  $\hat{\mathbf{y}}$ . Figure 7.9a plots the PDP along with the 2.5%ile and the 97.5%ile for the variable `horsepower`. By varying over most of the range of `horsepower`, the price is predicted to increase by about \$1000. Figure 7.9b plots the PDP along with the 2.5%ile and the 97.5%ile for the variable `stroke`. This predictor seemed to be relatively unimportant according to Figure 7.7 and the PDP confirms this, with a very small, yet nonlinear average partial effect. The code for both plots is below.

```
> pd_plot(bart_machine_cv, j = "horsepower")
> pd_plot(bart_machine_cv, j = "stroke")
```

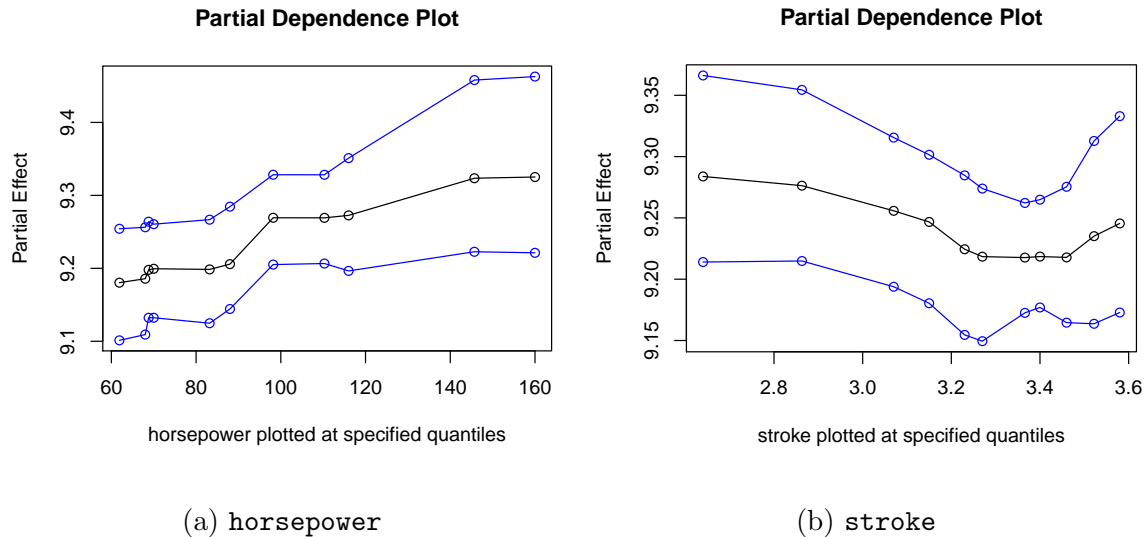


Figure 7.9: PDPs plotted in black and 95% credible intervals plotted in blue for variables in the automobile dataset. Points plotted are at the 5%ile, 10%ile, 20%ile, ..., 90%ile and 95%ile of the values of the predictor. Lines plotted between the points approximate the PDP by linear interpolation.

### 7.4.9 Incorporating Missing Data

The procedure for incorporating missing data was introduced in Section 7.3.2. We now build a `bartMachine` model using this procedure below:

```
> bart_machine = build_bart_machine(X, y, use_missing_data = TRUE,
  use_missing_data_dummies_as_covars = TRUE)
> bart_machine
Bart Machine v1.0b for regression

Missing data feature ON
training data n = 201 and p = 50
built in 1.3 secs on 1 core, 50 trees, 250 burn-in and 1000 post. samples

sigsq est for y beforehand: 0.016
avg sigsq estimate after burn-in: 0.01055

in-sample statistics:
  L1 = 12.77
  L2 = 1.28
  rmse = 0.08
  Pseudo-Rsq = 0.9746
p-val for shapiro-wilk test of normality of residuals: 0.6638
p-val for zero-mean noise: 0.95693
```

Note that we now use the complete data set including the 41 observations for which there were missing features. Also note that  $p$  has now increased from 46 to 51. The five new predictors are dummy variables which indicate missingness constructed from the predictors which exhibited missingness (due to the `use_missing_data_dummies_as_covars`

parameter being set to true). These variables are important if missingness *itself* shifts the response, as was the case in models explored in Chapter 8 of this document (Kapelner and Bleich, 2013). One way to test if this type of missingness is important is to run a covariate test (Section 7.4.7) on these new 5 covariates:

```
> cov_importance_test(bart_machine, covariates = c("M_normalized_losses",
  "M_bore", "M_stroke", "M_horsepower", "M_peak_rpm"))
BART test for importance of 5 covariates....p_val = 0.673
```

From the  $p$ -value, we cannot conclude that these variables have an effect on the response. This seems reasonable given that only 5 predictors featured missingness among only 40 observations. This suggests that it may be appropriate to shrink the model by leaving out these missing dummies, but still allow for missingness to be incorporated into the split rules:

```
> bart_machine = build_bart_machine(X, y, use_missing_data = TRUE)
```

The procedure of Section 7.3.2 also natively incorporates missing data during prediction. Missingness will yield larger credible intervals. In the example below, we suppose that the `curb_weight` and `symboling` values were unavailable for 20th automobile.

```
> x_star = X[20, ]
> calc_credible_intervals(bart_machine, x_star, ci_conf = 0.95)
      ci_lower_bd ci_upper_bd
[1,]      8.650093      8.824515
> x_star[c("curb_weight", "symboling")] = NA
> calc_credible_intervals(bart_machine, x_star, ci_conf = 0.95)
      ci_lower_bd ci_upper_bd
[1,]      8.622582      8.978313
```

### 7.4.10 Variable Selection

In this section we demonstrate the principled variable selection procedure introduced in Section 7.3.3. The following code will select variables based on the three thresholds and also displays the plot in Figure 7.10.<sup>5</sup>

```
> vs = var_selection_by_permute_response_three_methods(bart_machine,
               bottom_margin = 10, num_permute_samples = 10)
> vs$important_vars_local_names
"curb_weight" "city_mpg" "engine_size" "horsepower"
"length"      "width"    "num_cylinders" "body_style_convertible"
"wheel_base"  "peak_rpm" "highway_mpg"  "wheel_drive_fwd"
> vs$important_vars_global_max_names
"curb_weight" "city_mpg" "engine_size" "horsepower" "length"
> vs$important_vars_global_se_names
"curb_weight" "city_mpg" "engine_size" "horsepower" "length"
"width"       "num_cylinders" "wheel_base" "wheel_drive_fwd"
```

Usually, “Global Max” and “Global SE” perform similarly, as they are both more stringent in selection. However, in many situations it will not be clear to the data analyst which threshold is most appropriate. The “best” procedure can be chosen via cross-validation on out-of-sample RMSE as follows:

```
var_selection_by_permute_response_cv(bart_machine)
$best_method
[1] "important_vars_local_names"
```

---

<sup>5</sup>By default, variable selection is performed individually on dummy variables for a factor. The variable selection procedures return the permutation distribution and an aggregation of the dummy variables’ inclusion proportions can allow for variable selection to be performed on an entire factor.



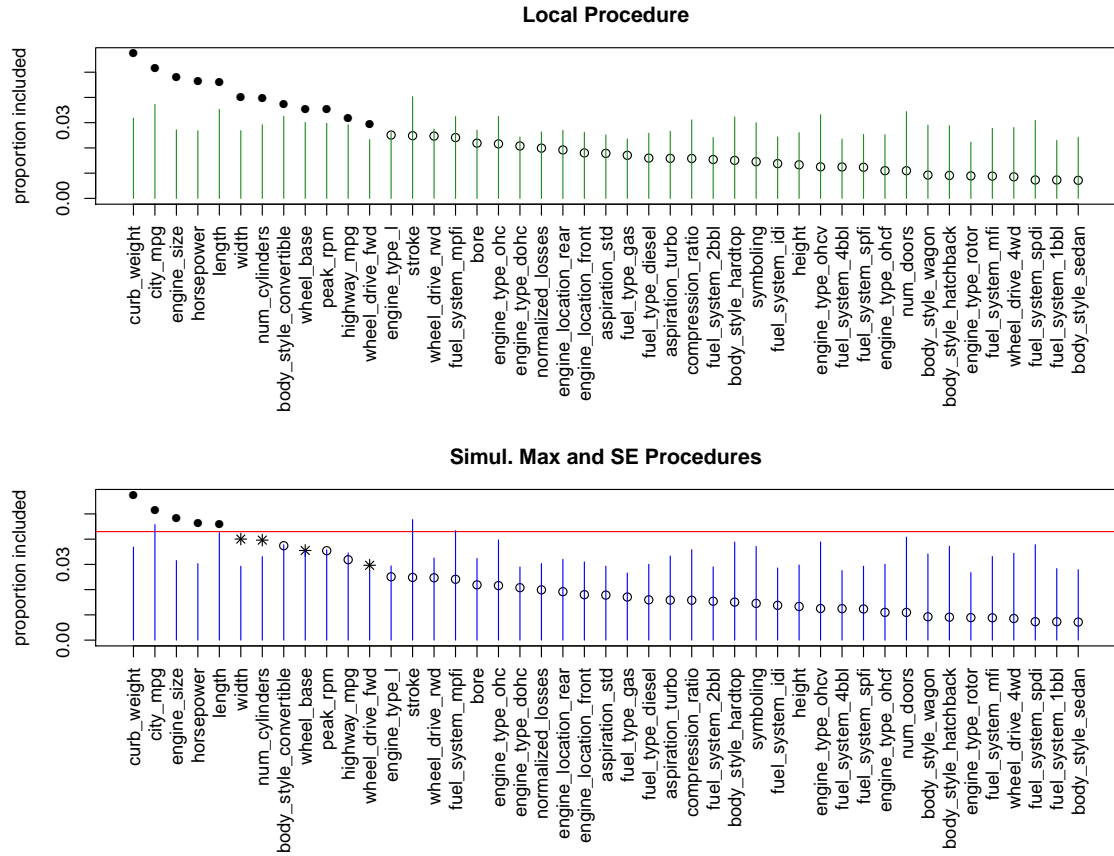


Figure 7.10: Visualization of the three variable selection procedures outlined in Section 7.3.3 with  $\alpha = 0.05$ . The top plot illustrates the “Local” procedure. The green lines are the threshold levels determined from the permutation distributions that must be exceeded for a variable to be selected. The plotted points are the variable inclusion proportions for the observed data (averaged over five duplicate `bartMachine` models). If the observed value is higher than the green bar, the variable is included and is displayed as a solid dot; if not, it is not included and it is displayed as an open dot. The bottom plot illustrates both the “Global SE” and “Global Max” thresholds. The red line is the cutoff for “Global Max” and variables pass this threshold are displayed as solid dots. The blue lines represent the thresholds for the “Global SE” procedure. Variables that exceed this cutoff but not the “Global Max” threshold are displayed as asterisks. Open dots exceed neither threshold.

```

$important_vars_cv

[1] "body_style_convertible" "city_mpg"           "curb_weight"
[4] "engine_size"           "engine_type_ohc"     "horsepower"
[7] "length"               "num_cylinders"       "peak_rpm"
[10] "wheel_base"           "wheel_drive_fwd"     "wheel_drive_rwd"
[13] "width"

```

On this dataset, the “best” approach (as defined by out-of-sample prediction error) is the “Local” procedure.

The following sections (7.4.11 and 7.4.12) demonstrate additional features using Friedman (1991)’s function:

$$\mathbf{y} = 10\sin(\pi\mathbf{x}_1\mathbf{x}_2) + 20(\mathbf{x}_3 - .5)^2 + 10\mathbf{x}_4 + 5\mathbf{x}_5 + \boldsymbol{\mathcal{E}}, \quad \boldsymbol{\mathcal{E}} \sim \mathcal{N}_n(\mathbf{0}, \sigma^2\mathbf{I}). \quad (7.6)$$

### 7.4.11 Informed Prior Information on Covariates

Chapter 9 of this document (Bleich et al., 2013) proposes a method for incorporating informed prior information about the predictors into BART. This can be achieved by modifying the prior on the splitting rules as well as the corresponding calculations in the Metropolis-Hastings step. In particular, covariates believed to influence the response can be proposed more often as candidates for splitting rules. Useful prior information can aid in both variable selection and prediction tasks. We illustrate the impact of a correctly informed prior in the context of the Friedman function (Equation 9.7). We include the 5 predictors which influence the response as well as 95 that do not.

All that is required is a specification of relative weights for each predictor. These are converted internally to probabilities. We assign 5 times the weight to the 5 true

covariates of the model relative to the 95 useless covariates.

```
> prior = c(rep(5, times = 5), rep(1, times = 95))
```

We now sample 500 observations from the Friedman function and construct a default `bartMachine` model as well as a `bartMachine` model with the informed prior and compare their performance on a test set of another 500 observations.

```
> bart_machine = build_bart_machine(X, y)
> bart_machine_informed = build_bart_machine(X, y, cov_prior_vec = prior)

> bart_predict_for_test_data(bart_machine, Xtest, ytest)$rmse
[1] 1.661159
> bart_predict_for_test_data(bart_machine_informed, Xtest, ytest)$rmse
[1] 1.232925
```

There is a substantial improvement in out-of-sample predictive performance when a properly informed prior is used.

Note that we recommend use of the prior vector to down-weight the indicator variables that result from dummifying factors so that the total set of dummy variables has the same weight as a continuous covariate.

### 7.4.12 Interaction Effect Detection

In Section 7.4.6, we explored using variable inclusion proportions to understand the relative influences of different covariates. A similar procedure can be carried out for examining interaction effects within a BART model. This question was initially explored in Damien et al. (2013) where an interaction was considered to exist between two variables if they both appeared in at least one splitting rule in a given tree. We refine the definition of an interaction as follows.

We first begin with a  $p \times p$  matrix of zeroes. Within a given tree, for each split rule variable  $j$ , we look at all split rule variables of child nodes,  $k$ , and we increment the  $j, k$  element of the matrix. Hence variables are considered to interact in a given tree *only if* they appear together in a contiguous downward path from the root node to a terminal node. Note that a variable may interact with itself (when fitting a linear effect, for instance). Since there is no order between the parent and child, we then add the  $j, k$  counts together with the  $k, j$  counts (if  $j \neq k$ ). Summing across trees and Gibbs samples gives the total number of interactions for each pair of variables from which relative importance can be assessed.

We demonstrate interaction detection on the Friedman function using 10 covariates using the code below:

```
> interaction_investigator(bart_machine, num_replicates_for_avg = 25,
  num_var_plot = 10, bottom_margin = 5)
```

Shown in Figure 7.11 are the ten most important interactions in the model. The illustration is averaged over many model constructions to obtain stable estimates across many posterior modes in the sum-of-trees distribution. Notice that the interaction between  $\mathbf{x}_1$  and  $\mathbf{x}_2$  dominates all other terms, as BART is correctly capturing the single true interaction effect in Equation 9.7. Choosing which of these interactions *significantly* affect the response is not addressed in this paper. The methods suggested in Section 7.3.3 may be applicable here and we consider this to be fruitful future work.

## 7.5 Classification Features

In this section we highlight the features that differ from the regression case when the response is dichotomous. The illustrative dataset consists of 332 Pima Indians

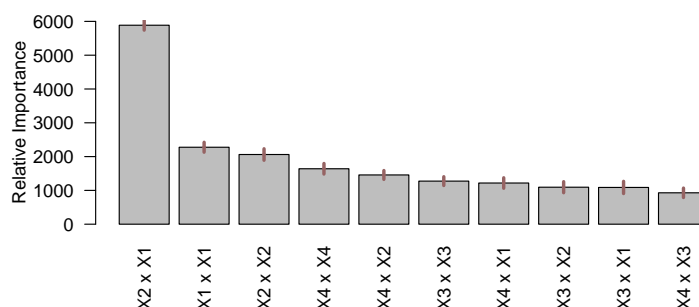


Figure 7.11: The top 10 average variable interaction counts (termed “relative importance”) in the default `bartMachine` model for the Friedman function data averaged over 25 model constructions. The segments atop the bars represent 95% confidence intervals.

obtained from the UCI repository. Of the 332 subjects, 109 were diagnosed with diabetes, the binary response variable. There are seven continuous predictors which are body metrics such as blood pressure, glucose concentration, etc. and there is no missing data.

Building a `bartMachine` model for classification has the same computing parameters except that  $q, \nu$  cannot be specified since there is no longer a prior on  $\sigma^2$  (see Section 7.2.3). We first build a cross-validated model below.

```
> bart_machine_cv = build_bart_machine_cv(X, y)
... BART CV win: k: 3 m: 50
> bart_machine_cv

Bart Machine v1.0b for classification

training data n = 332 and p = 7
built in 0.5 secs on 4 cores, 50 trees, 250 burn-in and 1000 post. samples

confusion matrix:
```

	predicted No	predicted Yes	model errors
actual No	211.000	12.00	0.054
actual Yes	41.000	68.00	0.376
use errors	0.163	0.15	0.160

Classification models have an added hyperparameter, `prob_rule_class`, which is the rule for determining if the probability estimate is great enough to be classified into the positive category. We can see above that the `bartMachine` at times predicts “NO” for true “YES” outcomes and we suffer from a 37.6% error rate for this outcome. We can try to mitigate this error by lowering the threshold to increase the number of “YES” labels predicted:

```
> build_bart_machine(X, y, prob_rule_class = 0.3)
```

```
Bart Machine v1.0b for classification
```

```
training data n = 332 and p = 7
```

```
built in 0.5 secs on 4 cores, 50 trees, 250 burn-in and 1000 post. samples
```

```
confusion matrix:
```

	predicted No	predicted Yes	model errors
actual No	178.000	45.000	0.202
actual Yes	12.000	97.000	0.110
use errors	0.063	0.317	0.172

This lowers the model error to 11% for the “YES” class, but at the expense of increasing the error rate for the “NO” class. We encourage the user to cross-validate this rule based on the appropriate objective function for the problem at hand.

We can also check out-of-sample statistics:

```
> oos_stats = k_fold_cv(X, y, k_folds = 10)
> oos_stats$confusion_matrix
```

	predicted No	predicted Yes	model errors
actual No	203.000	20.000	0.090
actual Yes	47.000	62.000	0.431
use errors	0.188	0.244	0.202

Note that it is possible to predict both class labels and probability estimates for given observations:

```
> predict(bart_machine_cv, X[1 : 2, ], type = "prob")
[1] 0.6253160 0.1055975
> predict(bart_machine_cv, X[1 : 2, ], type = "class")
[1] "Yes" "No"
```

When using the covariate tests of Section 7.4.7, total misclassification error becomes the statistic of interest instead of Pseudo- $R^2$ . The  $p$  value is calculated now as the proportion of null samples with *higher* misclassification error. Figure 7.12 illustrates the test showing that predictor **age** seems to matter in the prediction of **Diabetes**, controlling for other predictors.

The partial dependence plots of Section 7.4.8 are now scaled as probit of the probability estimate. Figure 7.13 illustrates that as glucose increases, the probability of contracting **Diabetes** increases linearly on a probit scale.

Credible intervals are implemented for classification **bartMachine** and are displayed on the probit scale. Note that the prediction intervals of Section 7.4.5 do not exist for classification.

```
> calc_credible_intervals(bart_machine_cv, X[1 : 2, ])
```

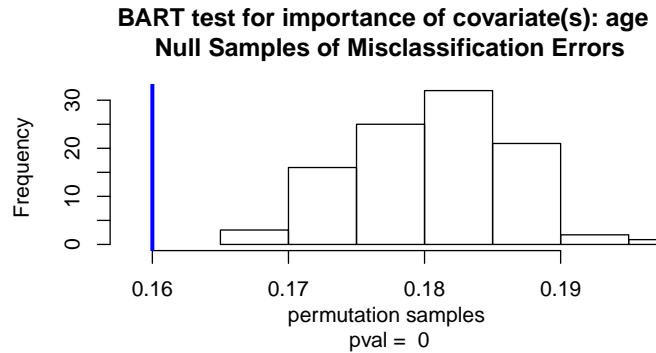


Figure 7.12: Test of covariate importance for predictor **age** on whether or not the subject will contract Diabetes.

```

      ci_lower_bd ci_upper_bd
[1,]  0.34865355  0.8406097
[2,]  0.01686486  0.2673171

```

Other functions work similarly to regression except those that plot the responses and those that explicitly depend on RMSE as an error metric.

## 7.6 Discussion

This article introduced **bartMachine**, a new R package which implements Bayesian Additive Regression Trees. The goal of this package is to provide a fast, extensive and user-friendly implementation accessible to a wide range of data analysts, and increase the visibility of **BART** to a broader statistical audience. We hope we have provided organized, well-documented open-source code and we encourage the community to make innovations on this package.



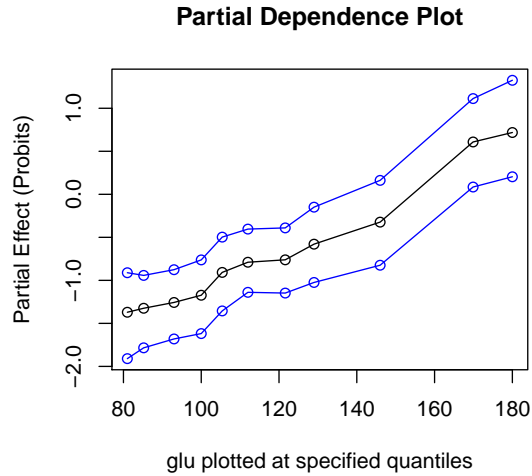


Figure 7.13: PDP for predictor `glu`. The blue lines are 95% credible intervals.

## Replication

The code for `bartMachine` is located at <http://github.com/kapelner/bartMachine> under the MIT license. Results, tables, and figures found in this paper can be replicated via the scripts located in the `bart_package_paper` folder within this `git` repository.

## Acknowledgements

We thank Richard Berk, Andreas Buja, Zachary Cohen, Ed George, Alex Goldstein, Shane Jensen, Abba Krieger, and Robert DeRubeis for helpful discussions. We thank Simon Urbanek for his generous help with `rJava`. Adam Kapelner acknowledges support from the National Science Foundation's Graduate Research Fellowship Program.

## Incorporating Missingness into BART

By Adam Kapelner and Justin Bleich

### Abstract

We present a method for incorporating missing data into general forecasting problems which use non-parametric statistical learning. We focus on a tree-based method, Bayesian Additive Regression Trees (BART), enhanced with “Missingness Incorporated in Attributes,” an approach recently proposed for incorporating missingness into decision trees. This procedure extends the native partitioning mechanisms found in tree-based models and does not require imputation. Simulations on generated models and real data indicate that our procedure offers promise for both selection model and pattern mixture frameworks as measured by out-of-sample predictive accuracy. We also illustrate BART’s abilities to incorporate missingness into uncertainty intervals. Our implementation is readily available in the R package `bartMachine`.

### 8.1 Introduction

This article addresses prediction problems where covariate information is missing during model construction and is also missing in future observations for which we

are obligated to generate a forecast. Our aim is to innovate a non-parametric statistical learning extension which incorporates missingness into *both* the training and the forecasting phases. In the spirit of non-parametric learning, we wish to incorporate the missingness in both phases automatically, without the need for pre-specified modeling.

We limit our focus to tree-based statistical learning, which has demonstrated strong predictive performance and has consequently received considerable attention in recent years. State-of-the-art algorithms include Random Forests (RF, Breiman, 2001a), stochastic gradient boosting (Friedman, 2002), and Bayesian Additive and Regression Trees (BART, Chipman et al., 2010), the algorithm of interest in this study. Popular implementations of these methods do not incorporate covariate missingness *natively* without relying on either imputation or a complete case analysis of observations with no missing information.

Previous simulations and real data set applications have indicated that BART is capable of achieving excellent predictive accuracy. Unlike most competing techniques, BART is composed of a probability model, rather than a procedure that is purely “algorithmic” (Breiman, 2001b). BART presents an alternative approach to statistical learning for those comfortable with the Bayesian framework. This framework provides certain advantages, such as built-in estimates of uncertainty in the form of credible intervals as well as the ability to incorporate prior information on covariates (Bleich et al., 2013). However, no means for incorporating missing data in BART has been published to date. Our goal here is to develop a principled way of adapting BART’s machinery to incorporate missing data that takes advantage of the Bayesian framework.

Our proposed method, henceforth named **BARTm**, modifies the recursive partitioning scheme during construction of the decision trees to incorporate missing data into splitting rules. By relying on the Metropolis-Hastings algorithm embedded in BART,

our method attempts to send missing data to whichever of the two daughter nodes increases overall model likelihood. Missingness *itself* also becomes a valid splitting criterion.

During model construction, taking advantage of this modified set of splitting rules does not require imputation, which relies on assumptions that cannot be easily verified. Our approach is equally viable for continuous and nominal covariate data and both selection and pattern mixture models. The latter models do not assume that missing data is necessarily free of information; the data may have gone missing for a reason crucial to the response function and therefore crucial to our forecast. **BARTm** is able to exploit this relationship when appropriate.

Since missingness is handled natively within the algorithm, **BARTm** can generate predictions on future data with missing entries as well. Additionally, **BART**'s Bayesian framework also naturally provides estimates of uncertainty in the form of credible intervals. The amount of uncertainty increases with the amount of information lost due to missingness; thereby missingness is appropriately incorporated into the standard error of the prediction. Also, our proposed procedure has negligible impact on the runtime during both model construction and prediction phases.

In Sections 8.2.1 - 8.2.3, we provide a framework for statistical learning with missingness with a focus on decision trees. We give a brief overview of the **BART** algorithm in Section 8.2.4 and explain the internals of **BARTm** in Section 8.3. We then demonstrate **BARTm**'s predictive performance on generated models in Section 8.4 as well as real data with a variety of missing data scenarios in Section 8.5. We conclude with Section 13.5. **BARTm** can be found in the R package **bartMachine** which is available on CRAN (Kapelner and Bleich, 2013).

## 8.2 Background

### 8.2.1 A Framework for Missing Data in Statistical Learning

Consider  $p$  covariates  $\mathbf{X} := [X_1, \dots, X_p]$ , a continuous response  $\mathbf{Y}$  and an unknown function  $f$  where  $\mathbf{Y} = f(\mathbf{X}) + \boldsymbol{\varepsilon}$ . When  $\mathbf{Y}$  is binary, we use a similar framework with an appropriate link function encapsulating  $f$ . We denote  $\boldsymbol{\varepsilon}$  as the noise in the response unexplained by  $f$ . The goal of statistical learning is to use the *training set*,  $[\mathbf{y}_{\text{train}}, \mathbf{X}_{\text{train}}]$  which consists of  $n$  observations drawn from the population  $\mathbb{P}(\mathbf{Y}, \mathbf{X})$ , to produce an estimate,  $\hat{f}$ , the best guess of  $\mathbb{E}[\mathbf{Y} \mid \mathbf{X}]$ . This function estimate can then be used to generate predictions on future test observations with an unknown response. We denote these future observations as  $\mathbf{X}_*$  which we assume are likewise drawn from the same population as the training set.

Missingness is one of the scourges of data analysis, plaguing statistical learning by causing missing entries in both the training matrix  $\mathbf{X}_{\text{train}}$  as well as missing entries in the future records,  $\mathbf{X}_*$ . In the statistical learning context, the training set is defined by observations which do not exhibit missingness in their response,  $\mathbf{y}_{\text{train}}$ . Records with missing responses cannot be used to build models for estimation of  $f$ . “Imputing missing values in the response” for the new  $\mathbf{X}_*$  is equivalent to “prediction” and is the primary goal of statistical learning. Thus, “missingness” considered in this paper is missingness *only* in  $\mathbf{X}_{\text{train}}$  and  $\mathbf{X}_*$ . We denote missingness in the  $p_M \leq p$  features of  $\mathbf{X}$  which suffer from missingness as  $\mathbf{M} := [M_1, \dots, M_{p_M}]$ , binary vectors where 1 indicates missing and 0 indicates present, and covariates that are present with  $\mathbf{X}_{\text{obs}} := [X_{\text{obs}_1}, \dots, X_{\text{obs}_p}]$ . The main goal of statistical learning with missingness is to estimate  $\mathbb{E}[\mathbf{Y} \mid \mathbf{X}_{\text{obs}}, \mathbf{M}]$ .

We now frame missing data models in statistical learning using the canonical framework of selection and pattern-mixture models (Little, 1993). Conditional on  $\mathbf{X}$ , *selection models* factor the full data likelihood as

$$\mathbb{P}(\mathbf{Y}, \mathbf{M} \mid \mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma}) = \mathbb{P}(\mathbf{Y} \mid \mathbf{X}, \boldsymbol{\theta}) \mathbb{P}(\mathbf{M} \mid \mathbf{X}, \boldsymbol{\gamma}) \quad (8.1)$$

where  $\boldsymbol{\theta}$  and  $\boldsymbol{\gamma}$  are parameter vectors and are assumed distinct. The first term on the right hand side reflects that the marginal likelihood for the response  $\mathbb{P}(\mathbf{Y} \mid \mathbf{X}, \boldsymbol{\theta})$  is independent of missingness. The second term on the right conventionally conditions on  $\mathbf{Y}$ . In the forecasting paradigm, missingness is *assumed independent* of the response because  $\mathbf{Y}$  is often yet to be realized and thus its unknown value should not influence  $\mathbf{M}$ , the missingness of the previously realized covariates.

Conditional on  $\mathbf{X}$ , *pattern mixture models* partition the full data likelihood as

$$\mathbb{P}(\mathbf{Y}, \mathbf{M} \mid \mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\gamma}) = \mathbb{P}(\mathbf{Y} \mid \mathbf{M}, \mathbf{X}, \boldsymbol{\theta}) \mathbb{P}(\mathbf{M} \mid \mathbf{X}, \boldsymbol{\gamma}). \quad (8.2)$$

where  $\boldsymbol{\theta}$  and  $\boldsymbol{\gamma}$  are parameter vectors and again assumed distinct. The difference between the above and Equation 8.1 is the marginal likelihood of the response is now a function of  $\mathbf{M}$ . This means there can be different response models under different patterns of missingness in the  $p_M$  covariates.

In both selection and pattern-mixture paradigms, the term on the right is the *missing data mechanism* (MDM), which traditionally is the mechanism controlling missingness in the response. In our framework however, the MDM controls missingness only in  $\mathbf{X}$ : the covariates (and parameters  $\boldsymbol{\gamma}$ ) create missingness within themselves which inevitably needs to be incorporated during model construction and forecasting. Thus, the MDM is conceptually equivalent in both the selection and pattern mixture paradigms.

The conceptual difference between the selection and pattern mixture models in the statistical learning framework can be envisioned as follows. Imagine the full

covariates  $\mathbf{X}$  are realized but due to the MDM,  $\mathbf{X}$  is latent and we instead observe  $\mathbf{X}_{\text{obs}}$  and  $\mathbf{M}$ . In the selection paradigm,  $\mathbf{Y}$  is realized only from the full covariates via  $\mathbb{P}(\mathbf{Y} \mid \mathbf{X}, \boldsymbol{\theta})$ . However, in the pattern-mixture paradigm, both  $\mathbf{X}$  and  $\mathbf{M}$  intermix to create many collated response models  $\{\mathbb{P}(\mathbf{Y} \mid \mathbf{X}, \boldsymbol{\theta}, \mathbf{M} = m)\}_{m \in \mathcal{M}}$  corresponding to different points in  $\mathbf{M}$ -space. Thus, under our assumptions, selection models are a subset of pattern mixture models. Note that pattern-mixture models are chronically under-identified and difficult to explicitly model in practice. We address why our proposed method is well-suited to handle prediction problems under this framework in Section 8.3.

We now present Little and Rubin (2002)’s taxonomy of MDM’s which are traditionally framed in the selection model paradigm but here apply to both paradigms: (1) missing completely at random (MCAR), (2) missing at random (MAR) and (3) not missing at random (NMAR). MCAR is a mechanism that generates missingness in covariate  $j$  without regard to the value of  $X_j$  itself nor the values and missingness of any other covariates, denoted  $\mathbf{X}_{-j}$ . MCAR is exclusively determined by exogenous parameter(s)  $\boldsymbol{\gamma}$ . The MAR mechanism generates missingness without regard to  $X_j$ , its own value, but can depend on values of other attributes  $\mathbf{X}_{-j}$  as well as  $\boldsymbol{\gamma}$ . The NMAR mechanism features the additional dependence on the value of  $X_j$  itself as well as unobserved covariates (note that explicit dependence on unobserved covariates was not explored as MDM’s in this paper). We summarize these mechanisms in Table 8.1. In our framework, each of the  $p_M \leq p$  covariates with missingness, denoted as  $X_j$ ’s, are assumed to have their own MDM $_j$ . Thus, the full MDM for the whole covariate space,  $\mathbb{P}(\mathbf{M} \mid \mathbf{X}, \boldsymbol{\gamma})$ , can be arbitrarily convoluted, exhibiting combinations of MCAR, MAR and NMAR among its  $p_M$  covariates and each MDM $_j$  relationship may be highly non-linear with complicated interactions.

We conclude this section by emphasizing that in the non-parametric statistical learning framework where predictive performance is the objective, there is no need

MDM	$\mathbb{P}(\mathbf{M}_j \mid X_{j,\text{miss}}, \mathbf{X}_{-j,\text{miss}}, \mathbf{X}_{-j,\text{obs}}, \boldsymbol{\gamma}) = \dots$
MCAR	$\mathbb{P}(\mathbf{M}_j \mid \boldsymbol{\gamma})$
MAR	$\mathbb{P}(\mathbf{M}_j \mid \mathbf{X}_{-j,\text{miss}}, \mathbf{X}_{-j,\text{obs}}, \boldsymbol{\gamma})$
NMAR	(does not simplify)

Table 8.1: MDM models in the context of statistical learning.  $\mathbf{M}_j$  is an indicator vector which takes the value one when the  $j^{\text{th}}$  covariate is missing for the  $i$ th observation.  $\mathbf{X}_{-j,\text{obs}}$  are the observed values of the other covariates, besides  $j$ .  $\mathbf{X}_{-j,\text{miss}}$  are the values of the other covariates, besides  $j$ , which are not observed because they are missing.

for explicit inference of  $\boldsymbol{\theta}$  (which may have unknown structure and arbitrary, possibly infinite, dimension). Instead, the algorithm performs “black-box” estimation of the data generating process such that the output  $\hat{f}$  estimates the  $\mathbb{E}[\mathbf{Y} \mid \mathbf{X}_{\text{obs}}, \mathbf{M}]$  function. Thus, if we can successfully estimate this conditional expectation function directly, then accurate forecasts can be obtained. This is the approach that BARTm takes.

### 8.2.2 Strategies for Incorporating Missing Data

A simple strategy for incorporating missingness into model building is to simply ignore the observations in  $\mathbf{X}_{\text{train}}$  that contain at least one missing measurement. This is called “list-wise deletion” or “complete case analysis.” It is well known that complete case analysis will be unbiased for MCAR and MAR selection models where missingness does not depend on the response when the target of estimation is  $\mathbb{E}[\mathbf{Y} \mid \mathbf{X}]$ . However, when forecasting, the data analyst must additionally be guaranteed that  $\mathbf{X}_*$  has no missing observations, since it is not possible to generate forecasts for these cases.

By properly modeling missingness, incomplete cases can be used and more information about  $\mathbb{E}[\mathbf{Y} \mid \mathbf{X}]$  becomes available, potentially yielding higher predictive



performance. One popular strategy is to guess or “impute” the missing entries. These guesses are then used to “fill in” the holes in  $\mathbf{X}_{\text{train}}$  and  $\mathbf{X}_*$ . The imputed  $\mathbf{X}_{\text{train}}$  is then used *as if* it were the real covariate data when constructing  $\hat{f}$  and the imputed  $\mathbf{X}_*$  is then used as if it were the real covariate data during forecasting. To carry out imputation, the recommended strategy is to properly model the predictive distribution and then draws from the model are used to fill in the missing entries. *Multiple imputation* involves imputing many times and averaging over the results from each imputation (Rubin, 1978). In statistical learning, a prediction could be calculated by averaging the predictions from many  $\hat{f}$ ’s built from many imputed  $\mathbf{X}_{\text{train}}$ ’s and then further averaging over many imputed  $\mathbf{X}_*$ ’s. In practice, having knowledge of both the missing data mechanism and each probability model is very difficult and has usually given way to nonparametric methods such as  $k$ -nearest neighbors (Troyanskaya et al., 2001) for continuous covariates and saturated multinomial modeling (Schafer, 1997) for categorical covariates. The widely used R package `randomForest` (Liaw and Wiener, 2002) imputes via “hot-decking” (Little and Rubin, 2002).

A more recent approach, `MissForest` (Stekhoven and Bühlmann, 2012), fits non-parametric imputation models for any combination of continuous and categorical input data, even when the response is unobserved. In this unsupervised procedure (i.e., no response variable needed), initial guesses for the imputed values are made. Then, for each attribute with missingness, the observed values of that attribute are treated as the response and a RF model is fit using the remaining attributes as predictors. Predictions for the missing values are made via the trained RF and serve as updated imputations. The process proceeds iteratively through each attribute with missingness and then repeats until a stopping criterion is achieved. The authors argue that their procedure intrinsically constitutes multiple imputation due to Random Forest’s averaging over many unpruned decision trees. The authors also state that their method will perform particularly well when “the data include complex

interactions or non-linear relations between variables of unequal scales and different type.” Although no explicit reference is given to Little and Rubin (2002)’s taxonomy in their work, we expect **MissForest** to perform well in situations generally well-suited for imputation, namely, the MCAR and MAR selection models discussed in Section 8.2.1. **MissForest** would not be suited for NMAR MDMs as imputation values for  $X_j$  can only be modeled from  $\mathbf{X}_{-j}$  in their implementation. Additionally, implementing **MissForest** would not be recommended for pattern-mixture scenarios because imputation is insufficient to capture differing response patterns.

Since **BART** is composed primarily of a sum-of-regression-trees model, we now review strategies for incorporating missing data in tree-based models.

### 8.2.3 Missing data in Binary Decision Trees

*Binary decision trees* are composed of a set of connecting nodes. *Internal nodes* contain a *splitting rule*, for example,  $\mathbf{x}_j < \mathbf{c}$ , where  $\mathbf{x}_j$  is the *splitting attribute* and  $\mathbf{c}$  is the *splitting value*. An observation that satisfies the rule is passed to the left daughter node otherwise it is passed to the right daughter node. This partitioning proceeds until an observation reaches a *terminal node*. Terminal nodes (also known as *leaves*) do not have splitting rules and instead have *leaf values*. When an observation “lands” in a terminal node it is assigned the leaf value of the terminal node in which it has landed. In *regression trees*, this leaf value is a real number, and is the estimate of the response  $y$  for the given covariates. Thus, regression trees are a nonparametric fitting procedure where the estimate of  $f$  is a partition of predictor space into various hyperrectangles. Regression trees are well-known for their ability to approximate complicated response surfaces containing both nonlinearities and interaction effects.

There are many different ways to build decision trees. Many classic approaches rely on a greedy procedure to choose the best splitting rule at each node based on some pre-determined criterion. Once the construction of the tree is completed, the

tree is then pruned back to prevent overfitting.

Previous efforts to handle missingness in trees include surrogate variable splitting (Therneau and Atkinson, 1997), “Missing Incorporated in Attributes” (MIA, Twala et al., 2008, section 2) and many others (see Ding and Simonoff, 2010 and Twala, 2009). MIA, the particular focus for this work, is a procedure that natively uses missingness when greedily constructing the rules for the decision tree’s internal nodes. We summarize the procedure in Algorithm 2 and we explain how the expanded set of rules is injected into the **BART** procedure in Section 8.3.

---

**Algorithm 2** *Splitting rule choices during construction of a new tree branch in MIA.*

---

The algorithm chooses one of the following three rules for all splitting attributes and all splitting values  $c$ . Since there are  $p$  splitting attributes and at most  $n - 1$  unique values to split on, the greedy splitting algorithm with MIA checks  $2(n - 1)p + p$  possible splitting rules at each iteration instead of the classic  $(n - 1)p$ .

---

- 1: If  $x_{ij}$  is present and  $x_{ij} \leq c$ , send this observation left ( $\leftarrow$ ); otherwise, send this observation right ( $\rightarrow$ ). If  $x_{ij}$  is missing, send this observation left ( $\leftarrow$ ).
  - 2: If  $x_{ij}$  is present and  $x_{ij} \leq c$ , send this observation left ( $\leftarrow$ ); otherwise, send this observation right ( $\rightarrow$ ). If  $x_{ij}$  is missing, send this observation right ( $\rightarrow$ ).
  - 3: If  $x_{ij}$  is missing, send this observation left ( $\leftarrow$ ); if it is present, regardless of its value, send this observation right ( $\rightarrow$ ).
- 

There are many advantages of the MIA approach. First, MIA has the ability to model complex MAR and NMAR relationships, as evidenced in both Twala et al. (2008) and the results of Sections 8.4 and 8.5. Since missingness is integrated into the splitting rules, forecasts can be made without imputing when  $\mathbf{X}_*$  contains missingness.

Another strong advantage of MIA is the ability to split on feature missingness (line 3 of Algorithm 2). This splitting rule choice allows for the tree to better capture pattern mixture models where missingness directly influences the response model.

Generally speaking, imputation ignores pattern mixture models; missingness is only viewed as holes to be filled-in and forgotten.

Due to these benefits as well as conceptual simplicity, we chose to implement MIA-within-BART, denoted “BARTm”, when enhancing BART to handle missing data.

## 8.2.4 BART

Bayesian Additive Regression Trees is a combination of many regression trees estimated via a Bayesian model. Imagine the true response function can be approximated by the sum of  $m$  trees with additive normal and homoskedastic noise:

$$\mathbf{Y} = f(\mathbf{X}) + \boldsymbol{\varepsilon} \approx \mathfrak{T}_1^{\text{leaf}}(\mathbf{X}) + \mathfrak{T}_2^{\text{leaf}}(\mathbf{X}) + \dots + \mathfrak{T}_m^{\text{leaf}}(\mathbf{X}) + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}_n(\mathbf{0}, \sigma^2 \mathbf{I}_n). \quad (8.3)$$

The notation,  $\mathfrak{T}^{\text{leaf}}$ , denotes both structure and splitting rules ( $\mathfrak{T}$ ) as well as leaf values ( $\text{leaf}$ ). Note that BART can be adapted for classification problems by using a probit link function and a data augmentation approach relying on latent variables (Albert and Chib, 1993).

BART can be distinguished from other purely algorithmic ensemble-of-trees models by its full Bayesian model, consisting of both a set of independent priors and likelihoods. Its posterior distribution is estimated via Gibbs Sampling (Geman and Geman, 1984) with a Metropolis-Hastings step (Hastings, 1970).

There are three regularizing priors within the BART model which are designed to prevent overfitting. The first prior, placed on the tree structure is designed to prevent trees from growing too deep, thereby limiting the complexity that can be captured by a single tree. The second prior is placed on the leaf value parameters (the predicted values found in the terminal nodes) and is designed to shrink the leaf values towards the overall center of the response’s empirical distribution. The third prior is placed

on the variance of the noise  $\sigma^2$  and is designed to curtail overfitting by introducing noise into the model if it begins to fit too snugly. Our development of **BARTm** uses the default hyperparameters recommended in the original work (Chipman et al., 2010). For those who do not favor a pure Bayesian approach, these priors can be thought of as tuning parameters.

In addition to the regularization priors, **BART** imposes an agnostic prior on the splitting rules within the decision tree branches. First, for a given branch, the splitting attribute is uniformly drawn from the set  $\{x_1, \dots, x_p\}$  of variables available at the branch. The splitting value is then selected by drawing uniformly from the available values conditional on the splitting attribute  $j$ . Selecting attributes and values from a uniform discrete distribution represents a digression from the approach used in decision tree algorithms of greedily choosing splits based on some splitting criteria. Extending this prior allows for **BART** to incorporate MIA, which is discussed in Section 8.3.

To generate draws from the posterior distribution, each tree is fit iteratively, holding the other  $m - 1$  trees constant, by using only the portion of the response left unfitted. To sample trees, changes to the tree structure are proposed then accepted or rejected via a Metropolis-Hastings step. The tree proposals are equally-likely alterations: growing a leaf by adding two daughter nodes, pruning two twin leaves (rendering their parent node into a leaf), or changing a splitting rule. Following the tree sampling, the posterior for the leaf value parameters are Gibbs sampled. The likelihood of the predictions in each node is assumed to be normal. Therefore, the normal-normal conjugacy yields the canonical posterior normal distribution. After sampling all tree changes and terminal node parameters, the variance parameter  $\sigma^2$  is Gibbs sampled. By model assumption, the likelihood for the errors is normal and the conjugacy with the inverse-gamma prior yields the canonical posterior inverse-gamma.

Usually around 1,000 Metropolis-within-Gibbs iterations are run as “burn-in” un-

til  $\sigma^2$  converges (by visual inspection). Another 1,000 or so are sampled to obtain “burned-in” draws from the posterior, which define the **BART** model. Forecasts are then obtained by dropping the observations of  $\mathbf{X}_*$  down the collection of sampled trees within each burned-in Gibbs sample. A point prediction  $\hat{y}$  is generated by summing the posterior leaf values across the trees as in Equation A.12. *Credible intervals*, which are intervals containing a desired percentage (e.g. 95%) of the posterior probability mass for a Bayesian parameter of interest, can be computed via the desired empirical quantiles over the burned-in samples.

For a thorough description about the internals of **BART** see Chipman et al. (2010) and Kapelner and Bleich (2013).

### 8.3 Missing Incorporated in Attributes within BART

Implementing **BARTm** is straightforward. We previously described the prior on the splitting rules within the decision tree branches as being discrete uniform on the possible splitting attributes and discrete uniform on the possible splitting values. To account for Lines 1 and 2 in the MIA procedure (Algorithm 2), the splitting attribute  $x_j$  and split value are proposed as explained in Section 8.2.4, but now we additionally propose a direction (left or right with equal probability) for records to be sent when the records have with missing values in  $x_j$ . A possible splitting rule would therefore be “ $x_{ij} < c$  and move left if  $x_{ij}$  is missing.” To account for Line 3 in the algorithm, splitting on missingness itself, we create dummy vectors of length  $n$  for each of the  $p_M$  attributes with missingness, denoted  $\mathbf{M}_1, \dots, \mathbf{M}_{p_M}$ , which assume the value 1 when the entry is missing and 0 when the entry is present. We then augment the original training matrix together with these dummies and use the augmented training matrix,  $\mathbf{X}'_{\text{train}} := [\mathbf{X}_{\text{train}}, M_1, \dots, M_{p_M}]$ , as the training data in the **BARTm** algorithm. Once again, the prior on splitting rules is the same as the original **BART** but now with

the additional consideration that the direction of missingness is equally likely left or right conditional on the splitting attribute and value. But why should this algorithm yield good predictive performance under the framework discussed in Section 8.2.1?

We expect **BARTm** to exhibit greater predictive performance over MIA in classical decision trees for two reasons. First, **BARTm**’s sum-of-trees model offers much greater fitting flexibility of interactions and non-linearities compared to a single tree; this flexibility will explore models that the data analyst may not have thought of. Additionally, due to the greedy nature of decision trees, once a split is chosen, the direction in which missingness is sent cannot be reversed. **BARTm** can alter its trees by pruning and regrowing nodes or changing splitting rules. These proposed modifications to the trees are accepted or rejected stochastically using the Metropolis-Hastings machinery depending on how strongly the proposed move increases the model’s likelihood.

We hypothesize that **BARTm**’s stochastic search for splitting rules allows observations with missingness to be grouped with observations having similar response values. Due to the Metropolis-Hastings step, only splitting rules and corresponding groupings that increase overall model likelihood  $\mathbb{P}(\mathbf{Y} \mid \mathbf{X}, \mathbf{M})$  will be accepted. In essence, **BARTm** is “feeling around” predictor space for a location where the missing data would most increase the overall marginal likelihood. For selection models, since splitting rules can depend on any covariate including the covariate with missing data, it should be possible to generate successful groupings for the missing data under both MAR and NMAR mechanisms.

As a simple MAR example, suppose there are two covariates  $X_1$  and  $X_2$  and a MAR mechanism where  $X_2$  is increasingly likely to go missing for large values of  $X_1$ . **BARTm** can partition this data in two steps to increase overall likelihood: (1) A split on a large value of  $X_1$  and then (2) a split on  $M_2$ . As a simple NMAR example, suppose a mechanism where  $X_2$  is more likely to be missing for large values of  $X_2$ . **BARTm** can select splits of the form “ $x_2 > c$  and  $x_2$  is missing” with  $c$  large. Here,

the missing data is appropriately kept with larger values of  $X_2$  and overall likelihood should be increased.

When missingness does not depend on any other covariates, it should be more difficult to find appropriate ways to partition the missing data, and we hypothesize that **BARTm** will be least effective for selection models with MCAR MDMs. We believe this is due to the regularization prior on the depths of the trees coupled with the fact that all missing data must move to the same daughter node. Thus, the trees do not grow deep enough to create sufficiently complex partitioning schemes to handle the MCAR mechanism. Additionally, we hypothesize that **BARTm** has potential to perform well on pattern mixture models due to the partitioning nature of the regression tree. **BARTm** can partition the data based on different patterns of missingness by creating splits based on missingness itself. Then, underneath these splits, different submodels for the different patterns can be constructed. If missingness is related to the response, there is a good chance **BARTm** will find it and exploit it, yielding accurate forecasts.

Another motivation for adapting MIA to **BART** arises from computational concerns. **BART** is a computationally intensive algorithm, but its runtime increases negligibly in the number of covariates (see Chipman et al., 2010, Section 6). Hence, **BARTm** leaves the computational time virtually unchanged with the addition of the  $p_M$  new missingness dummy covariates. Another possible strategy would be to develop an iterative imputation procedure using **BART** similar to that in Stekhoven and Bühlmann (2012) or a model averaging procedure using a multiple imputation framework, but we believe these approaches would be substantially more computationally intensive.



## 8.4 Generated Data Simulations

### 8.4.1 A Simple Pattern Mixture Model

We begin with an illustration of **BARTm**'s ability to directly estimate  $\mathbb{E}[\mathbf{Y} \mid \mathbf{X}_{\text{obs}}, \mathbf{M}]$  and additionally provide uncertainty intervals. We consider the following nonlinear response surface:

$$\begin{aligned} \mathbf{Y} &= g(X_1, X_2, X_3) + \mathbf{B}M_3 + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \stackrel{iid}{\sim} \mathcal{N}(0, \sigma_e^2), \quad B \stackrel{iid}{\sim} \mathcal{N}(\mu_b, \sigma_b^2), \quad (8.4) \\ g(X_1, X_2, X_3) &= X_1 + X_2 + 2X_3 - X_1^2 + X_2^2 + X_1X_2 \\ [X_1, X_2, X_3] &\stackrel{iid}{\sim} \mathcal{N}_3 \left( \mathbf{0}, \sigma_x^2 \begin{bmatrix} 1 & \rho_1 & \rho_2 \\ \rho_1 & 1 & \rho_1 \\ \rho_2 & \rho_1 & 1 \end{bmatrix} \right), \end{aligned}$$

where  $\sigma_x^2 = 1$ ,  $\rho_1 = 0.2$ ,  $\rho_2 = 0.4$ ,  $\sigma_e^2 = 1$ ,  $\mu_b = 10$  and  $\sigma_b^2 = 0.5$ . Note that the pattern mixture model is induced by missingness in  $X_3$ . Under this missingness pattern, the response is offset by  $B$ , a draw from a normal distribution. Figure 8.1a displays the  $n = 500$  sample of the response from the model colored by  $M_3$  to illustrate the separation of the two response patterns. We choose the following jointly NMAR MDM for  $X_2$  and  $X_3$  which were chosen to be simple for the sake of ensuring that the illustration is clear. The next section features more realistic mechanisms.

$$\begin{aligned} 1 : X_2 \text{ is missing with probability } 0.3 \text{ if } X_2 \geq 0 \\ 2 : X_3 \text{ is missing with probability } 0.3 \text{ if } X_1 \geq 0. \end{aligned} \quad (8.5)$$

If the **BARTm** model assumptions hold and is successfully able to estimate the conditional expectation function  $\mathbb{E}[\mathbf{Y} \mid \mathbf{X}_{\text{obs}}, \mathbf{M}]$ , then the true  $\mathbb{E}[\mathbf{Y} \mid \mathbf{X}_{\text{obs}}, \mathbf{M}]$  is

highly likely to be contained within a 95% credible interval for the prediction. We first check to see whether BARTm can capture the correct response when  $\mathbf{X}_{\text{train}}$  has missing entries but  $\mathbf{X}_*$  does not. Predicting on  $\mathbf{x}_* = [0 \ 0 \ 0]$  should give  $\mathbb{E}[\mathbf{Y} \mid \mathbf{X} = \mathbf{x}_*] = 0$  for the prediction. Figure 8.1b illustrates that BARTm captures the expected value within its 95% credible interval.

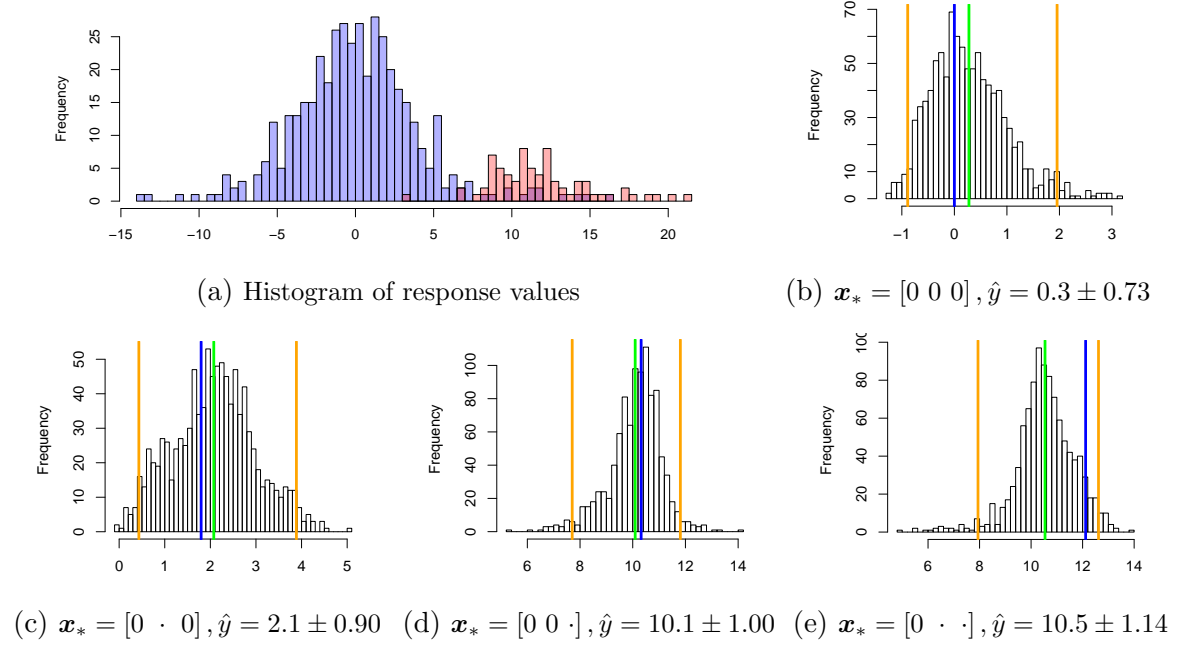


Figure 8.1: (a) A  $n = 500$  sample of the responses of the model in Equation 8.4. Colored in blue are the responses when  $X_3$  is present and red are responses when  $X_3$  is missing. (b-e) 1,000 burned-in posterior draws from a BARTm model for different values of  $\mathbf{x}_*$  drawn from the data generating process found in Equation 8.4. The **green** line is BARTm's forecast  $\hat{y}$  (the average of the posterior burned-in samples). The **blue** line is the true model expectation. The two **yellow** lines are the bounds of the 95% credible interval for  $\mathbb{E}[\mathbf{Y} \mid \mathbf{X}_{\text{obs}} = \mathbf{x}_*, \mathbf{M} = \mathbf{m}_*]$ .

Next we explore how well BARTm estimates the conditional expectation when missingness occurs within the new observation  $\mathbf{x}_*$ . We examine how BARTm handles missingness in attribute  $X_2$  by predicting on  $\mathbf{x}_* = [0 \ \cdot \ 0]$  where the “.” denotes missingness. By Equation 8.5,  $X_2$  is missing 30% of the time if  $X_2$  itself is greater than

0. By evaluating the moments of a truncated normal distribution, it follows that **BARTm** should guess  $\mathbb{E}[X_2 + X_2^2 \mid X_2 > 0] = \sqrt{2/\pi} + 1 \approx 1.80$ . Figure 8.1c indicates that **BARTm**'s credible interval captures this expected value. Note the larger variance of the posterior distribution relative to Figure 8.1b reflecting the higher uncertainty due to  $x_{*2}$  going missing. This larger interval is a great benefit of MIA. As the trees are built during the Gibbs sampling, the splitting rules on  $X_2$  are accompanied by a protocol for missingness: missing data will flow left or right in the direction that increases model likelihood and this direction is chosen with randomness. Thus, when  $\mathbf{x}_*$  is predicted with  $x_{*2}$  missing, missing records flow left and right over the many burned-in Gibbs samples creating a wider distribution of predicted values, and thus a wider credible interval. This is an important point — **BARTm** can give a rough estimate of how much information is lost when values in new records become missing by looking at the change in the standard error of a predicted value. Note that if **BART**'s hyperparameters are considered “tuning parameters,” the credible intervals' endpoints are not interpretable. However, the relative lengths of the intervals still signify different levels of forecast confidence to the practitioner.

We next consider how **BARTm** performs when  $X_3$  is missing by predicting on  $\mathbf{x}_* = [0 \ 0 \ \cdot]$ . By Equation 8.5, **BARTm** should guess  $\mathbb{E}[X_3 \mid X_1 > 0] = .4\sqrt{2/\pi} \approx .32$  (which follows directly from the properties of the conditional distribution of bivariate normal distribution, recalling that  $\text{Corr}[X_1, X_3] = 0.4$ ). When  $X_3$  is missing, there is a different response pattern, and the response is shifted up by  $B$ . Since  $\mathbb{E}[B] = 10$ , **BARTm** should predict approximately 10.32. The credible interval found in Figure 8.1d indicates that **BARTm**'s credible interval again covers the conditional expectation.

Finally, we consider the case where  $X_2$  and  $X_3$  are simultaneously missing. Predicting on  $\mathbf{x}_* = [0 \ \cdot \ \cdot]$  has a conditional expectation of  $\mathbb{E}[X_2 + X_2^2 \mid X_2 > 0] + \mathbb{E}[X_3 \mid X_1 > 0] + \mathbb{E}[B] \approx 12.12$ . Once again, the posterior draws displayed in Figure 8.1e indicate that **BARTm** reasonably estimates the conditional expectation. Note

that the credible interval here is wider than in Figure 8.1d due to the additional missingness of  $X_2$ .

### 8.4.2 Selection Model Performance

In order to gauge BARTm's out-of-sample predictive performance on selection models and to evaluate the improvement over model-building on complete cases, we construct the same model as Equation 8.4 withholding the offset  $B$  (which previously induced the pattern mixture). Thus  $\mathbf{Y} = g(X_1, X_2, X_3) + \boldsymbol{\varepsilon}$ . We imposed three scenarios illustrating performance under the following missingness mechanisms. The first is MCAR;  $X_1$  is missing with probability  $\gamma$ . The second is MAR;  $X_3$  is missing according to a non-linear probit model depending on the other two covariates:

$$\mathbb{P}(M_3 = 1 \mid X_1, X_2) = \Phi(\gamma_0 + \gamma_1 X_1 + \gamma_1 X_2^2). \quad (8.6)$$

The last is NMAR;  $X_2$  goes missing according to a similar non-linear probit model this time depending on itself and  $X_1$ :

$$\mathbb{P}(M_2 = 1 \mid X_1, X_2) = \Phi(\gamma_0 + \gamma_1 X_1^2 + \gamma_1 X_2). \quad (8.7)$$

For each simulation, we set the number of training observations to  $n = 250$  and simulate 500 times. Additionally, each simulation is carried out with different levels of missing data, approximately  $\{0, 10, \dots, 70\}$  percent of rows have at least one missing covariate entry. For the MCAR dataset, the corresponding  $\gamma = \{0, 0.03, 0.07, 0.11, 0.16, 0.26, 0.33\}$  and for both the MAR and NMAR datasets it was  $\gamma_0 = -3$  and  $\gamma_1 = \{0, 0.8, 1.4, 2.0, 2.7, 4.0, 7.0, 30\}$ .

We record results for four different scenarios: (1)  $\mathbf{X}_{\text{train}}$  and  $\mathbf{X}_*$  contain missing-

ness (2)  $\mathbf{X}_{\text{train}}$  contains missingness and  $\mathbf{X}_*$  is devoid of missing data (in this case,  $\mathbf{X}_*$  is generated without the MDM to maintain a constant number of rows). (3) only complete cases of  $\mathbf{X}_{\text{train}}$  are used to build the model but  $\mathbf{X}_*$  contains missingness and (4) only complete cases of  $\mathbf{X}_{\text{train}}$  are used to build the model and  $\mathbf{X}_*$  is devoid of missing data.

We make a number of hypotheses about the relationship between the predictive performance of using incomplete cases (all observations) compared to the complete case performance. As we discussed in Section 8.3, **BARTm** should be able model the expectation of the marginal likelihood in selection models, thus we expect models built with incomplete cases to predict better than models that were built with only the complete cases. The complete case models suffer from performance degradation for two main reasons (1) these models are built with a smaller sample size and hence their estimate of  $\mathbb{E}[\mathbf{Y} \mid \mathbf{X}_{\text{obs}}, \mathbf{M}]$  is higher in bias and variance (2) the lack of missingness during the training phase does not allow the model to learn how to properly model the missingness, resulting in the missing data being filtered randomly from node to node during forecasting. These hypotheses are explored in Figure 8.2 by comparing the solid blue and solid red lines.

Further, during forecasting, we expect  $\mathbf{X}_*$  samples with incomplete cases to have worse performance than the full  $\mathbf{X}_*$  samples (devoid of missingness) simply because missingness is equivalent to information loss. However, for the NMAR model, we expect prediction performance on  $\mathbf{X}_*$  without missingness to eventually, as the amount of missingness increases, be beaten by the predictive performance on  $\mathbf{X}_*$  with missingness. Eventually there will be so much missingness in  $X_2$  that (1) the trained model on missingness will only be able to create models by using  $\mathbf{M}_2$  and expect  $\mathbf{M}_2$  in the future  $\mathbf{X}_*$  and (2) the trained model on complete cases will never observe the response of the function where  $X_2$  went missing. These hypotheses are explored in Figure 8.2 by comparing the solid lines to the dashed lines within the same color.

The results for the four scenarios under the three MDM's comport with our hypotheses. The solid red line is uniformly higher than the solid blue line confirming degradation for complete-case model forecasting on data with missingness. The dotted lines are lower than their solid counterparts indicating that providing more covariate information yields higher predictive accuracy. The one exception is for NMAR. After the number of rows with missingness is more than 40%, forecasts on complete-cases only begin to perform worse than the forecast data with missingness for models built with missingness (**BARTm**).

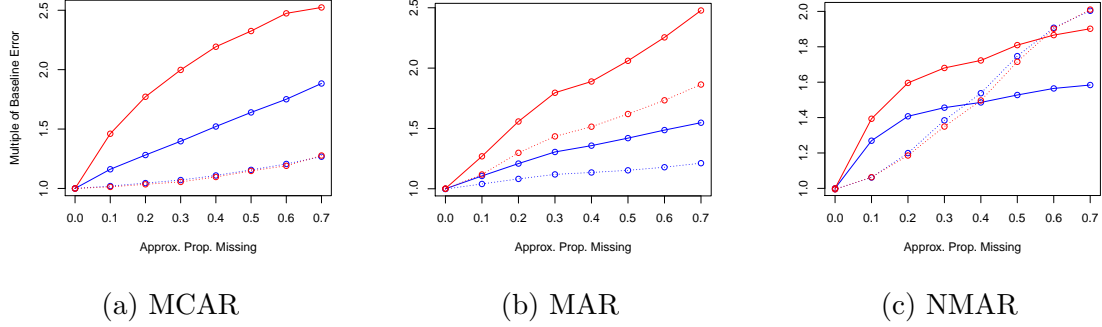


Figure 8.2: Simulation results of the response model for the three MDM's explained in the text. The  $y$ -axis measures the multiple of out-of-sample root mean square error (oosRMSE) relative to the performance under the absence of missingness. **Blue** lines correspond to the two scenarios where **BART** was built with all cases in  $\mathbf{X}_{\text{train}}$  and **red** lines correspond to the two scenarios where **BART** was built with only the complete cases of  $\mathbf{X}_{\text{train}}$ . Solid lines correspond to the two scenarios where  $\mathbf{X}_*$  included missing data and dotted lines correspond to the two scenarios where the MDM was turned off in  $\mathbf{X}_*$ .

In conclusion, for this set of simulations, **BARTm** performs better than **BART** models that ignore missingness in the training phase. The next section demonstrates **BARTm**'s performance in a real data set and compares its performance to a non-parametric statistical learning procedure that relies on imputation.

## 8.5 Real Data Example

The Boston Housing data (BHD) measures 14 features about housing in the  $n = 506$  census tracts in Boston in 1970. For model building, the response variable is usually taken to be the median home value. For this set of simulations, we evaluate the performance of three procedures (1) **BARTm** (2) **RF** with  $\mathbf{X}_{\text{train}}$  and  $\mathbf{X}_*$  imputed via **MissForest** and (3) **BART** with  $\mathbf{X}_{\text{train}}$  and  $\mathbf{X}_*$  imputed via **MissForest**. Note that in these simulations we assume *a priori* that  $\mathbf{X}_*$  will have missing data. Thus, the complete-case comparisons ala Section 8.4.2 were not possible. We gauge out-of-sample predictive performance as measured by the oosRMSE for the three procedures on the simulation scenarios described in Table 8.2.

Similar to Section 8.4.2, each simulation is carried out with different levels of missing data, approximately  $\{0, 10, 20, \dots, 70\}$  percent of rows have at least one missing covariate entry. For the MCAR scenario,  $\gamma = \{0, 0.02, 0.04, 0.07, 0.10, 0.13, 0.17\}$ , for the MAR scenario and pattern mixture scenario,  $\gamma_1 = \{0, 1.3, 1.5, 1.7, 2.1, 2.6, 3.1, 3.8\}$  and  $\gamma_0 = -3$  and for the NMAR scenario  $\gamma_1 = \{0, 3.3, 3.6, 3.9, 4.1, 4.3, 4.6, 4.8\}$  and  $\gamma_0 = -3$ . Similar to Section 8.4.1, we induce a pattern mixture model by creating a normally distributed offset based on missingness (we create two such offsets here). Here, we choose  $\mu_b$  to be 25% of the range in  $y$  and  $\sigma_b$  to be  $\mu_b/4$ . These values are arbitrarily set for illustration purposes. It is important to note that the performance gap of **BARTm** versus **RF** with imputation can be arbitrarily increased by making  $\mu_b$  larger.

For each scenario and each level of missing data, we run 500 simulations. In each simulation, we first draw missingness via the designated scenario found in Table 8.2. Then, we randomly partition 80% of the 506 BHD observations (now with missingness) as  $\mathbf{X}_{\text{train}}$  and the remaining 20% as  $\mathbf{X}_*$ . We build all three models (**BARTm**, **RF** with **MissForest** and **BART** with **MissForest**) on  $\mathbf{X}_{\text{train}}$ , forecast on  $\mathbf{X}_*$  and record

Scenario	Description
Selection Model MCAR	<code>rm</code> , <code>crim</code> , <code>lstat</code> , <code>nox</code> and <code>tax</code> are each missing w.p. $\gamma$
Selection Model MAR	<code>rm</code> and <code>crim</code> are missing according to the following models: $\mathbb{P}(\mathbf{M}_{\text{rm}} = 1) = \Phi(\gamma_0 + \gamma_1(\text{indus} + \text{lstat} + \text{age}))$ $\mathbb{P}(\mathbf{M}_{\text{crim}} = 1) = \Phi(\gamma_0 + \gamma_1(\text{nox} + \text{rad} + \text{tax}))$
Selection Model NMAR	<code>rm</code> and <code>crim</code> are missing according to the following models: $\mathbb{P}(\mathbf{M}_{\text{rm}} = 1) = \Phi(\gamma_0 + \gamma_1(\text{rm} + \text{lstat}))$ $\mathbb{P}(\mathbf{M}_{\text{crim}} = 1) = \Phi(\gamma_0 + \gamma_1(\text{crim} + \text{nox}))$
Pattern Mixture	The MAR selection model above and two offsets: (1) if $\mathbf{M}_{\text{rm}} = 1$ , the response is increased by $\mathcal{N}(\mu_b, \sigma_b^2)$ (2) if $\mathbf{M}_{\text{crim}} = 1$ , the response is decreased by $\mathcal{N}(\mu_b, \sigma_b^2)$

Table 8.2: Missingness scenarios for the BHD simulations. Monospace `codes` are names of covariates in the BHD. Note that `rm` has sample correlations with `indus`, `lstat` and `age` of -0.39, -0.61 and -0.24 and `crim` has sample correlations with `nox`, `rad`, and `tax` of 0.42, 0.63 and 0.58. These high correlations should allow for imputations that perform well.

the oosRMSE. Thus, we integrate over idiosyncrasies that could be found in a single draw from the MDM and idiosyncrasies that could be found in a single train-test partition. When using `MissForest` during training, we impute values for the missing entries in  $\mathbf{X}_{\text{train}}$  using  $[\mathbf{X}_{\text{train}}, \mathbf{y}_{\text{train}}]$  column-binded together. To obtain forecasts, we impute the missing values in  $\mathbf{X}_*$  using  $[\mathbf{X}_{\text{train}}, \mathbf{X}_*]$  row-binded together then predict using the bottom rows (i.e. those corresponding to the imputed test data). Note that we use `MissForest` in both `RF` and `BART` to ensure that the difference in predictive capabilities of `BART` and `RF` are not driving the results.

For the MCAR selection model, we hypothesize that the `MissForest`-based imputation procedures will outperform `BARTm` due to the conceptual reasons discussed

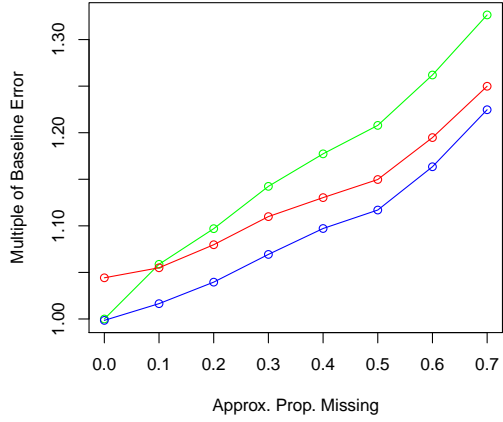


in Section 8.3. For the MAR selection model, we hypothesize similar performance between **BARTm** and both **MissForest**-based imputation procedures, as both MIA and imputation are designed to perform well in this scenario. In the NMAR selection model and pattern mixture model, we hypothesize that **BARTm** will outperform both **MissForest**-based imputation procedures, as **MissForest** (1) cannot make use of the values in the missingness columns it is trying to impute and (2) cannot construct different submodels based on missingness. Although imputation methods are not designed to handle these scenarios, it is important to run this simulation to ensure that **BARTm**, which *is* designed to succeed in these scenarios, has superior out-of-sample predictive performance.

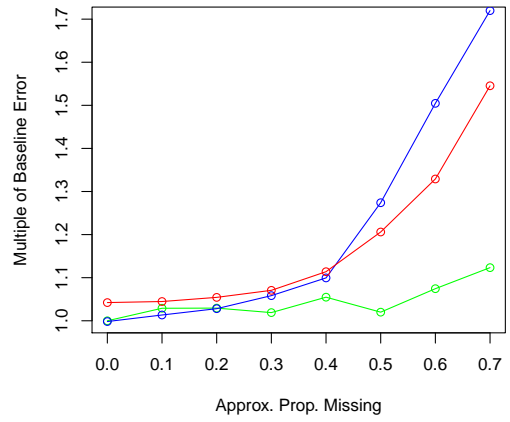
The results displayed in Figure 8.3 largely comport with our hypotheses. **MissForest**-based methods perform better on the MCAR selection model scenario (Figure 8.3a) and **BARTm** is stronger in the NMAR scenario (Figure 8.3c) and pattern mixture scenario (Figure 8.3d). It is worth noting that in the MAR selection model scenario (Figure 8.3b), **BARTm** begins to outperform the imputation-based methods once the percentage of missing data becomes greater than 20%. The performance of the imputation-based algorithms degrades rapidly here, while **BARTm**'s performance remains fairly stable, even with 70% of the rows having at least one missing entry. In conclusion, **BARTm** will generally perform better than **MissForest** because it is not “limited” to what can be imputed from the data on-hand. This advantage generally grows with the amount of missingness.

## 8.6 Discussion

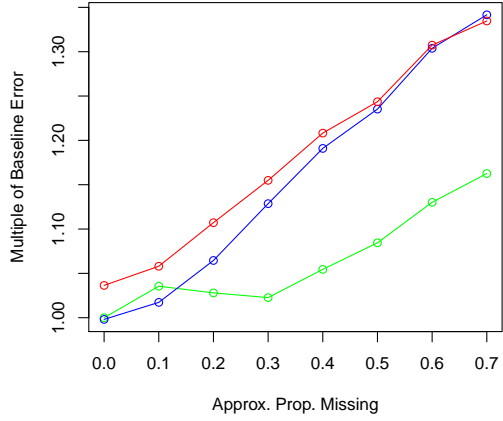
We propose a means of incorporating missing data into statistical learning for prediction problems where missingness may appear during both the training and forecast-  
ing phases. Our procedure, **BARTm**, implements “missing incorporated in attributes”



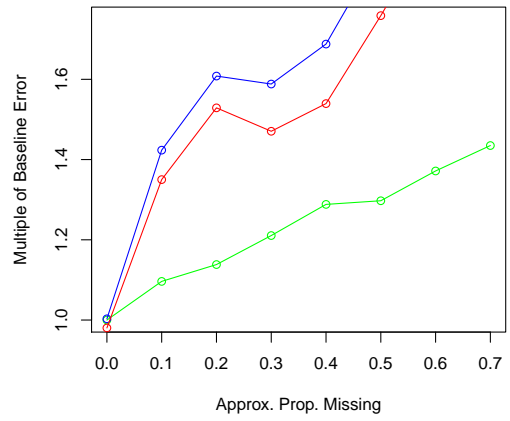
(a) MCAR



(b) MAR



(c) NMAR



(d) Pattern Mixture

Figure 8.3: Simulations for different probabilities of missingness across the four simulated missing data scenarios in the BHD. The y-axis is oosRMSE relative to BART's oosRMSE on the full dataset. Lines in **green** plot BARTm's performance, lines in **red** plot RF-with-MissForest's performance, and lines in **blue** plot BART-with-MissForest's performance. Note that the MissForest-based imputation might perform worse in practice because here we allow imputation of the entire test set. In practice, it is likely that test observations appear sequentially.

(MIA), a technique recently explored for use in decision trees, into Bayesian Additive Regression Trees, a newly developed tree-based statistical learning algorithm for classification and regression. MIA natively incorporates missingness by sending missing observations to one of the two daughter nodes. Due to the Bayesian framework and the Metropolis-Hastings sampling, missingness is incorporated into splitting rules which are chosen to increase overall model likelihood. This innovation allows missingness itself to be used as a legitimate value within splitting criteria, resulting in no need for imputing in the training or new data and no need to drop incomplete cases.

For the simulations explored in this article, **BARTm**'s performance was superior to models built using complete cases, especially when missingness appeared in the test data as well. Additionally, **BARTm** provided higher predictive performance on the MAR selection model relative to **MissForest**, a non-parametric imputation technique. We also observe promising performance on NMAR selection models and pattern mixture models in simulations. To the best of our knowledge, there is no clear evidence of other techniques that will exhibit uniformly better predictive performance in both selection and pattern mixture missingness models. Additionally, **BARTm**'s Bayesian nature provides informative credible intervals reflecting uncertainty when the forecasting data has missing covariates.

Although the exploration in this article was focused on regression, we have observed **BARTm** performing well in binary classification settings. **BARTm** for both classification and regression is implemented in the R package **bartMachine**.

Due to MIA's observed promise, we recommend it as a viable strategy to handle missingness in other tree-based statistical learning methods. Future work should also consider exploration of methods that combine imputation with MIA appropriately, in order to enhance predictive performance for MCAR missing data mechanisms.

## Supplementary Materials

Simulated results, tables, and figures can be replicated via the scripts located at <http://github.com/kapelner/bartMachine> (the home of the CRAN package `bartMachine`) in the `missing_data_paper` folder.

## Acknowledgements

We thank Richard Berk, Dana Chandler, Ed George, Dan Heitjan, Shane Jensen, and José Zubizarreta for helpful discussions. Adam Kapelner acknowledges support from the National Science Foundation's Graduate Research Fellowship Program.

## Variable Selection for BART

By Justin Bleich, Adam Kapelner, Ed George and Shane Jensen

### Abstract

We consider the task of discovering gene regulatory networks, which are defined as sets of genes and the corresponding transcription factors which regulate their expression levels. This can be viewed as a variable selection problem, potentially with high dimensionality. Variable selection is especially challenging in high dimensional settings, where it is difficult to detect subtle individual effects and interactions between factors. Bayesian additive regression trees (BART, Chipman et al., 2010) provides a novel nonparametric alternative to parametric regression approaches, such as the lasso or stepwise regression, especially when the number of relevant predictors is sparse relative to the total number of available predictors and the fundamental relationships are nonlinear. We develop a principled permutation-based inferential approach for determining when the effect of a selected predictor is likely to be real. Going further, we adapt the BART procedure to incorporate informed prior information about variable importance. We present simulations demonstrating that our method compares favorably to existing parametric and nonparametric procedures in a variety of data settings. To demonstrate the potential of our approach in a biological context,

we apply it to the task of inferring the gene regulatory network in yeast (*Saccharomyces cerevisiae*). We find that our BART-based procedure is best able to recover the subset of covariates with the largest signal compared to other variable selection methods. The methods developed in this work are readily available in the R package `bartmachine`.

## 9.1 Introduction

An important statistical problem in many application areas is variable selection: identifying the subset of covariates that exert influence on a response variable. We consider the general framework where we have a continuous response variable  $\mathbf{y}$  and a large set of predictor variables  $\mathbf{x}_1, \dots, \mathbf{x}_K$ . We focus on variable selection in the sparse setting: only a relatively small subset of those predictor variables truly influences the response variable.

One example of a sparse setting is the motivating application for this paper: inferring the gene regulatory network in budding yeast (*Saccharomyces cerevisiae*). In this application, we have a collection of approximately 40 transcription factor proteins (TFs) that act to regulate cellular processes in yeast by promoting or repressing transcription of specific genes. It is unknown which of the genes in our yeast data are regulated by each of the transcription factors. Therefore, the goal of the analysis is to discover the corresponding network of gene-TF relationships, which is known as a *gene regulatory network*. Each gene, however, is regulated by only a small subset of the TFs which makes this application a sparse setting for variable selection. The available data consists of gene expression measures for approximately 6000 genes in yeast across several hundred experiments, as well as expression measures for each of the approximately 40 transcription factors in those experiments (Jensen et al., 2007).

This gene regulatory network was previously studied in Jensen et al. (2007) with

a focus on modeling the relationship between genes and transcription factors. The authors considered a Bayesian linear hierarchical model with first-order interactions. In high-dimensional data sets, specifying even first-order pairwise interactions can substantially increase the complexity of the model. Additionally, given the elaborate nature of biological processes, there may be interest in exploring nonlinear relationships as well as higher-order interaction terms. In such cases, it may not be possible for the researcher to specify such terms in a linear model *a priori*. Indeed, Jensen et al. (2007) acknowledge the potential utility of such additions, but highlight the practical difficulties associated with the size of the resulting parameter space. Thus, we propose a variable selection procedure that relies on the nonparametric Bayesian model, Bayesian Additive Regression Trees (BART, Chipman et al., 2010). BART dynamically estimates a model from the data, thereby allowing the researcher to potentially identify genetic regulatory networks without the need to specify higher order interaction terms or nonlinearities ahead of time.

Additionally, we have data from chromatin immunoprecipitation (ChIP) binding experiments (Lee et al., 2002). Such experiments use antibodies to isolate specific DNA sequences which are bound by a TF. This information can be used to discover potential binding locations for particular transcription factors within the genome. The ChIP data can be considered “prior information” that one may wish to make use of when investigating gene regulatory networks. Given the Bayesian nature of our approach, we propose a straightforward modification to BART which incorporates such prior information into our variable selection procedure.

In Section 9.2, we review some common techniques for variable selection. We emphasize the limitations of approaches relying on linear models and highlight variable selection via tree-based techniques. We provide an overview of the BART algorithm and its output in Section 9.3.1. In Sections 9.3.2 and 9.3.3, we introduce how BART computes variable inclusion proportions and explore the properties of these propor-

tions. In Section 9.3.4, we develop procedures for principled variable selection based upon BART output. In Section 9.3.5 we extend the BART procedure to incorporate prior information about predictor variable importance. In Section 13.3, we compare our methodology to alternative variable selection approaches in both linear and nonlinear simulated data settings. In Section 9.5, we apply our BART-based variable selection procedure to the discovery of gene regulatory networks in budding yeast. Section 13.5 concludes with a brief discussion. We note that our variable selection procedures as well as the ability to incorporate informed prior information are readily available features in the R package `bartMachine` (Kapelner and Bleich, 2013), currently available on CRAN.

## 9.2 Techniques for Variable Selection

### 9.2.1 Linear Methods

The variable selection problem has been well-studied from both the classical and Bayesian perspective, though most previous work focuses on the case where the outcome variable is assumed to be a linear function of the available covariates. Stepwise regression (Hocking, 1976) is a common approach for variable selection from a large set of possible predictor variables. Best subsets regression (Miller, 2002) can also be employed, although this option becomes too computationally burdensome as  $K$  becomes large. Other popular linear variable selection methods are lasso regression (Tibshirani, 1996) and the elastic net (Zou and Hastie, 2005). Both of these approaches enforce sparsity on the subset of selected covariates by imposing penalties on non-zero coefficients. Park and Casella (2008) and Hans (2009) provide Bayesian treatments of lasso regression.

Perhaps the most popular Bayesian variable selection strategies are based on linear



regression with a “spike-and-slab” prior distribution on the regression coefficients. Initially proposed by Mitchell and Beauchamp (1988), who used a mixture prior of a point mass at zero and a uniform slab, George and McCulloch (1993) went on to use a mixture-of-normals prior, for which Markov Chain Monte Carlo stochastic search of the posterior could be implemented. Eventually, most applications gravitated towards a limiting form of the normal mixture with a degenerate point mass at zero. More recent work involving spike-and-slab models has been developed in Ishwaran and Rao (2005), Li and Zhang (2010), Hans et al. (2007), Bottolo and Richardson (2010), Stingo and Vannucci (2011) and Rockova and George (2013). In these approaches, variable selection is based on the posterior probability that each predictor variable is in the slab distribution, and sparsity can be enforced by employing a prior that strongly favors the spike distribution at zero.

### 9.2.2 Tree-Based Methods

Each of the aforementioned approaches assumes that the response variable is a linear function of the predictor variables. A major drawback to linear models, both in the frequentist and Bayesian paradigms, is that they are ill-equipped to handle complex, nonlinear relationships between the predictors and response. Nonlinearities and interactions, which are seldom known with certainty, must be specified in advance by the researcher. In the case where the model is mis-specified, incorrect variables may be included and correct variables excluded which can hurt predictive performance.

As an alternative, we consider nonparametric methods which are flexible enough to fit a wide array of functional forms. We focus on tree-based methods, examples of which include random forests (RF, Breiman, 2001b), stochastic gradient boosting (Friedman, 2002) BART, and dynamic trees (DT, Taddy et al., 2011). Compared with linear models, these procedures are better able to approximate complicated response surfaces but are “black-boxes” in the sense that they offer less insight into how specific

predictor variables relate to the response variable.

Tree-based variable selection makes use of the internals of the decision tree structure which we briefly outline. All observations begin in a single root node. The root node’s decision rule is chosen and consists of a splitting variable  $\mathbf{x}_k$  and a split point  $c$ . The observations in the root node are then split into two groups based on whether  $\mathbf{x}_k \geq c$  or  $\mathbf{x}_k < c$ . These two groups become a right daughter node and a left daughter node respectively. Within each of these two nodes, additional binary splits can be chosen.

Existing tree-based methods for variable selection focus on the set of splitting variables within the trees. For example, Gramacy et al. (2013) develop a backward stepwise variable selection procedure for DT by considering the average reduction in posterior predictive uncertainty within all nodes that use a particular predictor as the splitting variable. Also, the splitting variables in RF can also be used to develop variable selection approaches. For instance, one can consider the reduction in sum of square errors (node impurity in classification problems) associated with a particular predictor. Additionally, Díaz-Uriarte and Alvarez de Andrés (2006) consider reduction in out-of-bag mean square error associated with each predictor to develop a backward stepwise selection procedure.

We too consider the splitting variables for BART in developing our method, but our approach differs from the previously mentioned work in two aspects. First, we do not propose a backwards stepwise selection, but rather develop a permutation-based inferential approach. Second, we do not consider the overall improvement to fit provided by each predictor, but instead consider how often a particular predictor appears in a BART model. While simple, this metric shows promising performance for variable selection using BART.

## 9.3 Calibrating BART output for Variable Selection

### 9.3.1 Review of Bayesian Additive Regression Trees

BART is a Bayesian ensemble approach for modeling the unknown relationship between a vector of observed responses  $\mathbf{y}$  and a set of predictor variables  $\mathbf{x}_1, \dots, \mathbf{x}_p$  without assuming any parametric functional form of the relationship. The key idea behind BART is to model the regression function by a sum of regression trees with homoskedastic normal additive noise,

$$\mathbf{y} = \sum_{i=1}^m \mathcal{T}_i(\mathbf{x}_1, \dots, \mathbf{x}_K) + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}_n(\mathbf{0}, \sigma^2 \mathbf{I}_n). \quad (9.1)$$

Here, each  $\mathcal{T}_i(\mathbf{x}_1, \dots, \mathbf{x}_K)$  is a regression tree that partitions the predictor space based on the values of the predictor variables. Observations with similar values of the predictor variables are modeled as having a similar predicted response  $\hat{\mathbf{y}}$ .

Each regression tree  $\mathcal{T}_i$  consists of two components: a tree structure  $T_i$  and a set of terminal node parameters  $\boldsymbol{\mu}_i$ . The tree  $T_i$  partitions each observation into a set of  $B_i$  terminal nodes based on the decision rules contained in the tree. The terminal nodes are parameterized by  $\boldsymbol{\mu}_i = \{\mu_{i1}, \dots, \mu_{iB_i}\}$  such that each observation contained within terminal node  $b$  is assigned the same response value of  $\mu_{ib}$ . Regression trees yield a flexible model that allows for nonlinearities and interaction effects in the unknown regression function.

As we see in Equation 9.1, the response vector  $\mathbf{y}$  is modeled by the sum of  $m$  regression trees. For each observation, the predicted response  $\hat{y}_j$  is the sum of the terminal node parameters  $\mu_{ib}$  for that observation  $j$  from each tree  $\mathcal{T}_i$ . Compared to single tree, the sum of trees allows for easier modeling of additive effects (Chipman et al., 2010). The residual variance  $\sigma^2$  is considered a global parameter shared by all

observations.

In this fully Bayesian approach, prior distributions must also be specified for all unknown parameters, which are all of the tree structures and terminal node parameters  $(T_i, \boldsymbol{\mu}_i)$ , as well as the residual variance  $\sigma^2$ . The prior distributions for  $(T_i, \boldsymbol{\mu}_i)$  are specified to give a strong preference to small simple trees with modest variation of the terminal node parameter values, thereby limiting the impact on the model fit of any one tree. The result is that **BART** consists of an ensemble of “weak learners,” each contributing to the approximation of the unknown response function in a small and distinct fashion. The prior for  $\sigma^2$  is the inverse chi-square distribution with hyperparameters chosen based on an estimate of the residual standard deviation of the data.

The number of trees  $m$  in the ensemble is considered to be a pre-specified hyperparameter. The usual goal of **BART** is predictive performance, in which case a large value of  $m$  allows for increased flexibility when fitting a complicated response surface, thereby improving predictive performance. However, Chipman et al. (2010) recommend using a smaller value of  $m$  for the purposes of variable selection (we default to  $m = 20$ ). When the number of trees in the ensemble is smaller, there are fewer opportunities for predictor variables to appear in the model and so they must compete with each other to be included. However, if  $m$  is too small, the Gibbs sampler in **BART** becomes trapped in local modes more often which can destabilize the results of the estimation procedure (Chipman et al., 1998). Also, there is not enough flexibility in the model to fit a variety of complicated functions. However, when the number of trees becomes too large, there is opportunity for unimportant variables to enter the model without impacting the overall model fit, thereby making variable selection more challenging.

Our explorations have shown that  $m = 20$  represents a good compromise, although similar choices of  $m$  should not impact results. Under the sparse data settings we

will examine in Sections 13.3 and 9.5, we show that this medium level of  $m$  aids the selection of important predictor variables even when the number of predictor variables is relatively large.

It is also worth noting that in the default **BART** formulation, each predictor variable  $\mathbf{x}_k$  has an equal *a priori* chance of being chosen as a splitting rule for each tree in the ensemble. However, in many applications, we may have real prior information that suggests the importance of particular predictor variables. In Section 9.3.5, we will extend the **BART** procedure to incorporate prior information about specific predictor variables, which will be used to aid in discovering the yeast gene regulatory network in Section 9.5.

The full posterior distribution for the **BART** model is estimated using Markov chain Monte Carlo methods. Specifically, a Gibbs sampler (Geman and Geman, 1984) is used to iteratively sample from the conditional posterior distribution of each set of parameters. Most of these conditional posterior distributions are standard, though a Metropolis-Hastings step (?) is needed to alter the tree structures  $T_i$ . Details are given in Chipman et al. (2010) and Kapelner and Bleich (2013).

### 9.3.2 BART Variable Inclusion Proportions

The primary output from **BART** is a set of predicted values  $\hat{\mathbf{y}}$  for the response variable  $\mathbf{y}$ . Although these predicted values  $\hat{\mathbf{y}}$  serve to describe the overall fit of the model, they are not directly useful for evaluating the relative importance of each predictor variable in order to select a subset of predictor variables. For this purpose, Chipman et al. (2010) begin exploring the “variable inclusion proportions” of each predictor variable. We extend their exploration into a principled method.

Across all  $m$  trees in the ensemble (Equation 9.1), we examine the set of predictor variables used for each splitting rule in each tree. Within each posterior Gibbs sample, we can compute the proportion of times that a split using  $\mathbf{x}_k$  as a splitting variable

appears among all splitting variables in the ensemble. Since the output of BART consists of many posterior samples, so we estimate the *variable inclusion proportion*  $p_k$  as the posterior mean of the these proportions across all of these posterior samples.

Intuitively, a large variable inclusion proportion  $p_k$  is suggestive that variable  $\mathbf{x}_k$  is an important driver of the response. Chipman et al. (2010) suggest using  $\mathbf{p} = (p_1, \dots, p_K)$  to rank variables  $\mathbf{x}_1, \dots, \mathbf{x}_K$  in terms of relative importance. These variable inclusion proportions naturally build in some amount of multiplicity control since the  $p_k$ 's have a fixed budget (in that they must sum to one) and that budget will become more restrictive as the number of predictor variables increases.

However, each variable inclusion proportion  $p_k$  cannot be interpreted as a posterior probability that the predictor variable  $\mathbf{x}_k$  has a “real effect”, defined as the impact of some linear or nonlinear association, on the response variable. The primary question being addressed by this paper is *how large does the variable inclusion proportion  $p_k$  have to be in order to select predictor variable  $\mathbf{x}_k$  as an important variable?*

As a preliminary study, we evaluate the behavior of the variable inclusion proportions in a “null” data setting, where we have a set of  $K$  predictor variables  $\mathbf{x}_k$  that are all unrelated to the outcome variable  $\mathbf{y}$ . Specifically, we generate each response variable  $y_i$  and each predictor variable  $x_{ik}$  independently from a standard normal distribution. In this null setting, one might expect that BART would choose amongst the predictor variables uniformly at random when adding variables to the ensemble of trees (Equation 9.1). In this scenario, each variable inclusion proportion would then be close to the inverse of the number of predictor variables, i.e.  $p_k \approx 1/K$  for all  $k$ .

However, we have found empirically that in this scenario the variable inclusion proportions do not approach  $1/K$  for all predictor variables. As an example, Figure 9.1 gives the variable inclusion proportions from a null simulation with  $n = 250$  observations and  $K = 40$  predictor variables, all of which are unrelated to the response variable  $\mathbf{y}$ .

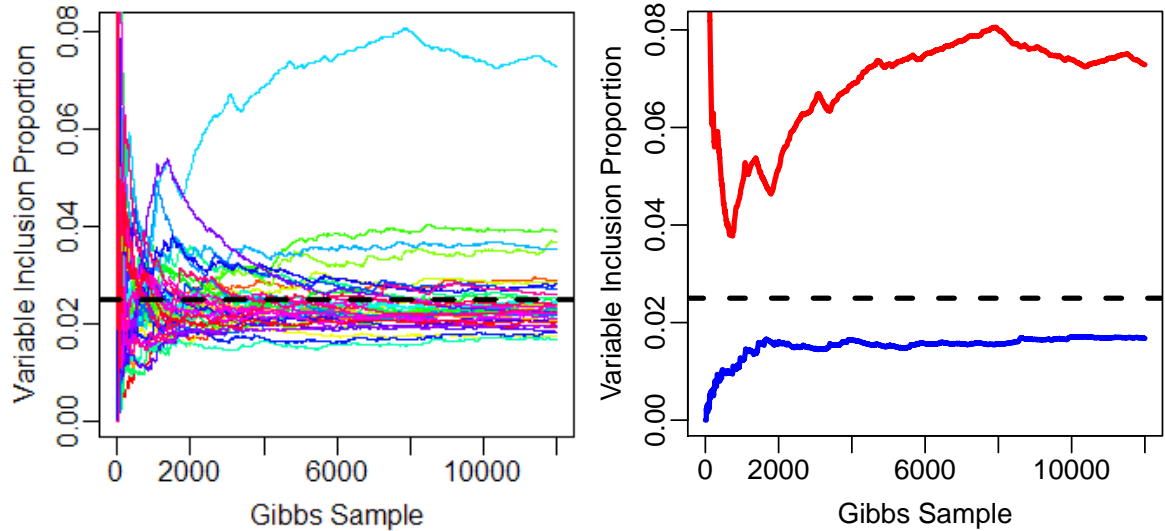


Figure 9.1: Variable inclusion proportions from BART model in null setting where each predictor variable is unrelated to the response variable. (a) variable inclusion proportions for all  $K = 40$  predictor variables over 12,000 Gibbs samples. (b) maximum and minimum of these variable inclusion proportions.

In this setting, the variable inclusion proportions do not converge to  $1/40 = .025$ . As we see in Figure 9.1, some variable inclusion proportions remain substantially larger than  $1/K$  and some are substantially lower. We observed this same phenomenon with different levels of noise in the response variable.

### 9.3.3 Further Exploration of Null Simulation

We hypothesize that the variation between  $p_k$ 's in Figure 9.1 can stem from two causes. First, even though the response and predictors were generated independently, they will still exhibit some random association. BART may be fitting noise or “chance capitalizing;” given its nonparametric flexibility, BART could be capitalizing on perceived nonlinear associations. Second, there might be inherent variation in the BART estimation procedure itself, possibly due to the Gibbs sampler getting stuck in a local maximum.

Thus, we consider an experiment to explore the source of this variation among the  $p_k$ 's. We generate 100 data sets under the same setting as that in Figure 9.1. Within each data set, we run **BART** 50 times with different initial values for the model parameters randomly drawn from the respective prior distributions. Let  $p_{ijk}$  denote the variable inclusion proportion for the  $i^{th}$  data set,  $j^{th}$  **BART** run and the  $k^{th}$  predictor. We then consider the decomposition into three nested variances listed in Table 9.1. Note that we use standard deviations in our illustration that follows.

Table 9.1: The Three Nested Variances.

$$\begin{aligned}
s_{ik}^2 &= \frac{1}{50} \sum_{j=1}^{50} (p_{ijk} - \bar{p}_{i \cdot k})^2 && \text{This is the variability of } \mathbf{BART} \text{ estimation for a particular predictor } k \text{ in a particular data set } i. \\
s_k^2 &= \frac{1}{100} \sum_{i=1}^{100} (\bar{p}_{i \cdot k} - \bar{p}_{\cdot \cdot k})^2 && \text{This is the variability due to chance capitalization of the } \mathbf{BART} \text{ procedure for predictor } k \text{ across datasets.} \\
s^2 &= \frac{1}{40} \sum_{k=1}^{40} (\bar{p}_{\cdot \cdot k} - \bar{p}_{\cdot \cdot \cdot})^2 && \text{This is the variability across predictors.}
\end{aligned}$$

First consider what may happen if the source of Figure 9.1's observed pathology is purely due to **BART**'s Gibbs sampler getting stuck in different local posterior modes. On the first run on the first data set, **BART** would fall into a local mode where some predictors are naturally more important than others and hence the  $p_{11k}$ 's would be unequal. In the same dataset, second run, **BART** might fall into a different local mode where the  $p_{12k}$ 's are unequal, but in a way that is different from the first run's  $p_{11k}$ 's. This type of process would occur over all 50 runs. Thus, the  $s_{1k}$ , the standard deviation of  $p_{ijk}$  over runs of **BART** on the first dataset, would be large. Note that if there is no chance capitalization or overfitting, there should be no reason that averages of the proportions, the  $\bar{p}_{1 \cdot k}$ 's, should be different from  $1/K$  over repeated runs. Then, when the second data set is introduced, **BART** will continue to get stuck in different local posterior modes and the  $s_{2k}$ 's should be large, but the  $\bar{p}_{2 \cdot k}$ 's should be near  $1/K$ . Hence, over all of the datasets,  $\bar{p}_{i \cdot k}$ 's should be approximately  $1/K$ ,



implying that the  $s_k$ 's should be low. In sum, BART getting stuck in local modes suggests high  $s_{ik}$ 's and low  $s_k$ 's.

Next consider what may happen if the source of Figure 9.1's observed pathology is purely due to BART chance capitalizing on noise. On the first data set, over each run, BART does not get stuck in local modes, and therefore the  $p_{i1k}$ 's across runs would be fairly stable. Hence, the  $s_{1k}$ 's would be small. However, in each of the runs, BART overfits in the *same way* for each dataset. For example, perhaps BART perceives an association between  $x_1$  and  $y$  on the first data set. Hence, the  $p_{1j1}$ 's would be larger than  $1/K$  on all restarts (BART would select  $x_1$  as a splitting rule often due to the perceived association) and thus  $\bar{p}_{1\cdot 1} > 1/K$ . Then, in the second data set, BART may perceive an association between  $x_3$  and  $y$ , resulting in  $p_{2j3}$ 's being larger on all runs ( $\bar{p}_{2\cdot 3} > 1/K$ ). Thus, BART overfitting is indicated by small  $s_{ik}$ 's and high  $s_k$ 's.

Figure 9.2 illustrates the results of the simulations. Both sources of variation appear but for all predictors, the average  $s_{ik}$  is significantly smaller than the  $s_k$ . This finding suggests that within a particular data set, BART is chance-capitalizing and overfitting to the noise, which prevents the  $p_k$  from converging to  $1/K$ .<sup>1</sup>

Also note the overall average inclusion proportion  $\bar{p}_{\dots}$  is  $.025 = 1/K$ , so across data sets and BART runs the variable inclusion proportions are correct on average. Further, the standard deviation across predictors  $s$  is small. This implies that the  $\bar{p}_{\cdot k}$ 's are approximately  $1/K$  as well, which indicates there is no systematic favoring of different covariates once the effect of overfitting by dataset and remaining in local modes by run is averaged out.

We believe this experiment demonstrates that there is a large degree of chance capitalization present in the variable inclusion proportions in the “null” model. This

---

<sup>1</sup>We also considered this experiment with orthogonalized predictors (not shown). This reduces the  $s_k$ 's (chance capitalization) in Figure 9.2 slightly, but the  $s_k$ 's are still larger than the average  $s_{ik}$ 's. Hence, even if there is no *linear* correlation between the predictors and the response, BART is capitalizing on nonlinear associations.

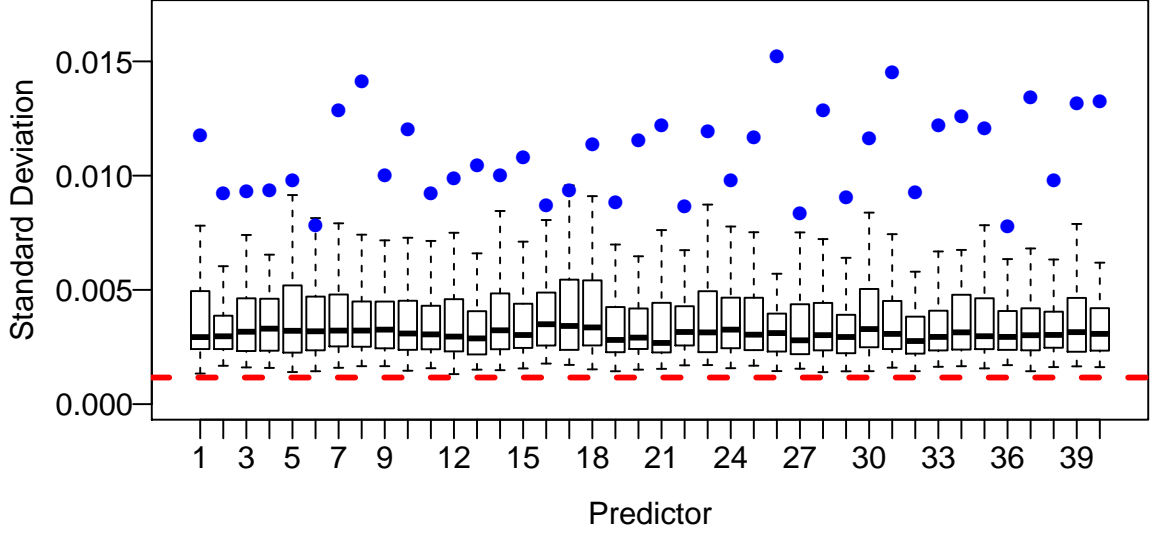


Figure 9.2: The boxplots represent the distribution of  $s_{ik}$  for each predictor. The circles represent the  $s_k$  and the dashed line corresponds to  $s$ . Note that the results are reported as standard deviations and points in the boxplots beyond the whiskers are removed.

means it is not possible to decide on an appropriate threshold for the  $p_k$ 's when selecting a subset of important predictor variables in real data settings. Further, the chance capitalization is idiosyncratic for any data set, making it challenging to pose a simple parametric model for the behavior in Figure 9.1 that would be useful in practice. This motivates our nonparametric approach to establishing thresholds for the variable inclusion proportions based on permutations of the response variable  $\mathbf{y}$ .

As noted above, there is some variability in the  $p_k$ 's between **BART** runs from different starting points. We found that averaging over results from five repetitions of the **BART** algorithm from different starting points was sufficient to provide stable estimates of the variable inclusion proportions and use these averaged values as our variable inclusion proportions for the remainder of the article.

### 9.3.4 Variable Inclusion Proportions under Permuted Responses

We now address our key question: *how large does the variable inclusion frequency  $p_k$  have to be in order to select predictor variable  $\mathbf{x}_k$ ?* To determine an appropriate selection threshold, we employ a permutation-based approach to generate a null distribution of our variable inclusion proportions  $\mathbf{p} = (p_1, \dots, p_K)$ .

Specifically, we create  $P$  permutations of the response vector:  $\mathbf{y}_1^*, \mathbf{y}_2^*, \dots, \mathbf{y}_P^*$ . For each of these permuted response vectors  $\mathbf{y}_p^*$ , we run the BART model using  $\mathbf{y}_p^*$  as the response and the original  $\mathbf{x}_1, \dots, \mathbf{x}_K$  as predictor variables. This permutation strategy preserves possible dependencies among the predictor variables while removing any dependency between the predictor variables and the response variable.

We retain the variable inclusion proportions estimated from the BART run using each permuted response  $\mathbf{y}_p^*$ . We use the notation  $p_{k,p}^*$  for the variable inclusion proportion from BART for predictor  $\mathbf{x}_k$  from the  $p$ -th permuted response, and we use the notation  $\mathbf{p}_p^*$  for the vector of all variable inclusion proportions from the  $p$ -th permuted response. We use the variable inclusion proportions  $\mathbf{p}_1^*, \mathbf{p}_2^*, \dots, \mathbf{p}_P^*$  across all  $P$  permutations as the null distribution for our variable inclusion proportions  $\mathbf{p}$  from the real (unpermuted) response  $\mathbf{y}$ .

The remaining issue is selecting an appropriate threshold for predictor  $\mathbf{x}_k$  based on the permutation null distribution  $\mathbf{p}_1^*, \mathbf{p}_2^*, \dots, \mathbf{p}_P^*$ . We will consider three different threshold strategies that vary in terms of the stringency of their resulting variable selection procedure.

The first strategy is a “**local**” threshold: we calculate a threshold for each variable inclusion proportion  $p_k$  for each predictor  $\mathbf{x}_k$  based only on the permutation null distribution of  $p_k$ . Specifically, we take the  $1 - \alpha$  quantile of the distribution of  $p_{k,1}^*, p_{k,2}^*, \dots, p_{k,P}^*$  and only select predictor  $\mathbf{x}_k$  if  $p_k$  exceeds this  $1 - \alpha$  quantile.

The second strategy is a “**global max**” threshold: we calculate a threshold for

the variable inclusion proportion  $p_k$  for predictor  $\mathbf{x}_k$  based on the maximum across the permutation distributions of the variable inclusion proportions for all predictor variables. Specifically, we first calculate  $p_{\max,p}^* = \max \{p_{1,p}^*, p_{2,p}^*, \dots, p_{K,p}^*\}$ , the largest variable inclusion proportion across all predictor variables in permutation  $p$ . We then calculate the  $1 - \alpha$  quantile of the distribution of  $p_{\max,1}^*, p_{\max,2}^*, \dots, p_{\max,P}^*$  and only select predictor  $\mathbf{x}_k$  if  $p_k$  exceeds this  $1 - \alpha$  quantile.

The first “local” strategy and the second “global max” strategy are opposite extremes in terms of the stringency of the resulting variable selection. The local strategy is least stringent since the variable inclusion proportion  $p_k$  for predictor  $\mathbf{x}_k$  needs to only be extreme within its own permutation distribution to be selected. The global maximum strategy is most stringent since the variable inclusion proportion  $p_k$  for predictor  $\mathbf{x}_k$  must be extreme relative to the permutation distribution across all predictor variables in order to be selected.

We consider a third strategy that is also global across predictor variables, but is less stringent than the global max strategy. The third “**global SE**” strategy uses the mean and standard deviation from the permutation distribution of each variable inclusion proportion  $p_k$  to create a global threshold for all predictor variables. Specifically, letting  $m_k$  and  $s_k$  be the mean and standard deviation of variable inclusion proportion  $p_k^*$  for predictor  $\mathbf{x}_k$  across all permutations, we calculate

$$C^* = \inf_{C \in \mathbb{R}^+} \left\{ \forall k, \frac{1}{P} \sum_{p=1}^P \mathbb{1}_{p_{k,p}^* \leq m_k + C \cdot s_k} > 1 - \alpha \right\}.$$

The value  $C^*$  is the smallest global multiplier that gives simultaneous  $1 - \alpha$  coverage across the permutation distributions of  $p_k$  for all predictor variables  $k$ . The predictor  $\mathbf{x}_k$  is then only selected if  $p_k > m_k + C^* \cdot s_k$ . This third strategy is a compromise between the local permutation distribution for variable  $k$  (by incorporating each mean  $m_k$  and standard deviation  $s_k$ ) and the global permutation distributions

of the other predictor variables (through  $C^*$ ). We outline all three thresholding procedures in more detail in Appendix A.3.1.

As an example of these three thresholding strategies, we provide a brief preview of our application to the yeast gene regulatory network in Section 9.5. In that application, the response variable  $\mathbf{y}$  consists of the expression measures for a particular gene across approximately 300 conditions and the predictor variables are the expression values for approximately 40 transcription factors in those same 300 conditions.

In Figure 9.3, we give the fifteen predictor variables with the largest variable inclusion proportions from the BART model implemented on the data for a particular yeast gene YAL004W. In the top plot, we see the different “local” thresholds for each predictor variable. Four of the predictor variables had variable inclusion proportions  $p_k$  that exceeded their local threshold and were selected under this first strategy. In the bottom plot, we see the single “global max” threshold for all predictor variables as well as the different “global SE” thresholds for each predictor variable. Two of the predictor variables had variable inclusion proportions  $p_k$  that exceeded their global SE thresholds whereas only one predictor variable exceeded the global max threshold.

This example illustrates that our three threshold strategies can differ substantially in terms of the stringency of the resulting variable selection. Depending on our prior expectations about the sparsity in our predictor variables, we may prefer the high stringency of the global max strategy, the low stringency of the local strategy, or the intermediary global SE strategy.

In practice, it may be difficult to know *a priori* the level of stringency that is desired for a real data application. Thus, we propose a **cross-validation strategy** for deciding between our three thresholding strategies for variable selection. Using k-fold cross validation, the available observations can be partitioned into training and holdout subsets. For each partition, we can implement all three thresholding strategies on the training subset of the data and use the thresholding strategy with

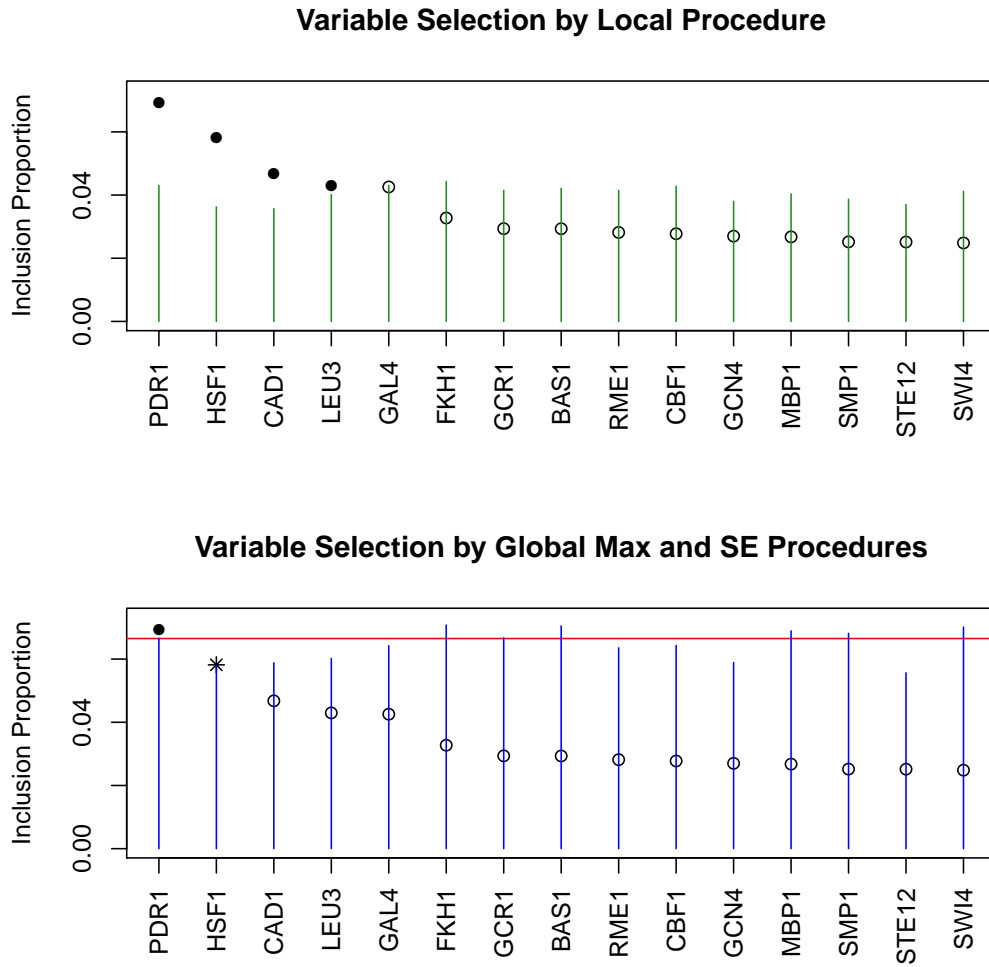


Figure 9.3: The fifteen largest variable inclusion proportions from BART implemented on the yeast gene YAL004W with  $\alpha = .05$ . **Top:** The tips of the green bands are the “local” thresholds of our first variable selection strategy. Solid dots are selected predictor variables whereas hollow dots are unselected predictor variables. **Bottom:** The red line is the threshold from our second “global max” strategy. The tips of the blue bands are the thresholds from our “global SE” strategy. The one solid dot is the predictor selected by both strategies. The star is the additional predictor variable selected by only the global SE strategy. The hollow dots are unselected predictor variables.

the smallest the prediction error across the holdout subsets. We call this procedure “BART-Best” and provide implementation details in Appendix A.3.1.

Our permutation-based approach for variable selection does not require any additional assumptions beyond those of the BART model. Once again, the sum-of-trees plus normal errors is a flexible assumption that should perform well across a wide range of data settings, especially relative to methods that make stronger parametric demands. Also, it is important to note that we view each of the strategies described in this section as a procedure for variable selection based on well-founded statistical principles, but do not actually associate any particular formal hypothesis testing with our approach. Finally, a disadvantage of our permutation-based proposal is the computational cost of running BART on a large set of permuted response variables  $\mathbf{y}^*$ . However, it should be noted that the permuted response vector runs can be parallelized on multiple computing cores when these resources are available.

### 9.3.5 Real Prior Information in BART-based Variable Selection

Most variable selection approaches do not allow for *a priori* preferences for particular predictor variables. However, in many applications, there may be available prior information that suggests particular predictor variables to be more valuable than others.

As an example, the yeast regulatory data in Section 9.5 consist of expression measures  $\mathbf{y}_g$  for a particular gene  $g$  as the response variable with predictor variables  $\mathbf{x}_k$  being the expression values for  $\approx 40$  transcription factors (TFs). In addition to the expression data, we also have an accompanying ChIP-binding dataset (Lee et al., 2002) that indicates for each gene  $g$  which of the  $\approx 40$  transcription factors are likely to bind near that gene. We can view these ChIP-binding measures as prior probabilities that particular predictor variables  $\mathbf{x}_k$  will be important for the response variable  $\mathbf{y}$ .

The most natural way to give prior preference to particular variables in **BART** is to increase the likelihood they are chosen when suggesting splitting rules in the Metropolis-Hastings tree-altering step. As mentioned in Section 9.3.1, each predictor  $\mathbf{x}_k$  has an equal *a priori* chance of being chosen as a splitting rule for each tree branch in the **BART** ensemble. We propose altering the standard **BART** implementation so that, when randomly selecting a particular predictor variable for a splitting rule, more weight is given to the predictor variables that have a higher prior probability of being important. Additionally, the prior on the tree structure, which is needed for the Metropolis-Hastings step, is appropriately adjusted. This strategy has some precedent as Chipman et al. (1998) discuss non-uniform criteria for splitting rules in the context of an earlier Bayesian Classification and Regression Tree implementation. Note that when employing one of the strategies discussed in Section 9.3.4, the prior is reset to discrete uniform when generating the permutation distribution as it is assumed that there is no relationship between the predictors and the response.

In Section 9.4.3, we present a simulated-data evaluation of the effects on correct variable selection when an informed prior distribution is either correctly specified, giving additional weight to the predictor variables with true influence on the response, or incorrectly specified, giving additional weight to predictor variables that are unrelated to the response. Before our simulation study of the effects of prior information, we first present an extensive simulation study that compares our **BART**-based variable selection procedure to several other approaches.

## 9.4 Simulation Evaluation of **BART**-based Variable Selection

We use a variety of simulated data settings to evaluate the ability of our **BART**-based procedure to select the subset of predictor variables that have a true influence on a



response variable. We examine settings where the response is a linear function of the predictor variables in Section 9.4.1 as well as settings where the response is a nonlinear function of the predictor variables in Section 9.4.2. We also examine the effects of correctly versus incorrectly specified informed prior distributions in Section 9.4.3. For each simulated data setting, we will compare the performance of several different variable selection approaches:

1. **BART-based Variable Selection:** As outlined in Section 9.3, we use the variable inclusion proportions from the BART model to rank and select predictor variables. We will evaluate the performance of the three proposed thresholding strategies as well as “BART-Best,” the (five-fold) cross-validation strategy for choosing amongst our thresholding strategies. In each case, we set  $\alpha = 0.05$  and the number of trees  $m$  is set to 20. Default settings from Chipman et al. (2010) are used for all other hyperparameters. The variable selection procedures are implemented in the R package `bartMachine` (Kapelner and Bleich, 2013).
2. **Stepwise Regression:** Backward stepwise regression using the `stepAIC` function in R.<sup>2</sup>
3. **Lasso Regression:** Regression with a Lasso (L1) penalty can be used for variable selection by selecting the subset of variables with nonzero coefficient estimates. For this procedure, an additional penalty parameter  $\lambda$  must be specified which controls the amount of shrinkage towards zero in the coefficients. We use the `glmnet` package in R (Friedman et al., 2010), which uses ten-fold cross-validation to select the value of the penalty parameter  $\lambda$ .
4. **Random Forests (RF):** Similarly to BART, RF must be adapted to the task of

---

<sup>2</sup>We also considered forward stepwise regression but found that backward stepwise regression gave uniformly better performance in our simulated data settings.

variable selection.<sup>3</sup> The `randomForest` package in R (Liaw and Wiener, 2002) produces an “importance score” for each predictor variable: the reduction in out-of-bag mean square error when that predictor is not allowed to contribute to the model. Breiman and Cutler (2013) suggest selecting only variables where the importance score exceeds the  $1 - \alpha$  quantile of a standard normal distribution. We follow their approach and further suggest a new approach: using the Bonferroni-corrected  $(1 - \alpha)/p$  quantile of a standard normal distribution. We employ a five-fold cross-validation approach to pick the best of these two thresholding strategies in each simulated data setting and let  $\alpha = .05$ . Default parameter settings for RF are used.

5. **Dynamic Trees (DT):** Gramacy et al. (2013) introduces a backwards variable selection procedure for DT. For each predictor, the authors compute the average reduction in posterior predictive uncertainty across all nodes using the given predictor as a splitting variable. The authors then propose a relevance probability, which is the proportion of posterior samples in which the reduction in predictive uncertainty is positive. Variables are deselected if their relevance probability does not exceed a certain threshold. After removing variables, the procedure is repeated until the log-Bayes factor of the larger model over the smaller model is positive, suggesting a preference for the larger model. We construct DT using the R package `dynaTree` with 5000 particles and a constant leaf model. We employ the default relevance threshold suggested by the authors of .50.

6. **Spike-and-Slab Regression (Spike-slab):** We employ the spike-and-slab regression procedure outlined in Ishwaran and Rao (2005) and Ishwaran and Rao (2010). The procedure first fits a spike-and-slab regression model and then per-

---

<sup>3</sup>Existing variable selection implementations for RF from Díaz-Uriarte and Alvarez de Andrés (2006) and Deng and Runger (2012) are not implemented for regression problems to the best of our knowledge.

forms variable selection via the generalized elastic net. Variables with non-zero coefficients are considered relevant. The method is applicable to both high and low dimensional problems, as in the high dimensional setting, a filtering of the variables is first performed for dimension reduction. The procedure is implemented in the R package `spikeslab` (Ishwaran et al., 2013).

Each of the above methods will be compared on the ability to select “useful” predictor variables, the subset of predictor variables that truly affect the response variable. We can quantify this performance by tabulating the number of true positive (TP) selections, false positive (FP) selections, true negative (TN) selections and false negative (FN) selections. The *precision* of a variable selection method is the proportion of truly useful variables among all predictor variables that are selected,

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \quad (9.2)$$

The *recall* of a variable selection method is the proportion of truly useful variables selected among all truly useful predictor variables,

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (9.3)$$

We can combine the precision and recall together into a single performance criterion,

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \quad (9.4)$$

which is the harmonic mean of precision and recall, balancing a procedure’s capability to make necessary identifications with its ability to avoid including irrelevant predictors. This  $F_1$  *measure* is called the “effectiveness” by van Rijsbergen (1979) and is used routinely in information retrieval and categorization problems.

While many variable selection simulations found in the literature rely on out-of-sample root mean square error (RMSE) to assess performance of a procedure, we believe the  $F_1$  score is a better alternative. Out-of-sample RMSE inherently overweights recall vis-à-vis precision since predictive performance depends more heavily on including covariates which generate signal. This is especially true for adaptive learning algorithms.

We chose the balanced<sup>4</sup>  $F_1$  metric because we want to demonstrate flexible performance while balancing both objectives, recall and precision. For example, if an investigator is searching for harmful physiological agents that can affect health outcomes, identifying all of the agents is most important (recall). If the investigator is looking to fund new, potentially expensive research based on discoveries (as in our application in Section 9.5) avoiding fruitless directions is most important (precision).

#### 9.4.1 Simulation Setting 1: Linear Relationship

We first examine the performance of the various variable selection approaches in a situation where the response variable is a linear function of the predictor variables. Specifically, we generate each predictor vector  $\mathbf{x}_j$  from a normal distribution

$$\mathbf{x}_1, \dots, \mathbf{x}_p \stackrel{iid}{\sim} \mathcal{N}_n(\mathbf{0}, \mathbf{I}), \quad (9.5)$$

and then the response variable  $\mathbf{y}$  is generated as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}_n(\mathbf{0}, \sigma^2 \mathbf{I}), \quad (9.6)$$

where  $\boldsymbol{\beta} = [\mathbf{1}_{p_0}, \mathbf{0}_{p-p_0}]^\top$ . In other words, there are  $p_0$  predictor variables that are truly related to the response  $\mathbf{y}$ , and  $p - p_0$  predictor variables that are spurious. The *sparsity* of a particular data setting is reflected in the proportion  $p_0/p$  of predictor

---

<sup>4</sup>The  $F_1$  measure can be generalized with different weights on precision and recall.

variables that actually influence the response.

Fifty datasets were generated for each possible combination of the following different parameter settings:  $p \in \{20, 100, 200, 500, 1000\}$ ,  $p_0/p \in \{0.01, 0.05, 0.1, 0.2\}$  and  $\sigma^2 \in \{1, 5, 20\}$ . In each of the 60 possible settings, the sample size was fixed at  $n = 250$ .

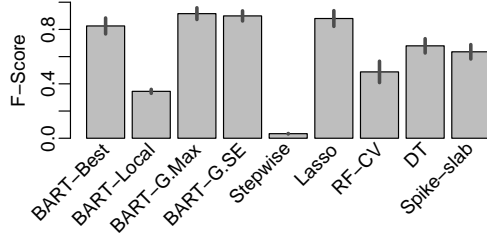
Figure 9.4 gives the  $F_1$  performance measure for each variable selection method for 8 of the 60  $p$ ,  $p_0/p$ ,  $\sigma^2$  simulation settings. We have chosen to illustrate these simulation results as they are representative of our overall findings. Here, higher values of  $F_1$  indicate better performance. Complete tables of precision, recall, and F1 measure values for the simulations shown in Figure 9.4 can be found in the online supplementary materials.<sup>5</sup>

We first focus on the comparisons in performance between the four thresholding strategies for our BART-based variable selection procedure: our three thresholding strategies plus the BART-Best cross-validated threshold strategy. First, we consider the case where  $p = 200$ . In the sparser settings (Figures 9.4a and 9.4b), the more stringent global-max and global-SE strategies perform better than the less stringent local thresholding strategy. However, the local thresholding strategy performs better in the less sparse settings (Figures 9.4c and 9.4d). The BART-Best procedure with a cross-validated threshold performs slightly worse than the best of the three thresholds in each setting, but fairs quite well uniformly. Hence, the cross-validated threshold strategy represents a good choice when the level of sparsity is not known *a priori*.

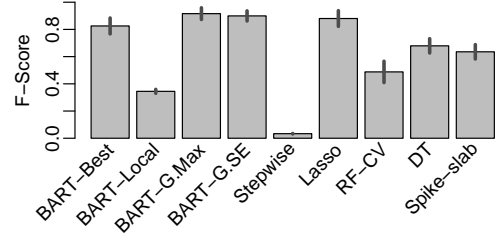
For the settings where  $p = 500$ , the findings are relatively similar. The local thresholding strategy performs well given the fact that the data is less sparse. Performance also degrades when moving from the low noise settings (Figure 9.4e and 9.4g) to the high noise settings (Figure 9.4f and 9.4h). Note that BART-Best does not

---

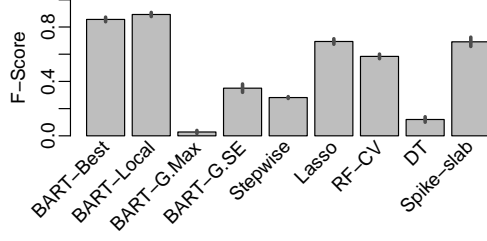
<sup>5</sup>Supplementary materials and source code can be found at [https://github.com/jbleich89/BART\\_variable\\_selection](https://github.com/jbleich89/BART_variable_selection).



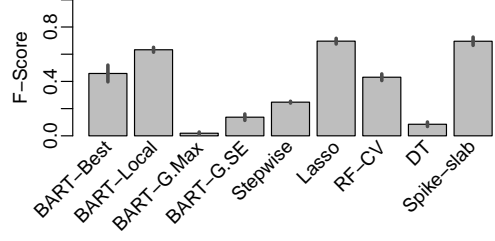
(a)  $p = 200, p_0 = 2, \sigma^2 = 5$



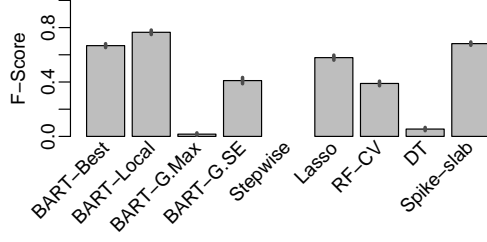
(b)  $p = 200, p_0 = 2, \sigma^2 = 20$



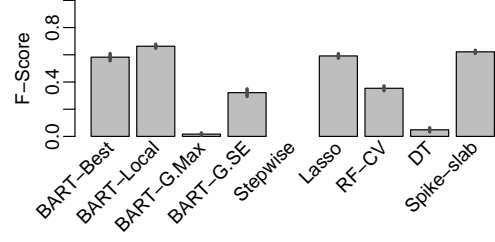
(c)  $p = 200, p_0 = 20, \sigma^2 = 5$



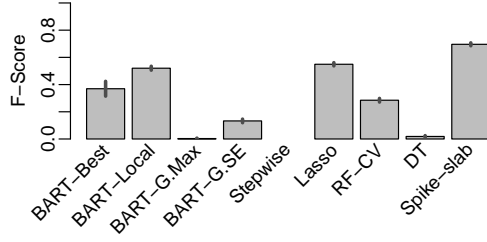
(d)  $p = 200, p_0 = 20, \sigma^2 = 20$



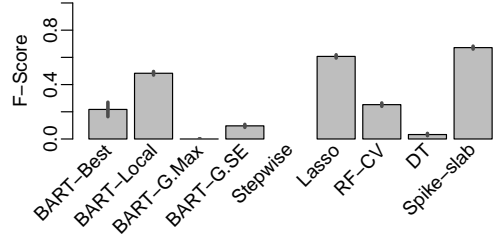
(e)  $p = 500, p_0 = 25, \sigma^2 = 1$



(f)  $p = 500, p_0 = 25, \sigma^2 = 5$



(g)  $p = 500, p_0 = 50, \sigma^2 = 1$



(h)  $p = 500, p_0 = 50, \sigma^2 = 5$

Figure 9.4: Average  $F_1$  measures for different variable selection approaches on simulated data under the linear model setting across 50 simulations. The black bars represent 90% error bars for the average. Results for  $p = 200$  and  $p = 500$  are shown. Within each choice of  $p$ , moving down a column shifts from high to low sparsity and moving across a row shifts from low to high noise.

perform particularly well in Figure 9.4h.

Comparing with the alternative approaches when  $p = 200$ , we see that **BART-Best** performs better than all of the alternatives in the lower noise, less sparse setting (Figures 9.4c and 9.4d) and is competitive with the lasso in the lower noise, more sparse setting (Figure 9.4c). **BART-Best** is competitive with the lasso in the higher noise, more sparse setting (Figure 9.4b) and beaten by the linear methods in the higher noise, less sparse setting (Figure 9.4d). When  $p = 500$ , the cross-validated **BART** is competitive with the lasso and **Spike-slab** and outperforms the nonlinear methods when  $p_0 = 25$  (Figure 9.4e and 9.4f). When  $p_0 = 50$  (Figure 9.4g and 9.4h), the cross-validated **BART** performs worse than the lasso and **Spike-slab** and has performance on par with the cross-validated **RF**.

Overall, the competitive performance of our **BART**-based approach is especially impressive since **BART** does not assume a linear relationship between the response and predictor variables. One would expect that stepwise regression, lasso regression, and **Spike-slab** would have an advantage since both methods assume a linear model which matches the data generating process in this setting. Like **BART**, **RF** and **DT** also do not assume a linear model, but in most of the cases we examined, our **BART**-based variable selection procedure performs better than **RF** and **DT**. We note that **DT** does not perform well on this simulation, possibly suggesting the need for a cross-validation procedure to choose appropriate relevance thresholds in different data settings.

Additionally, we briefly address the computation runtime of our four proposed approaches here by giving an estimate of the runtimes. For this data,  $n = 250$  and  $p = 200$ , the three strategies (local, global max and global SE) are estimated together in one **bartMachine** function in about 90 seconds. The cross-validated **BART-Best** procedure takes about 7 minutes.

### 9.4.2 Simulation Setting 2: Nonlinear Relationship

We next examine the performance of the variable selection methods in a situation where the response variable is a nonlinear function of the predictor variables. Specifically, we generate each predictor vector  $\mathbf{x}_j$  from a uniform distribution,

$$\mathbf{x}_1, \dots, \mathbf{x}_p \stackrel{iid}{\sim} U_n(0, 1),$$

and then the response variable  $\mathbf{y}$  is generated as

$$\mathbf{y} = 10\sin(\pi\mathbf{x}_1\mathbf{x}_2) + 20(\mathbf{x}_3 - .5)^2 + 10\mathbf{x}_4 + 5\mathbf{x}_5 + \boldsymbol{\mathcal{E}}, \quad \boldsymbol{\mathcal{E}} \sim \mathcal{N}_n(\mathbf{0}, \sigma^2\mathbf{I}). \quad (9.7)$$

This nonlinear function from Friedman (1991), used to showcase BART in Chipman et al. (2010), is a challenge for variable selection models due to its interactions and nonlinearities. In this data setting, only the first five predictors truly influence the response, while any additional predictor variables are spurious.

Fifty datasets were generated for each possible combination of  $\sigma^2 \in \{5, 100, 625\}$  and  $p \in \{25, 100, 200, 500, 1000\}$ . Since the number of relevant predictor variables is fixed at five, we simulate over a wide range of sparsity values, from  $p_0/p = 0.2$  down to  $p_0/p = 0.005$ . In each dataset, the sample size was fixed at  $n = 250$ .

Figure 9.5 illustrates the  $F_1$  performance measure for each variable selection method for four of the  $p, \sigma^2$  simulation pairs. We have chosen to illustrate these simulation results as they are representative of our overall findings. Backward stepwise regression via `stepAIC` could not be run in these settings where  $n < p$  and is excluded from these comparisons (values in Figure 9.5 for this procedure are set to 0). Complete tables of precision, recall, and F1 measure values for the simulations shown in Figure 9.5 are given in our online supplementary materials.



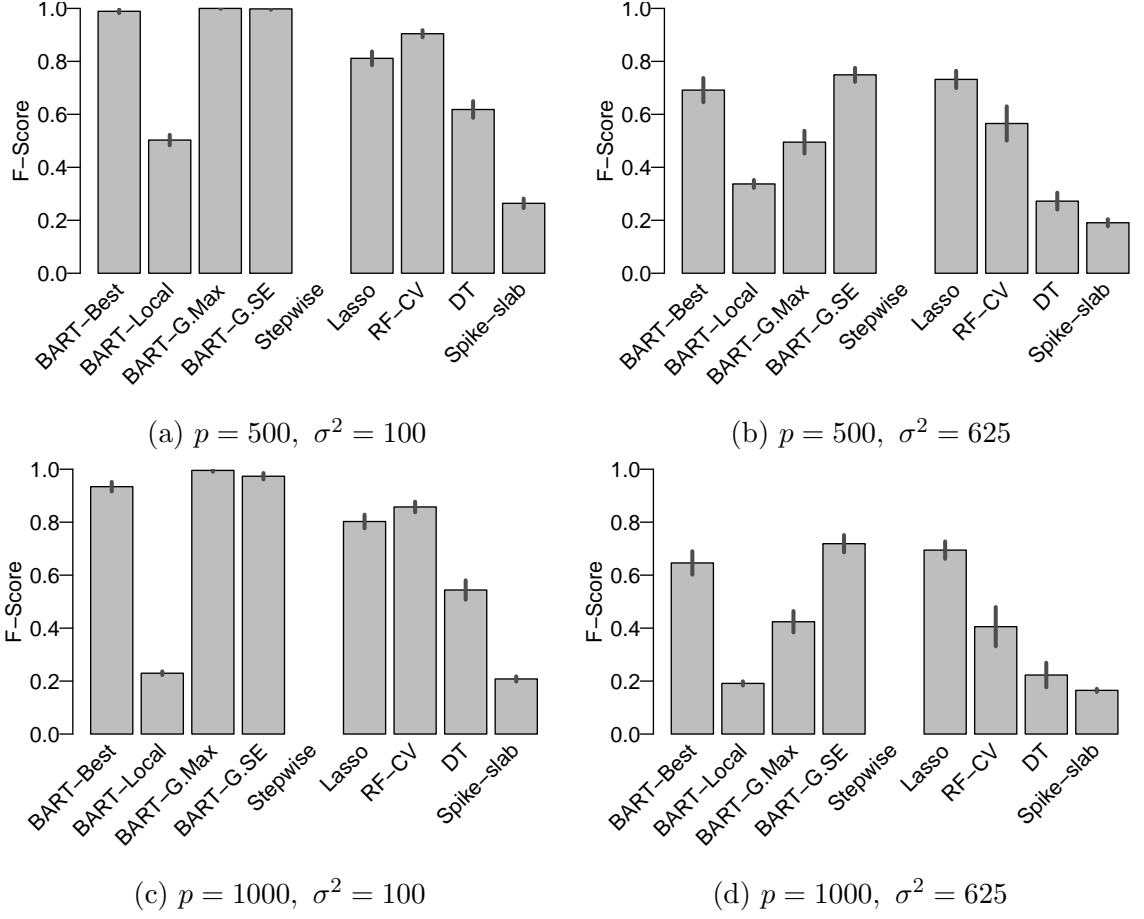


Figure 9.5: Average  $F_1$  measures across 50 simulations for different variable selection approaches on simulated data under the Friedman model setting. The black bars represent 90% error bars for the average. Moving from the top row to the bottom shifts from low to high dimensionality and moving from the left column to the right shifts from low to high noise.

Just comparing the four thresholding strategies of our **BART**-based procedure, we see that the more stringent selection criteria have better  $F_1$  performance measures in all of these sparse cases. The cross-validated threshold version of our **BART** procedure performs about as well as the best individual threshold in each case.

Compared to the other variable selection procedures, the cross-validated threshold version of our **BART** procedure has the best overall performance. Our cross-

validated procedure outperforms DT and RF-CV in all situations. The assumption of linearity puts the Lasso and **Spike-slab** at a disadvantage in this nonlinear setting. **Spike-slab** does not perform well on this data, although lasso performs well.<sup>6</sup> The cross-validated BART and the cross-validated RF have the best performance in the low noise settings (Figures 9.5a and 9.5c), as they do not assume linearity. Moving to the high noise settings (Figure 9.5d), BART and RF both see a degradation in performance, and cross-validated BART and the lasso are the best performers, followed by the cross-validated RF.

### 9.4.3 Simulation Setting 3: Linear Model with Informed Priors

In the next set of simulations, we explore the impact of incorporating informed priors into the BART model, as discussed in Section 9.3.5. We will evaluate the performance of our BART-based variable selection procedure in cases where the prior information is correctly specified as well as in cases where the prior information is incorrectly specified.

We will use the linear model in Section 9.4.1 as our data-generating process. We will consider a specific case of the scheme outlined in Section 9.3.5 where particular subsets of predictor variables are given twice as much weight as the rest of the predictor variables. With a noninformative prior, each predictor variable has a probability of  $1/p$  of being selected as a splitting rule. In the informed prior, a subset of  $p_0$  predictor variables are given twice as much weight, which gives those variables a larger probability of  $2/(p + p_0)$  of being selected as a splitting rule.

---

<sup>6</sup>We note that lasso's performance here is unexpectedly high. For this example, lasso is able to recover the predictors that are interacted within the sine function. This seems to be an artifact of this particular data generating process, and we would expect lasso to perform worse on other nonlinear response functions.

For the fifty datasets generated under each combination of the parameter settings in the simulations of Section 9.4.1, we implemented three different versions of **BART**: (1) **BART** with noninformative priors on predictor variables, (2) **BART** with “correct” informed priors (twice the weight on the subset of predictor variables that have a true effect on response) and (3) **BART** with “incorrect” informed priors (twice the weight on a random subset of spurious predictor variables). For each of these **BART** models, predictor variables were then selected using the cross-validated threshold strategy.

Figure 9.6 gives the  $F_1$  measures for the three different **BART** priors in four of the data settings outlined in Section 9.4.1.

There are two key observations from the results in Figure 9.6. First, correct prior information can substantially benefit the variable selection ability of our **BART** adaptation, especially in higher noise settings (Figures 9.6b and 9.6d). Second, the incorrect prior information does not degrade performance in any of the cases, which suggests that our **BART**-based variable selection procedure is robust to the misspecification of an informed prior on the predictor variables. This seems to be a consequence of the Metropolis-Hastings step, which tends to not accept splitting rules that substantially reduce model likelihood, regardless of how often they are proposed.

To summarize our simulation studies in Section 13.3, our **BART**-based variable selection procedure is competitive with alternative approaches when there is linear relationship between the predictor variables and the response, and performs better than alternative approaches in a nonlinear data setting. **BART**-based variable selection can be further improved by correctly specifying prior information (when available) that gives preference to particular predictor variables, and appears to be robust to mis-specification of this prior information.

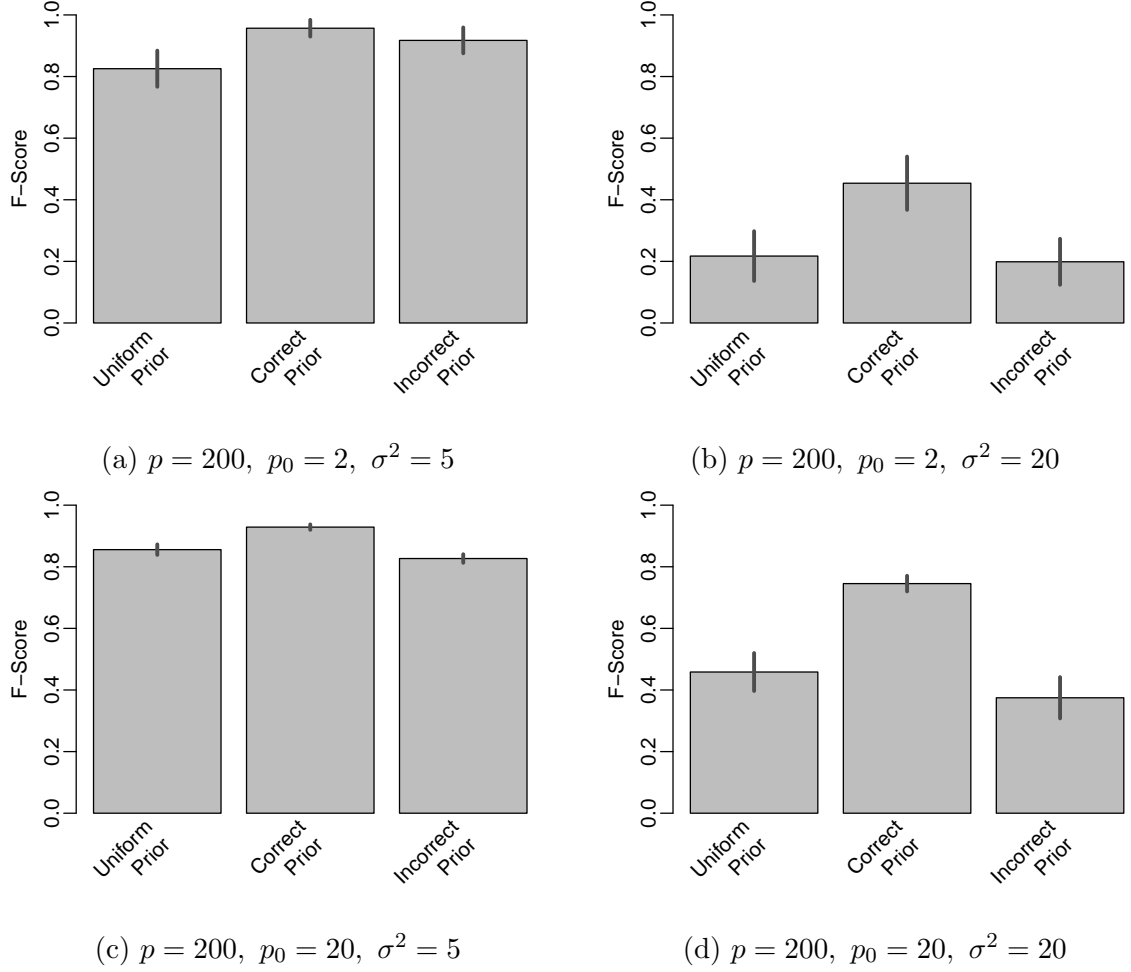


Figure 9.6: Average  $F_1$  measures across 50 simulations for BART-based variable selection under three different prior choices. The black bars represent 90% error bars for the average. The settings shown are the same as those in Figures 9.4a-9.4d.

## 9.5 Application to Gene Regulation in Yeast

Experimental advances in molecular biology have led to the availability of high-dimensional genomic data in a variety of biological applications. We will apply our BART-based variable selection methodology to infer the gene regulatory network in budding yeast (*Saccharomyces cerevisiae*). One of the primary mechanisms by which genes are regulated is through the action of transcription factors, which are proteins

that increase or decrease the expression of a specific set of genes.

The data for our analyses are expression measures for 6026 genes in yeast across 314 experiments. For those same 314 experiments, we also have expression measures for 39 known transcription factors. For each of the 6026 genes, our goal is to identify the subset of the 39 transcription factors that have a real regulatory relationship with that particular gene.

We consider each of the 6026 genes as a separate variable selection problem. For a particular gene  $g$ , we model the expression measures for that gene as a  $314 \times 1$  response vector  $\mathbf{y}_g$  and we have 39 predictor variables  $(\mathbf{x}_1, \dots, \mathbf{x}_{39})$  which are the expression measures of each of the 39 transcription factors. This same data was previously analyzed using a linear regression approach in Jensen et al. (2007), but we will avoid assumptions of linearity by employing our BART-based variable selection model.

We also have additional data available for this problem that can be used as prior information on our predictor variables. Lee et al. (2002) performed chromatin immunoprecipitation (ChIP) experiments for each of the 39 transcription factors that we are using as predictor variables. The outcome of these experiments is the estimated probabilities  $m_{gk}$  that gene  $g$  is physically bound by each transcription factor  $k$ . Chen et al. (2007) give details of how these probabilities  $m_{gk}$  are derived from the ChIP data.<sup>7</sup>

We will incorporate these estimated probabilities into our BART-based variable selection approach as prior information. When selecting predictor variables for splitting rules, we give more weight to the transcription factors  $k$  with larger prior probabilities  $m_{gk}$  in the BART model for gene  $g$ . Specifically, we have a splitting variable weight  $w_{gk}$  for predictor  $\mathbf{x}_k$  in the BART model for gene  $g$ , which we calculate as

---

<sup>7</sup>Probabilities were truncated to be between 5% and 95%.

$$w_{gk} = 1 + c \cdot m_{gk}. \quad (9.8)$$

In the **BART** model for gene  $g$ , each predictor  $\mathbf{x}_k$  is chosen for a splitting rule with probability proportional to  $w_{gk}$ . The global parameter  $c$  controls how influential the informed prior probabilities  $m_{gk}$  are on the splitting rules in **BART**. Setting  $c = 0$  reduces our informed prior to the uniform splitting rules of the standard **BART** implementation. Larger values of  $c$  increases the weights of predictor variables with large prior probabilities  $m_{gk}$ , giving the informed prior extra influence.

In a real data setting such as our yeast application, it is difficult to know how much influence to give our informed priors on the predictor variables. We will consider several different values of  $c = \{0, 1, 2, 4, 10000\}$  and choose the value that results in the smallest prediction error on a subset of the observed data that is held-out from our **BART** model estimation. Specifically, recall that we have 314 expression measures for each gene in our dataset. For each gene, we randomly partition these observations into an 80% training set, 10% tuning set and a 10% hold-out set. For each value of  $c = \{0, 1, 2, 4, 10000\}$ , we fit a **BART** model on the 80% training set and then choose the value of  $c$  that gives the smallest prediction error on the 10% tuning set. This same 10% tuning set is also used to choose the best threshold procedure among the three options outlined in Section 9.3.4. We will use the terminology “**BART-Best**” to refer to the **BART**-based variable selection procedure that is validated over the choice of  $c$  and the three threshold strategies. While we could also cross-validate over the significance level  $\alpha$ , we fix  $\alpha = .05$  due to computational concerns given the large number of data sets to be analyzed.

For each gene, we evaluate our approach by re-fitting **BART** using only the variables selected by our **BART**-based variable selection model and evaluate the prediction accuracy on the final 10% hold-out set of data for that gene. This same 10% hold-out set of data for each gene is also used to evaluate the prediction accuracy of various

alternative variable selection methods. Just as in Section 13.3, we consider the alternative methods of stepwise regression, lasso regression and RF, DT, and **Spike-slab** in similar fashions to Section 13.3. The 10% tuning set is used to choose the value of the penalty parameter  $\lambda$  for lasso regression as well as the importance score threshold for RF. For DT, we use a constant leaf model for variable selection and then construct a linear leaf model using the selected variables for prediction.

We also consider three simpler approaches that do not select particular predictor variables: (1) “**BART-Full**” which is the **BART** model using all variables, (2) ordinary least squares regression (OLS) with all predictor variables included and (3) the “null” model: the sample average of the response which does not make use of any predictors. In the null model, we do include an intercept so we are predicting the hold-out set of expression measures for each gene with the average expression level of that gene in the training set.

We first examined the distribution of RMSEs across the 6026 genes. We found that each procedure improves over the null model with no covariates, suggesting that some subset of transcription factors is predictive of gene expression for most of the 6026 genes. However, for a minority of genes, the null model is competitive which suggests that the 39 available transcription factors may not be biologically relevant to every one of these genes. The non-null variable selection methods show generally similar performance in terms of the distribution of RMSEs, and a corresponding figure can be found in the online supplementary materials. It is important to note that predictive accuracy in the form of out-of-sample RMSE is not the most desirable metric for comparing variable selection techniques because it overweights recall relative to precision.

In Figure 9.7, we show the distribution of the number of selected predictor variables across the 6026 genes, where we see substantial differences between the variable selection procedures. Figure 9.7 confirms that **BART-G.max** is selecting very few pre-

dictor variables for each gene. Even more interesting is the comparison of **BART-Best** to stepwise regression, lasso regression, **RF**, and **Spike-slab**. **BART-Best** is selecting far fewer predictor variables than these alternative procedures. Interestingly, **DT**, the other Bayesian tree-based algorithm, a number of variables most similar to **BART-Best**.

Given the relatively similar performance of methods in terms of RMSE and the more substantial differences in number of variables selected, we propose the following combined measure of performance for each variable selection method:

$$(\text{RMSE reduction per predictor})_{\text{method}} = \frac{\text{RMSE}_{\text{null}} - \text{RMSE}_{\text{method}}}{\text{NumPred}_{\text{method}}},$$

where  $\text{RMSE}_{\text{method}}$  and  $\text{NumPred}_{\text{method}}$  are respectively the out-of-sample RMSE and number of predictors selected for a particular method. This performance metric answers the question: how much “gain” are we getting for adding each predictor variable suggested by a variable selection approach? Methods that give larger RMSE reduction per predictor variable are preferred.

Figure 9.8 gives the RMSE reduction per predictor for each of our variable selection procedures. Note that we must restrict ourselves to plotting only cases where at least one predictor variable is selected, since RMSE reduction per predictor is only defined if the number of predictors selected is greater than zero.

Our **BART-Best** variable selection procedure gives generally larger (better) values of the RMSE reduction per predictor measure than stepwise regression, lasso regression, **RF**, and **Spike-slab**. **DT** is the closer competitor, but does slightly worse, on average, than **BART-Best**. Also, both **BART-Full** and **OLS** procedures, where no variable selection is performed, do even worse than the other variable selection procedures.

**BART-G.max**, the **BART**-based procedure under the global max threshold, seems to perform even better than the **BART-Best** procedure in terms of the RMSE reduction



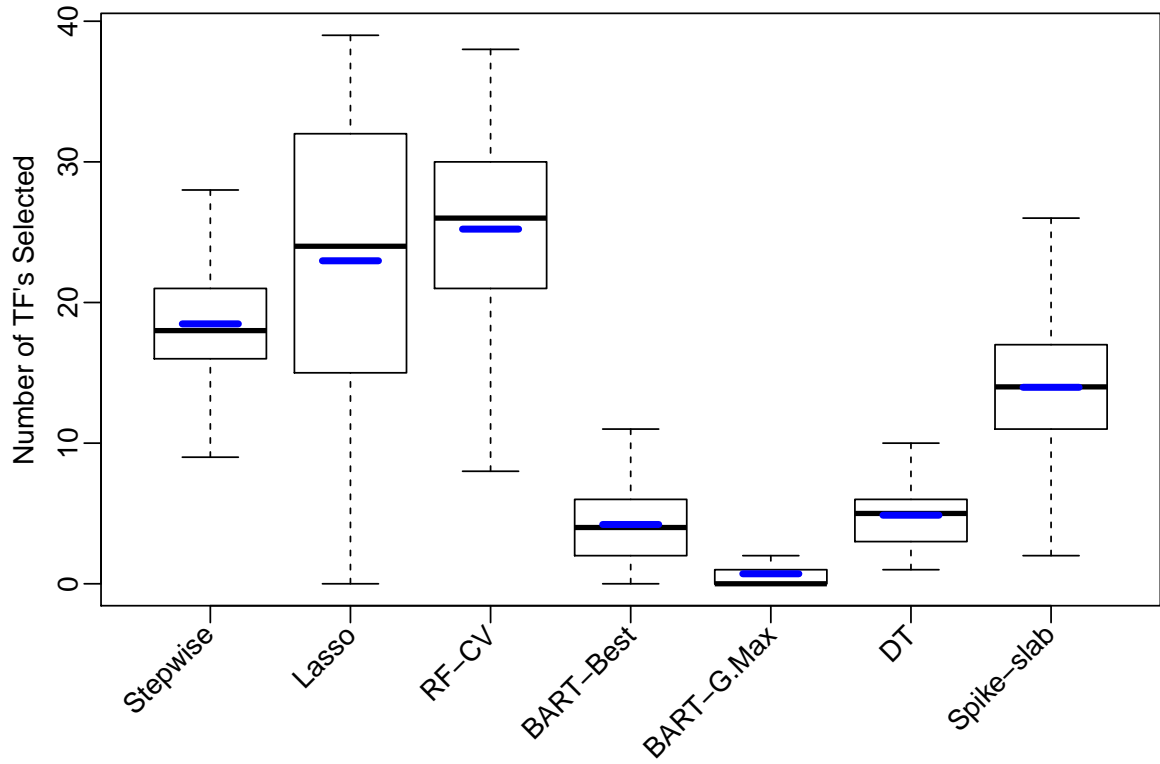


Figure 9.7: Distributions of the number of predictor variables selected for each method across all 6026 genes. Blue bars represent the average number of selected predictor variables for each method. Not shown are the null model which uses no predictors as well as OLS and the full BART model which both use all predictors. Points beyond the whiskers are omitted.

per predictor measure. However, recall that we are plotting only cases where at least one predictor was selected. **BART-G.max** selects at least one transcription factor for only 2866 of the 6026 genes, though it shows the best RMSE reduction per predictor in these cases. By comparison, **BART-Best** selects at least one transcription for 5459 of the 6026 genes while showing better RMSE reduction per predictor than the non-BART variable selection procedures.

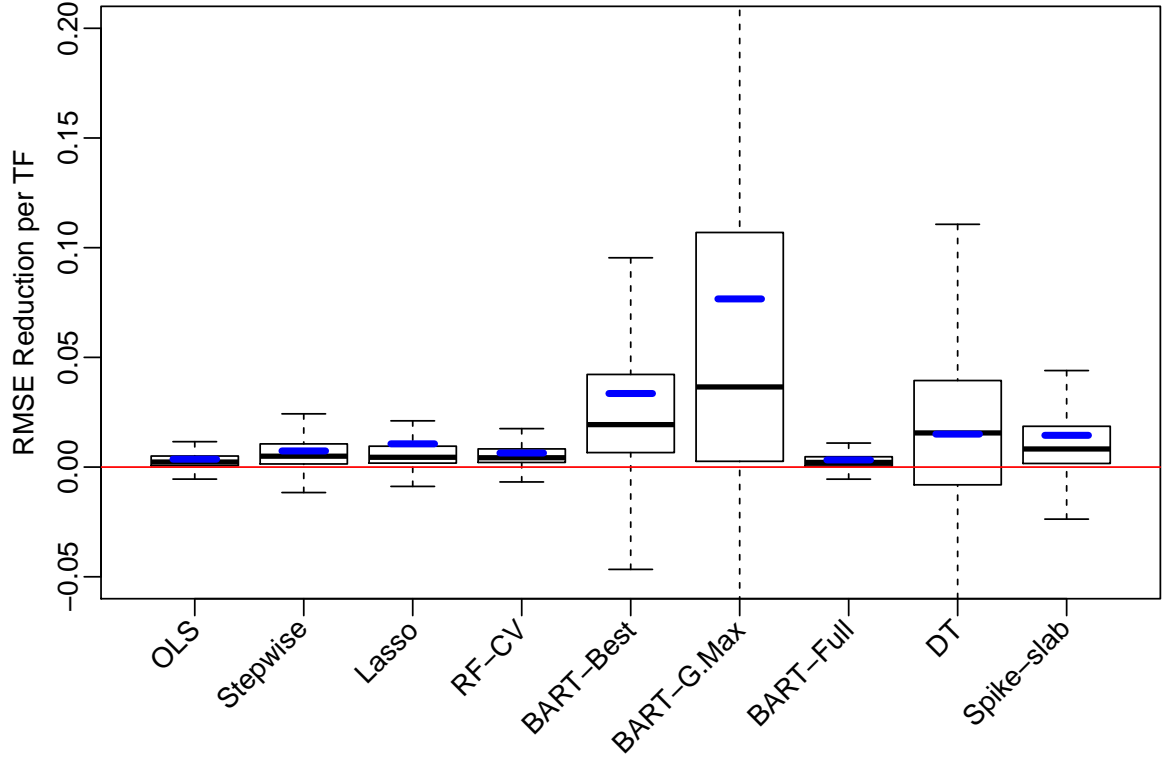


Figure 9.8: Distributions of the RMSE reduction per predictor for each method across all 6026 genes. Blue bars represent the average RMSE reduction per predictor across the 6026 genes for each method. Points beyond the whiskers are omitted.

Additionally, Table 9.2 shows the proportion of times each choice of prior influence  $c$  appeared in the “BART-Best” model. Almost a quarter of the time, the prior information was not used. However, there is also a large number of genes for which the prior was considered to have useful information and was incorporated into the procedure.

Jensen et al. (2007) also used the same gene expression data (and ChIP-based prior information) to infer gene-TF regulatory relationships. A direct model comparison between our BART-based procedures and their approach is difficult since Jensen et al. (2007) fit a simultaneous model across all genes whereas our current BART-based analysis fits a predictive model for each gene separately. In both analyses, prior information for each gene-TF pairing from ChIP binding data (Lee et al., 2002) was

$c$ Value	Percentage of Genes
0	23.3%
0.5	16.1%
1	15.4%
2	14.9%
4	14.6%
10000	15.7%

Table 9.2: Distribution of prior influence values  $c$  used across the 6026 genes.

used<sup>8</sup>. However, in Jensen et al. (2007) the prior information for a particular TF was given the same weight (relative to the likelihood) for each gene in the dataset. In our BART-based analysis, each gene was analyzed separately and so the prior information for a particular TF can be weighted differently for each gene.

A result of this modeling difference is that the prior information appears to have been given less weight by our BART-based procedure across genes, as evidenced by the substantial proportion of genes in Table 9.2 that were given zero or low weight ( $c = 0$  or  $c = 0.5$ ). Since that prior information played the role in Jensen et al. (2007) of promoting sparsity, a consequence of that prior information being given less weight in our BART-based analysis is reduced promotion of sparsity.

This consequence is evident in Figure 9.9, where we compare the number of selected predictor variables. The x-axis gives the 39 transcription factors that served as the predictor variables for each of our 6026 genes. The y-axis is the number of genes for which that TF was selected as a predictor variable by each of three procedures: BART-Best, BART-G.max and the analysis of Jensen et al. (2007). The most striking feature of Figure 9.9 is that each TF was selected for many more genes under our BART-Best procedure compared to BART-G.max, which also selected more variables than the analysis of Jensen et al. (2007). This result indicates that selecting more TFs per gene leads to the best out-of-sample predictive performance (i.e. BART-Best).

---

<sup>8</sup>Jensen et al. (2007) used additional prior information based on promoter sequence data that we did not use in our BART-based analysis.

It could be that Jensen et al. (2007) was over-enforcing sparsity, but that previous method also differed from our current approach in terms of assuming a linear relationship between response and predictor variables.

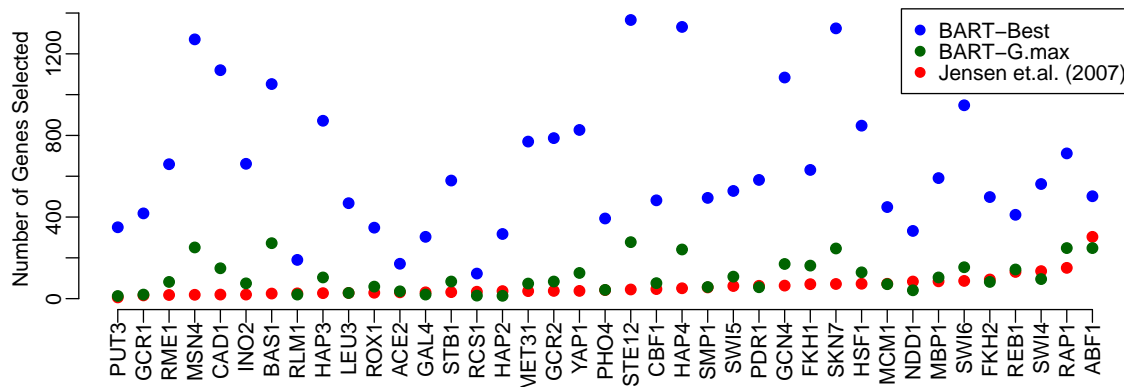


Figure 9.9: Number of genes for which each TF was selected. Results are compared for BART-Best, BART-G.Max, and the linear hierarchical model developed in Jensen et al. (2007).

## 9.6 Conclusion

Chipman et al. (2010)’s Bayesian additive regression trees is a rich and flexible model for estimating complicated relationships between a response variable and a potentially large set of predictor variables. We adapt **BART** to the task of variable selection by employing a permutation procedure to establish a null distribution for the variable inclusion proportion of each predictor. We present several thresholding strategies that reflect different beliefs about the degree of sparsity amongst the predictor variables, as well as a cross-validation procedure for choosing the best threshold when the degree of sparsity is not known *a priori*.

In contrast with popular variable selection methods such as step-wise regression and lasso regression, our **BART**-based approach does not make strong assumptions of

linearity in the relationship between the response and predictors. We also provide a principled means to incorporate prior information about the relative importance of different predictor variables into our **BART**-based variable selection procedure.

We used several simulated data settings to compare our **BART**-based approach to alternative variable selection methods such as stepwise regression, lasso regression, random forests, and dynamic trees. Our variable selection procedure is competitive with these alternatives in the setting where there is a linear relationship between response and predictors, and performs better than these alternatives in a nonlinear setting. Additional simulation studies suggest that our **BART**-based procedure can be further improved by correctly specifying prior information (if such information is available) and seems to be robust when the prior information is incorrectly specified.

We applied our **BART**-based variable selection procedure, as well as alternative methods, to the task of selecting a subset of transcription factors that are relevant to the expression of individual genes in yeast (*Saccharomyces cerevisiae*). In this application, our **BART**-based variable selection procedure generally selected fewer predictor variables while achieving similar out-of-sample RMSE compared to the lasso and random forests. We combined these two observations into a single performance measure, RMSE reduction per predictor. In this application to inferring regulatory relationships in yeast, our **BART**-based variable selection demonstrates much better predictive performance than alternative methods such as lasso and random forests while selecting more predictor variables than the previous approach of Jensen et al. (2007).

While we found success using the variable inclusion proportions as the basis for our procedure, fruitful future work would be to explore the effect of a reduction in variance metric, such as that explored in Gramacy et al. (2013) within **BART**.

## Acknowledgments

We would like to thank Hugh MacMullan for help with grid computing. Adam Kapelner acknowledges the National Science Foundation for his graduate research fellowship.

## Online Supplement

The complete set of results for simulations in Section 13.3 and source code can be found at [https://github.com/jbleich89/BART\\_variable\\_selection](https://github.com/jbleich89/BART_variable_selection).

## BART with Parametric Models of Heteroskedasticity

By Justin Bleich and Adam Kapelner

### Abstract

We incorporate heteroskedasticity into Bayesian Additive Regression Trees (BART) by modeling the log of the error variance parameter as a linear function of prespecified covariates. Under this scheme, the Gibbs sampling procedure for the original sum-of-trees model is easily modified, and the parameters for the variance model are updated via a Metropolis-Hastings step. We demonstrate the promise of our approach by providing more appropriate posterior predictive intervals than homoskedastic BART in heteroskedastic settings and demonstrating the model's resistance to overfitting. Our implementation will be offered in an upcoming release of the R package `bartMachine`.

### 10.1 Introduction

We consider the the following general heteroskedastic regression framework to characterize the relationship between a continuous response vector  $\mathbf{y}$  and a set of  $p$  predictor variables  $\mathbf{X} := [\mathbf{x}_{\cdot 1}, \dots, \mathbf{x}_{\cdot p}]$  which can be continuous or categorical:

$$\mathbf{y} = f(\mathbf{X}) + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}_n(\mathbf{0}, \sigma^2 \mathbf{D})$$

$\mathbf{D}$  denotes the diagonal matrix whose entries are scaling factors for the error variance for each observation. In this model, the response is considered an unknown function  $f$  of the predictors and the observations, while independent, exhibit non-constant error variance. The goal of this article is to model the relationship between the predictors and response with the aim of generating accurate predictions. To this end we model  $f$  with Bayesian Additive Regression Trees (**BART**, Chipman et al., 2010) which is a sum-of-trees model that has demonstrated predictive performance competitive with the best statistical learning algorithms. The original **BART** model is constrained to homoskedastic error variance ( $\mathbf{D} = \mathbf{I}$ ). Here, we extend the model to flexibly handle an error variance structure which is a linear model of prespecified covariates and we name our procedure “heteroskedastic **BART**” or “**HBART**.” Similar to Huber-White sandwich estimation (White, 1980), appropriately modeling the diagonal entries of  $\mathbf{D}$  “downweights” high variance observations. This allows for (a) a more accurate model as measured by predictive performance on future observations as well as (b) posterior credible and predictive intervals which appropriately reflect the changing heteroskedasticity in predictor space.

In Section 10.2, we provide an overview of the literature on heteroskedastic regression modeling in a Bayesian paradigm. In Section 10.3, we introduce **HBART**, highlighting the necessary modifications to the original homoskedastic **BART**. In Section 10.4, we provide simulations to showcase the desirable properties of **HBART**, including less overfitting for high noise observations as well as more appropriate uncertainty intervals for predictions in the presence of heteroskedasticity. Section 13.4 explores two applications to real data. We conclude and offer future research directions in Sec-



tion 13.5. The method developed in this paper will be implemented in an upcoming release of the R package `bartMachine` (Kapelner and Bleich, 2013), which is available on CRAN.

## 10.2 Bayesian Heteroskedastic Regression

Early approaches for heteroskedastic regression primarily focused on point estimation for the parameters governing the underlying heteroskedasticity of the model (for an overview, see Carroll and Ruppert, 1988). Potential problems with point estimation gave rise to a proposal of a fully Bayesian approach for heteroskedastic linear regression, where the non-constant variance depends on simple functions of an unknown parameter  $\theta$  and a set of weights  $w_i$  (Boscardin and Gelman, 1994).

Cepeda and Gamerman (2001) introduce a Bayesian regression model where the conditional mean of the response is modeled as a linear function of covariates  $\mathbf{x}_1, \dots, \mathbf{x}_p$  plus heteroskedastic noise. They model the variance for each observation as a monotonic differentiable function of a linear combination of another set of covariates,  $g(\mathbf{z}_1, \dots, \mathbf{z}_k)$ . Additionally, the function  $g$  is chosen to ensure positivity of the variance terms. The authors rely on a block Gibbs sampling approach (Geman and Geman, 1984), sampling the parameters for the mean function and variance function in two stages. In particular, the parameters for the variance function are updated via a Metropolis-Hastings step (Hastings, 1970) using the approach of Gamerman (1997), which relies on an iteratively reweighted least squares model to generate suitable proposal distributions.

More recent approaches have focused on relaxing the assumptions of linear additive components for modeling the mean and variance functions. Yau and Kohn (2003) propose nonparametric models for each of these two functions by employing penalized regression spline estimation for both models. Chan et al. (2006) extend this nonpara-

metric model to allow for semiparametric modeling of both the mean and variance functions, using radial basis functions for nonparametric components. Additionally, their approach can handle a large number of basis terms by introducing Bayesian variable selection priors, allowing model estimation to be locally adaptive. Leslie et al. (2007) relax the assumption of normal errors and developed a heteroskedastic linear regression model with general error distributions by relying on a Dirichlet process mixture prior.

Both Chan et al. (2006) and Leslie et al. (2007) rely on the sampling scheme developed in Gamerman (1997) to obtain draws from the posterior distribution of the parameters for the variance function. Our work similarly draws heavily on this technique.

### 10.3 Augmenting BART to Incorporate Heteroskedasticity

In its original formulation in Chipman et al. (2010), the authors assume that the response  $\mathbf{y}$  could be modeled as a sum-of-trees model of the covariates  $\mathbf{X} := [\mathbf{x}_1, \dots, \mathbf{x}_p]$  plus homoskedastic normal noise:

$$\mathbf{Y} = \sum_{i=1}^m \mathfrak{T}_i^{\varnothing}(\mathbf{X}) + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}_n(\mathbf{0}, \sigma^2 \mathbf{I}_n) \quad (10.1)$$

Each of the  $m$  distinct binary regression trees consists of a tree structure, denoted by  $\mathfrak{T}$ , and a set of parameters in the terminal nodes of the tree (also called “leaves”), denoted by  $\varnothing$ . A tree with both its structure and set of leaf parameters is denoted by  $\mathfrak{T}^{\varnothing}$ .

The structure of a given tree  $\mathfrak{T}_t$  includes information on how any observation recurses down the tree until a leaf node is reached. Each nonterminal, internal node

contains a “splitting rule”  $\mathbf{x}_j < c$ . When this condition is satisfied for a given observation, the observation moves to the left daughter node or moves to the right daughter node otherwise. Once an observation lands in a terminal node, the leaf value in that node is assigned as the predicted value for that observation.  $\mathcal{L}_t = \{\mu_{t,1}, \mu_{t,2}, \dots, \mu_{t,b_t}\}$  represents the set of leaf parameters for a given tree, where  $b_t$  is the number of terminal nodes in that tree.

Thus, **BART** estimates the mean function  $f$  using a sum-of-regression trees. Regression trees rely on recursive binary partitioning of predictor space into a menagerie of hyperrectangles in order to approximate the unknown function of interest. By employing a sum-of-trees approach, **BART** is able to take advantage of the regression trees’ ability to successfully model nonlinearities and interactions while also capturing additive components of the fit. As a Bayesian model, **BART** is composed of a set of priors and a likelihood. Using Gibbs sampling, posterior inference for the unknown  $f$  can be obtained.

We propose an extension to **BART** by allowing each  $\sigma_1^2, \dots, \sigma_n^2$  to be scaled by the exponential of a linear parametric function of  $k$  covariates  $\mathbf{Z} := [\mathbf{z}_{\cdot 1}, \dots, \mathbf{z}_{\cdot k}]$ , the “heteroskedasticity covariates” which are potentially distinct from the covariates used to model the mean function,  $\mathbf{x}_{\cdot 1}, \dots, \mathbf{x}_{\cdot p}$ . Our heteroskedastic model, **HBART**, is given as

$$\mathbf{y} = \sum_{i=1}^m \mathfrak{F}_i^{\mathcal{L}}(\mathbf{X}) + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}_n \left( \mathbf{0}, \sigma^2 \begin{bmatrix} \exp(\mathbf{z}_{1\cdot} \cdot \boldsymbol{\gamma}) & & 0 \\ & \ddots & \\ 0 & & \exp(\mathbf{z}_{n\cdot} \cdot \boldsymbol{\gamma}) \end{bmatrix} \right) \quad (10.2)$$

where  $\boldsymbol{\gamma} := [\gamma_1, \dots, \gamma_k]^\top$  is a column vector of linear coefficients for the  $k$  heteroskedasticity covariates. Thus, the variance of each observation is specified as a log-linear model:

$$\ln(\sigma_i^2) = \ln(\sigma^2) + \mathbf{z}_i \cdot \boldsymbol{\gamma} \quad \text{for } i = 1, \dots, n. \quad (10.3)$$

It is important to note that **BART**, by design, is an overparameterized model with “an abundance of unidentified parameters” (Chipman et al., 2010) allowing for a highly flexible fit. First, given the unidentifiable nature of the model, our focus is not on valid inference for  $\boldsymbol{\gamma}$ . Instead, we incorporate heteroskedasticity to aid in forecasting and generating posterior uncertainty intervals. Second, due to the already complex nature of the original **BART** algorithm, we employ parametric models for heteroskedasticity versus more sophisticated alternatives (such as the proposal of Chan et al., 2006) in order to prevent the model from becoming “too flexible.” Given highly flexible, unidentifiable estimation of both the mean and variance functions, the algorithm may have difficulty distinguishing between signal and noise, thereby shirking on its primary duty which is accurate estimation of the mean model. Hence, parametric models of heteroskedasticity represent the first step towards understanding how flexible **BART** can be in nonparametric function estimation when the homoskedasticity assumption is relaxed.

The remainder of the section is dedicated to describing the priors on **HBART** as well as the Gibbs sampling procedure for obtaining posterior inference.

### 10.3.1 Priors and Likelihood

The **BART** model requires three priors. The first prior is on the tree structures themselves and the second is on the leaf parameters. The third prior is on the error variance  $\sigma^2$ . **HBART** requires these same three priors as well as a prior on  $\boldsymbol{\gamma}$ . By assumption, the priors on  $\sigma^2$ ,  $\boldsymbol{\gamma}$ ,  $\mathfrak{F}$  and  $\mathfrak{L}$  are independent of one another:

$$\begin{aligned}
\mathbb{P}\left(\mathfrak{F}_1^{\text{leaf}}, \dots, \mathfrak{F}_m^{\text{leaf}}, \sigma^2, \gamma\right) &= \left[ \prod_t \mathbb{P}\left(\mathfrak{F}_t^{\text{leaf}}\right) \right] \mathbb{P}\left(\sigma^2\right) \mathbb{P}(\gamma) \\
&= \left[ \prod_t \mathbb{P}\left(\mathcal{E}_t \mid \mathfrak{F}_t\right) \mathbb{P}\left(\mathfrak{F}_t\right) \right] \mathbb{P}\left(\sigma^2\right) \mathbb{P}(\gamma) \\
&= \left[ \prod_t \prod_{\ell} \mathbb{P}\left(\mu_{t,\ell} \mid \mathfrak{F}_t\right) \mathbb{P}\left(\mathfrak{F}_t\right) \right] \mathbb{P}\left(\sigma^2\right) \mathbb{P}(\gamma)
\end{aligned}$$

where the last line follows from an assumption of conditional independence of the leaf parameters given the tree structure.

The prior on the tree structures and leaf parameters are designed to provide regularization to the model by respectively preventing the trees from growing too deep and shrinking the predicted values towards the range center of the response. These priors, as well as a prior on proposing splitting rules, are the same as those used in the original **BART** model and we refer the interested reader to Chipman et al. (2010).

For the homoskedastic implementation, the prior is on the error variance and is chosen to be  $\sigma^2 \sim \text{InvGamma}(\nu/2, \nu\lambda/2)$ .  $\lambda$  is determined from the data so that there is a  $q = 90\%$  a priori chance (by default) that the **BART** model will improve upon the root mean square error (RMSE) from an ordinary least squares regression (therefore, the majority of the prior probability mass lies below the RMSE of a least squares regression). Additionally, this prior limits the probability mass placed on small values of  $\sigma^2$  to prevent overfitting. We use this same data-informed prior for the  $\sigma^2$  parameter in **HBART**, the logarithm of which serves as the intercept term in the log-linear model for the variances (Equation 10.3).

Following the approach of Gamerman (1997), we place a multivariate normal prior on  $\gamma$  given as:

$$\boldsymbol{\gamma} \sim \mathcal{N}_k \left( \boldsymbol{\gamma}_0, \begin{bmatrix} \Sigma_{11} & & 0 \\ & \ddots & \\ 0 & & \Sigma_{kk} \end{bmatrix} \right) \quad (10.4)$$

We make the simplifying assumption that each component of the prior is independent of one another, but this can be easily generalized.

Along with a set of priors, **HBART** (and **BART**) consists of the likelihood of responses in the leaf nodes. The likelihood is assumed to be normal with the mean being the “best guess” of the leaf parameters at the current moment and variance being the best guess of the variance at the moment i.e.  $\mathbf{y}_\ell \sim \mathcal{N}(\mu_\ell, \sigma_i^2/m)$ . These “best guesses” are the values being conditioned on in the Gibbs sampler during each iteration.

### 10.3.2 Sampling from the Posterior

The Gibbs sampler can be used to obtain draws from  $\mathbb{P}(\mathfrak{T}_1^{\text{leaf}}, \dots, \mathfrak{T}_m^{\text{leaf}}, \sigma^2, \boldsymbol{\gamma} \mid \mathbf{y}, \mathbf{X})$ , the posterior distribution of the model parameters. As with the original sampling scheme for **BART**, **HBART** also relies on “Bayesian backfitting” (Hastie and Tibshirani, 2000) to fit each tree iteratively, holding all other  $m - 1$  trees constant. This is achieved by considering the residual response when updating the  $j^{\text{th}}$  tree  $\mathbf{R}_j := \mathbf{y} - \sum_{t \neq j} \mathfrak{T}_t^{\text{leaf}}(\mathbf{X})$ .

The Gibbs sampler for **HBART** proceeds by first proposing a change to the first tree’s structure  $\mathfrak{T}_1$  which are accepted or rejected via a Metropolis-Hastings step (Hastings, 1970). Tree structures are altered by introducing small changes: growing a terminal node by adding two terminal daughter nodes, pruning two terminal daughter nodes, or changing a split rule. We denote these possible alterations as: GROW, PRUNE, and CHANGE.<sup>1</sup> Given the tree structure, samples from the posterior of the  $b$  leaf

---

<sup>1</sup>An additional step known as SWAP was proposed in Chipman et al. (2010), but this step is not implemented in the **bartMachine** package which was used to develop **HBART**.

parameters  $\vartheta_1 := \{\mu_1, \dots, \mu_b\}$  are then drawn from the conjugate-normal posterior distribution. This procedure proceeds iteratively for each tree, using an updated set of partial residuals  $\mathbf{R}_j$ .

Once each tree structure and leaf values has been updated, a draw from the posterior of  $\sigma^2$  conditional on all other parameters is made based on the full model residuals  $\mathcal{E} := \mathbf{y} - \sum_{t=1}^m \mathfrak{F}_t^{\vartheta}(\mathbf{X})$ . Finally, a draw from the posterior of  $\gamma$  conditional the other parameters is computed via a Metropolis-Hastings step.

The steps of the full procedure are illustrated below:

$$\begin{aligned}
1 : \quad \mathfrak{F}_1 & \mid R_{-1}, \sigma^2, \gamma \\
2 : \quad \vartheta_1 & \mid \mathfrak{F}_1, R_{-1}, \sigma^2, \gamma \\
& \vdots \\
2m-1 : \quad \mathfrak{F}_m & \mid R_{-m}, \sigma^2, \gamma \\
2m : \quad \vartheta_m & \mid \mathfrak{F}_m, R_{-m}, \sigma^2, \gamma \\
2m+1 : \quad \sigma^2 & \mid \mathfrak{F}_1^{\vartheta}, \dots, \mathfrak{F}_m^{\vartheta}, \gamma, \mathcal{E} \\
2m+2 : \quad \gamma & \mid \sigma^2, \mathcal{E}
\end{aligned} \tag{10.5}$$

All  $2m+2$  steps represent a *single* Gibbs iteration<sup>2</sup>. After a sufficient burn-in period, burned-in draws from the posterior of  $f$  are obtained. A point prediction  $\hat{f}(\mathbf{x})$  can be obtained by taking the average of the burned-in values of  $\mathfrak{F}_1^{\vartheta}, \dots, \mathfrak{F}_m^{\vartheta}$  evaluated at a given  $\mathbf{x}$ ; posterior credible intervals for  $f$  are computed by using the quantiles of the burned-in values. Posterior predictive intervals at a given  $\mathbf{x}$  are computed as follows:

1. Draw  $f(\mathbf{x})$  via

---

<sup>2</sup>BART relies on a similar scheme, removing the conditioning on  $\gamma$  at each step and not requiring step  $2m+2$ .

- i) drawing one of the burned-in sum-of-trees collections  $\mathfrak{T}_1^{\text{leaf}}, \dots, \mathfrak{T}_m^{\text{leaf}}$  and
  - ii) computing  $f(\mathbf{x}) = \sum_{i=1}^m \mathfrak{T}_i^{\text{leaf}}(\mathbf{x})$ .
2. Draw  $\sigma^2(\mathbf{x})$  via
- i) obtaining  $\boldsymbol{\gamma}$  and  $\sigma^2$  from the same Gibbs sample from which the sum-of-trees was obtained,
  - ii) determining the  $\mathbf{z}$  which corresponds to the  $\mathbf{x}$  of interest and
  - iii) computing  $\sigma^2(\mathbf{x})$  by evaluating  $\boldsymbol{\gamma}$ ,  $\sigma^2$  and  $\mathbf{z}$  in the exponentiation of Equation 10.3.
3. Take one draw from  $\mathcal{N}(f(\mathbf{x}), \sigma^2(\mathbf{x}))$  which is the BART model (Equation 10.1)
4. Collect draws from step 3 by repeating steps 1–3 many times. Then, return the desired quantiles.

Note that **HBART** requires modifications to the original **BART** likelihood calculations necessary for Metropolis-Hastings steps to alter the tree structures (steps 1, 3,  $\dots$ ,  $2m - 1$  of Equation A.14). Also, the posterior distributions for the leaf parameters must be updated (steps 2, 4,  $\dots$ ,  $2m$  of Equation A.14). It is worth noting that these modifications are valid for any heteroskedastic **BART** model and not just the log-linear **HBART** model proposed in this work, as they computed as functions of  $\sigma_1^2, \dots, \sigma_n^2$ . Additionally, modifications to the posterior distribution of  $\sigma^2$  (step  $2m + 1$  of Equation A.14) is required. Finally, the sample of  $\boldsymbol{\gamma}$  (step  $2m + 2$  of Equation A.14) is obtained using the Metropolis-Hastings procedure with the proposal distribution outlined in Gamerman (1997). We provide explicit computational details for each of these steps in Appendix A.4.0.1.



## 10.4 Simulations

### 10.4.1 Univariate Model

We begin by comparing the performance of **BART** versus **HBART** in a simple univariate setting. Consider a single predictor  $\mathbf{x}$  which is a uniformly spaced sequence of  $n$  points on  $[0, 1]$ . Then, we consider two models. The first is heteroskedastic and is given by:

$$Y_i = 100x_i + \mathcal{E}_i, \quad \mathcal{E}_i \stackrel{ind}{\sim} \mathcal{N}(0, \sigma_i^2), \quad \sigma_i^2 = \exp(7x_i) \quad (10.6)$$

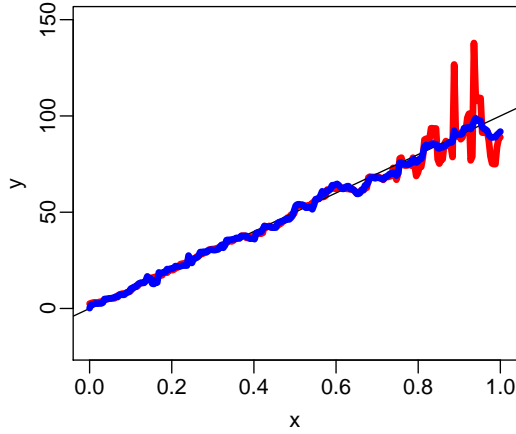
The second is homoskedastic and is given by:

$$Y_i = 100x_i + \mathcal{E}_i, \quad \mathcal{E}_i \stackrel{ind}{\sim} \mathcal{N}(0, 5^2) \quad (10.7)$$

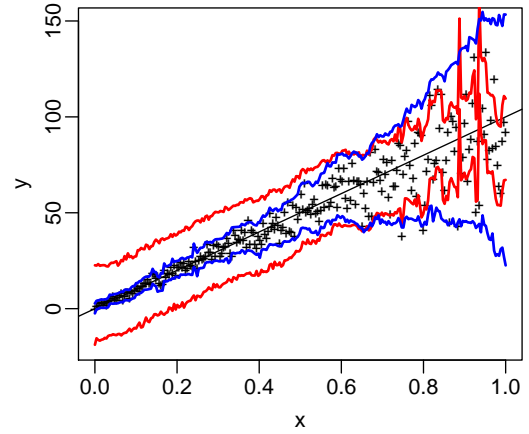
For **HBART**, the matrix of covariates for the parametric variance model will be set to  $\mathbf{Z} = [\mathbf{x}]$ , our one uniformly spaced covariate. Parenthetically, note that the default in our software implementation is to set  $\mathbf{Z} = \mathbf{X}$ . Therefore, **HBART** will have a correctly specified variance function for the model given in Equation 10.6. We include the homoskedastic model as well to benchmark **HBART**'s performance to determine if the unnecessary extra complexity of the variance model degrades the algorithm's performance.

Figure 10.1 compares **BART** to **HBART** in both the heteroskedastic and homoskedastic simulated models by gauging two metrics: accuracy of estimation of  $f$  and appropriateness of posterior predictive intervals.

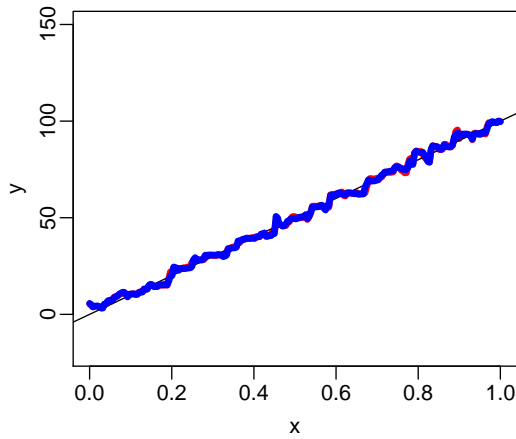
Figure 10.1a highlights the posterior means  $\hat{f}$  estimated by the two different models. Note that **HBART** provides a better estimate of the true  $f$  when  $x > 0.7$ , the region



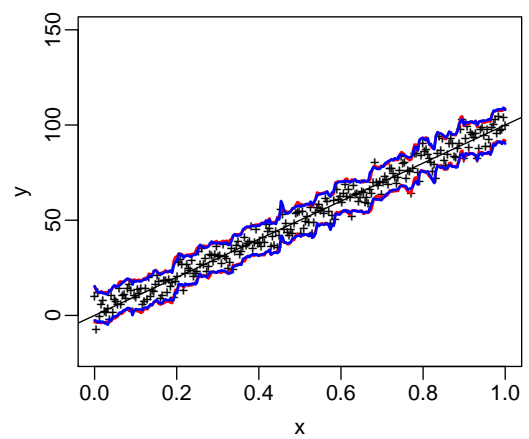
(a) Heteroskedastic Model,  
Posterior Mean Estimates



(b) Heteroskedastic Model,  
Posterior Predictive Intervals



(c) Homoskedastic Model,  
Posterior Mean Estimates



(d) Homoskedastic Model,  
Posterior Predictive Intervals

Figure 10.1: BART's and HBART's posterior mean estimates and 90% posterior predictive intervals for each algorithm built from a sample of  $n = 250$  observations drawn from the processes in Equation 10.6 (a and b) and Equation 10.7 (c and d). Red lines correspond to the results of a BART model and blue lines correspond to the results from a HBART model. The black lines in (b) and (d) correspond to the true conditional mean function ( $Y_i = 100x_i$ ) and the black + 's represent the actual observations.

of relatively high variance. By estimating large variance in this region, it has the flexibility to downweight these high variance observations, allowing for more shrinkage towards the global average and away from the noisy local (within-node) sample mean. In contrast, **BART** overfits in this region. By assuming homoskedasticity, the algorithm is handicapped, and is obligated to move its mean function up when the noise term is large and positive and down when the noise term is large and negative. Note that both algorithms perform well when  $x < 0.7$ , where the data has lower variance.

Figure 10.1b provides 90% posterior predictive intervals for future observations (as explained in the procedure outlined in Section 10.3.2). Given the homoskedasticity assumption of **BART**, the prediction intervals at each  $x$ -location are of constant width. This implies that the intervals are too wide at low values of  $x$  and too narrow at higher values of  $x$ . In contrast, **HBART** provides more appropriate prediction intervals, narrow at low  $x$  and wide at high  $x$ , thus correctly reflecting the heteroskedasticity in the underlying data-generating model. Although not the primary focus of this paper, examining the burned in  $\gamma$  values yielded a 90% credible interval of  $[6.52, 7.65]$ , which captures 7, the value of the linear coefficient in the log-linear variance model of Equation 10.6.

For the homoskedastic model, Figures 10.1c and 10.1d highlight that **BART** and **HBART** yield virtually identical results in terms of mean function estimation and predictive intervals. Thus, **HBART** seems to be robust in the absence of heteroskedasticity for this illustration.

## 10.4.2 Multivariate Model

We now consider the following data generating process, similar to the model simulated in Cepeda and Gamerman (2001):

$$Y_i = f(\mathbf{x}_i) + \mathcal{E}_i, \quad f(\mathbf{x}_i) = -35 + .35x_{1,i} - 1.7x_{2,i}, \quad \mathcal{E}_i \stackrel{iid}{\sim} \mathcal{N}(0, \sigma_i^2) \quad (10.8)$$

$$x_{1,i} \stackrel{iid}{\sim} \text{U}(0, 400), \quad x_{2,i} \stackrel{iid}{\sim} \text{U}(10, 23), \quad x_{3,i} \stackrel{iid}{\sim} \text{U}(0, 10)$$

$$\sigma_i^2 = \exp(-6 + .03x_{1,i} + .4x_{3,i}) \quad (10.9)$$

We set the number of observations to be  $n = 500$  and then we fit an **HBART** model using the default  $\mathbf{Z} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3]$ . Thus, the variance model is misspecified (the true model is not a function of  $\mathbf{x}_2$ ). We again compare **HBART**'s performance to that of **BART**.

Figure 10.2 plots actual values of the conditional mean function versus fitted values. This illustration demonstrates that in areas of high variance, **BART** has difficulty separating the mean function from the noise, and as a result, provides wide credible intervals for the true  $f$ . **HBART**, on the other hand, provides more narrow credible intervals, indicating that the algorithm was able to successfully separate the mean function from the heteroskedastic variance structure in the data generating process.

We next evaluate the performance of **HBART** versus **BART** in terms of out-of-sample predictive performance. We consider two models: one with homoskedastic errors,  $\mathcal{E} \stackrel{iid}{\sim} \mathcal{N}(0, 3^2)$ , and one with heteroskedastic error structure according to Equation 10.9. For each error structure, we then generate  $n = 500$  training observations based on the data generating process outlined in Equation 10.8. Both **HBART** and **BART** models are constructed on the training set and performance is evaluated in terms of root mean square error (RMSE) on an additional  $n = 500$  independent test observations drawn from the same data generating process.

Figure 10.3 displays the out-of-sample performance results for 100 simulations. For the heteroskedastic model, **HBART** significantly outperforms **BART** (Figure 10.3a). For the homoskedastic model, the performance of **BART** and **HBART** are statistically

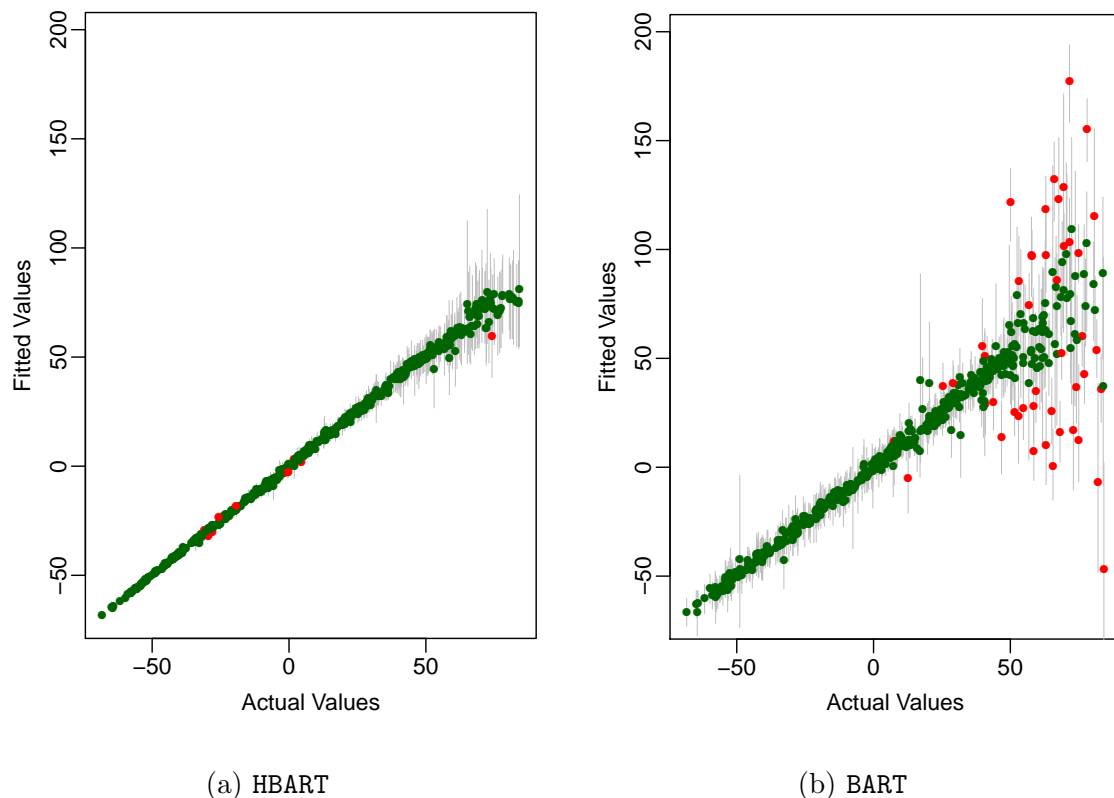


Figure 10.2: Estimates of the conditional mean function  $f$  for HBART and BART with associated 90% credible intervals. The x-axis is the true value of the conditional mean function and the y-value is the model estimate. Gray lines illustrate 90% credible intervals. If the true conditional mean falls within the interval, the point is colored **green** and points in **red** signify the true value falls outside of the interval.

equal and the results of both algorithms are quite stable (Figure 10.3b). This plot provides more evidence of the robustness of HBART's performance in the absence of heteroskedasticity.

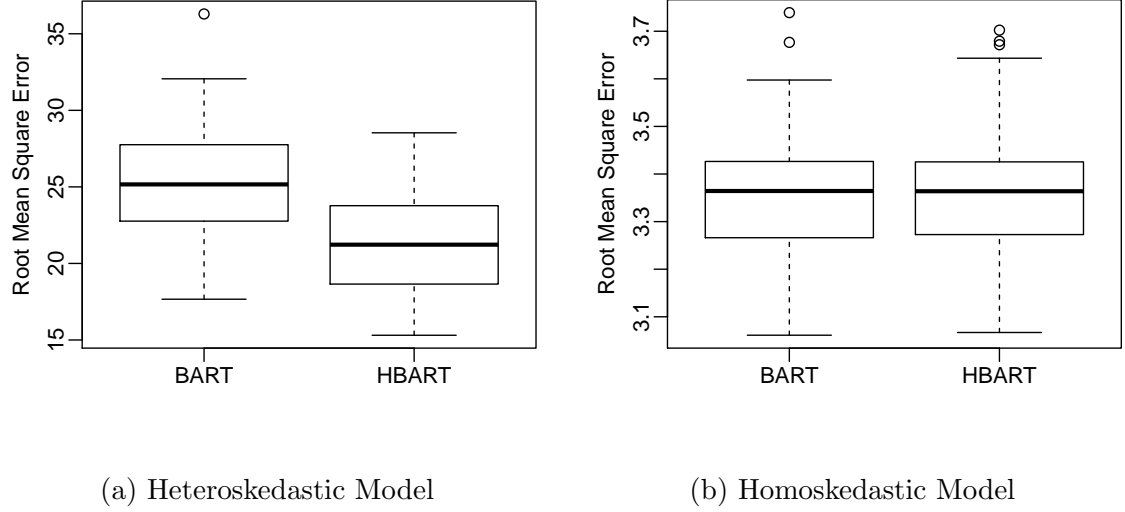


Figure 10.3: Distribution of out-of-sample RMSE for **BART** and **HBART** fit in 100 simulations for (a) the heteroskedastic data generating process of Equations 10.8 and 10.9 and (b) the homoskedastic data generating process of Equation 10.8 with  $\mathcal{E} \stackrel{iid}{\sim} \mathcal{N}(0, 3^2)$ . The models are built on 500 observations and tested on 500 independent observations.

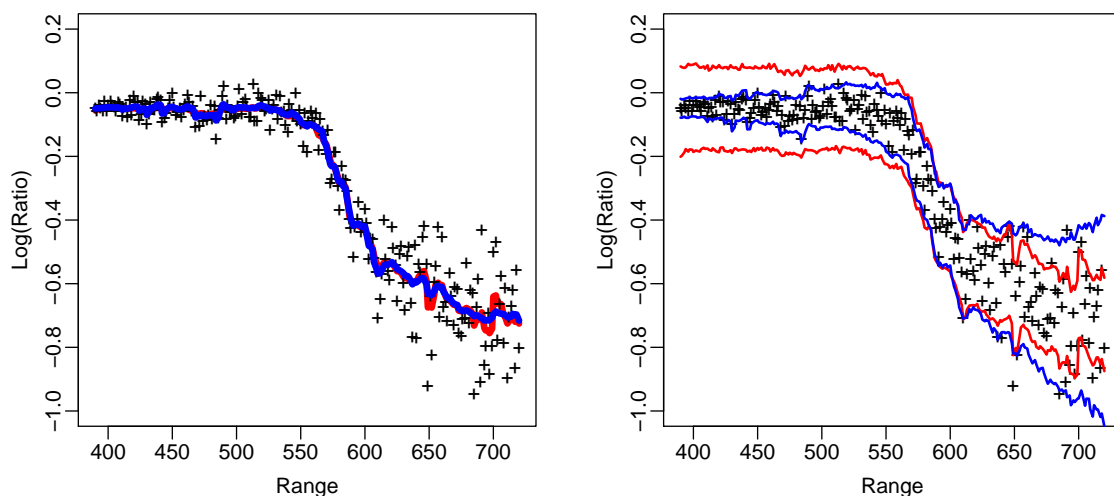
## 10.5 Real Data Examples

### 10.5.1 Lidar Data

We consider the LIDAR data set explored in Ruppert et al. (2003). The data set contains 221 observations. The response denoted **Log(Ratio)** is the logarithm of the ratio of reflected laser-emitted light from two sources and the predictor denoted **Range** is the distance traveled before the light is reflected back to it's source. Leslie et al. (2007) explored this data set by fitting both nonparametric mean and variance functions to the data under the assumptions of both normal and non-normal error distributions.

We fit the data using **BART** and **HBART**. For **HBART**,  $\mathbf{Z}$  is taken to be  $[\mathbf{Range}, \mathbf{Range}^2]$  where the two columns are orthogonalized. Figure 10.4 illustrates the posterior mean

estimates for both HBART and BART. One will notice that both algorithms estimate relatively similar posterior mean functions with HBART's estimation being slightly more smooth in the region of higher variance than that of BART. Figure 10.1b shows 90% posterior predictive intervals for the two algorithms. The intervals for HBART seem to appropriately reflect the heteroskedasticity in the data, while BART's predictive intervals are too wide at low values of `Range` and too narrow at high values of `Range`.



(a) Posterior Mean Estimate

(b) Posterior Predictive Intervals

Figure 10.4: BART's and HBART's posterior mean estimates and 90% posterior predictive intervals for the Lidar data. **Red** lines correspond to the BART model and **blue** lines correspond to the HBART model.

## 10.5.2 Motorcycle Data

We next consider a dataset of simulated motorcycle crashes that was compiled by Schmidt et al. (1981). The observations in the data set consist of accelerometer readings (`acceleration`) taken from riders' helmets at 133 different points in time (`time`) after a simulated impact. As discussed in Gramacy (2007), many researchers find that this dataset exhibits multiple regimes in both the mean function and variance

function over time.

This dataset was also explored by Taddy et al. (2011) who remarked that the 90% posterior predictive interval for **BART** appeared to “variously over or under estimate data uncertainty around the regression mean. In particular, **BART**’s global variance term is misspecified for this heteroskedastic data.” We attempt to remedy this problem with **HBART**. Exploring a scatterplot of the data, the model seems to be characterized by a low variance regime followed by a high variance regime and then an additional low variance regime (see Figure 10.5). Hence, when building an **HBART** model, we do not use the default  $\mathbf{Z}$ . To capture the low-high-low variance relationship, we specify the model quadratic in the predictor,  $\mathbf{Z} = [\text{time}, \text{time}^2]$  where the two columns are orthogonalized.

Figure 10.5 displays 90% posterior predictive intervals for **HBART**, **BART**, dynamic trees (**dynaTree**, Taddy et al., 2011), and treed Gaussian processes (**tgp**, Gramacy, 2007). Each of the algorithms has a similar estimate of the posterior mean process (unshown), but there are some differences in the posterior predictive intervals. **BART**’s predictive intervals are much too wide at low and high values of time and perhaps too narrow for the intermediate values of time. However, **HBART**’s intervals are quite similar to those of **dynaTree**, being widest for intermediate values of time and more narrow near the beginning and end. Interestingly, **HBART** is the only of the four models that builds a narrower predictive interval at the higher values of time.

## 10.6 Discussion

We have proposed **HBART**, an extension of **BART** that relaxes the assumption of homoskedasticity in the model errors. In particular, we have developed a model that allows for a multiplicative heteroskedastic error structure, where the multiplicative factor is the exponential of a parametric function of some set of covariates.



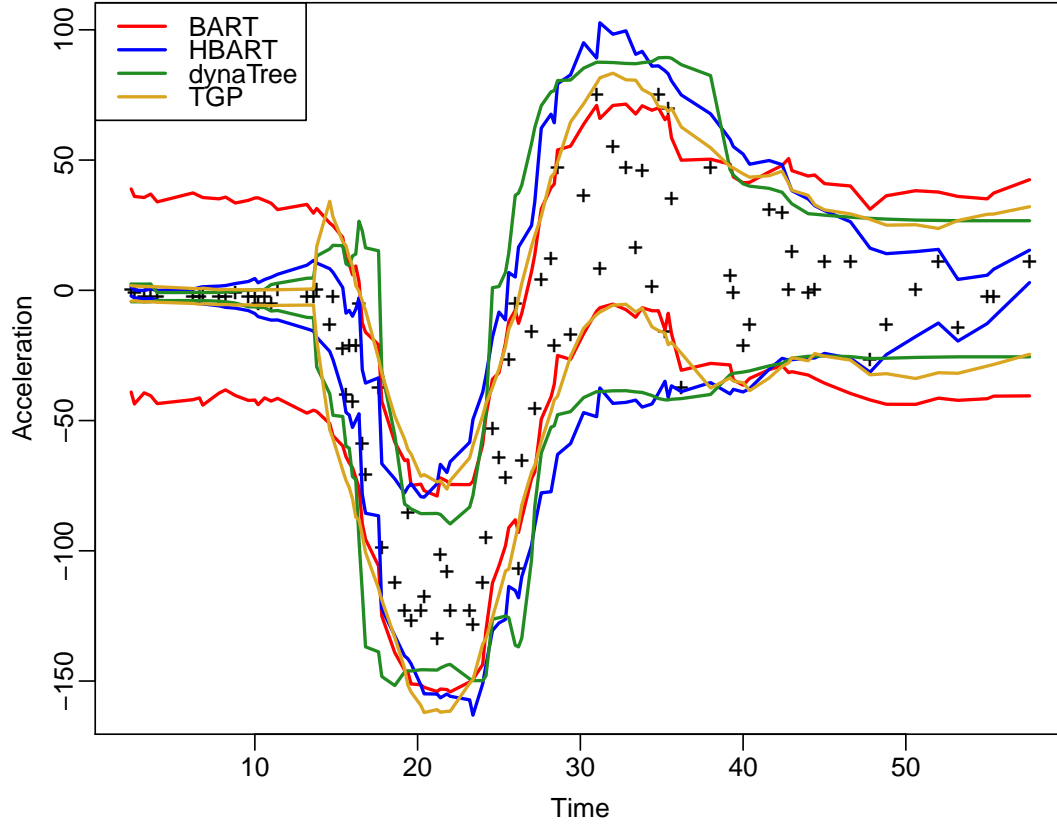


Figure 10.5: 90% posterior predictive intervals for BART, HBART, dynaTree and TGP.

Through simulations and explorations of real data, we have demonstrated **HBART**'s potential for generating more appropriate posterior predictive intervals in the presence of appropriately modeled heteroskedastic data versus **BART**. Additionally, **BART** suffers from overfitting in areas of very high variance and **HBART** seems to offer promise in ameliorating this issue. In our explorations, the added complexity of fitting a model to the error terms did not hinder **HBART**'s performance on homoskedastic data. In this situation, **HBART**'s estimates of the posterior means, predictive intervals, and out-of-sample RMSE were very similar to **BART**'s.

Originally proposed in Chipman et al. (2010) and implemented in Kapelner and Bleich (2013), it is possible to cross-validate over a number of the hyperparameters of the **BART** model. Future work will involve extending **BART-CV** to **HBART-CV**,

where it is possible to cross-validate over a selection of prior variances for the parametric variance model to impose varying degrees of shrinkage by modifying the  $\Sigma_{jj}$  hyperparameters. One final direction of research is to further relax the assumptions on the error structure. For instance, one could incorporate more flexible variance models such as smoothing splines instead of the standard linear model or relax the assumption of normality of the errors by considering Dirichlet mixture priors.

## Replication

The code for HBART is located at <http://github.com/kapelner/bartMachine> in the branch “hBART” and it is open source under the MIT license. The results, tables, and figures found in this paper can be replicated via the `replication.R` script located in the `hbart_paper` folder within this git repository. The code can be conveniently installed as a local R package.

## Acknowledgements

We thank Richard Berk, Ed George and Abba Krieger for helpful discussions. We thank Simon Urbanek for his generous help with rJava. Adam Kapelner acknowledges support from the National Science Foundation’s Graduate Research Fellowship Program.

## Statistical Learning Model Visualization

By Alex Goldstein, Adam Kapelner, Justin Bleich and Emil Pitkin

### Abstract

This article presents Individual Conditional Expectation (ICE) plots, a tool for visualizing the model estimated by any supervised learning algorithm. Classical partial dependence plots (PDPs) help visualize the average partial relationship between the predicted response and one or more features. In the presence of substantial interaction effects, the partial response relationship can be heterogeneous. Thus, an average curve, such as the PDP, can obfuscate the complexity of the modeled relationship. Accordingly, ICE plots refine the partial dependence plot by graphing the functional relationship between the predicted response and the feature for *individual* observations. Specifically, ICE plots highlight the variation in the fitted values across the range of a covariate, suggesting where and to what extent heterogeneities might exist. In addition to providing a plotting suite for exploratory analysis, we include a visual test for additive structure in the data generating model. Through simulated examples and real data sets, we demonstrate how ICE plots can shed light on estimated models in ways PDPs cannot. Procedures outlined are available in the R package `ICEbox`.

## 11.1 Introduction

The goal of this article is to present Individual Conditional Expectation (ICE) plots, a toolbox for visualizing models produced by “black box” algorithms. These algorithms use training data  $\{\mathbf{x}_i, y_i\}_{i=1}^N$  (where  $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,p})$  is a vector of predictors and  $y_i$  is the response) to construct a model  $\hat{f}$  that maps the features  $\mathbf{x}$  to fitted values  $\hat{f}(\mathbf{x})$ . Though these algorithms can produce fitted values that enjoy low generalization error, it is often difficult to understand how the resultant  $\hat{f}$  uses  $\mathbf{x}$  to generate predictions. The ICE toolbox helps visualize this mapping.

ICE plots extend Friedman (2001)’s Partial Dependence Plot (PDP), which highlights the average partial relationship between a set of predictors and the predicted response. ICE plots disaggregate this average by displaying the estimated functional relationship for each observation. Plotting a curve for each observation helps identify interactions in  $\hat{f}$  as well as extrapolations in predictor space.

The paper proceeds as follows. Section 11.2 gives background on visualization in machine learning and introduces PDPs more formally. Section 11.3 describes the procedure for generating ICE plots and its associated plots. In Section 13.3 simulated data examples illustrate that ICE plots can be used to identify features of  $\hat{f}$  that are not visible in PDPs, or where the PDPs may even be misleading. Each example is chosen to illustrate a particular principle. Section 13.4 provides examples of ICE plots on real data. In Section 11.6 we shift the focus from the fitted  $\hat{f}$  to a data generating process  $f$  and use ICE plots as part of a visual test for additivity in  $f$ . Section 13.5 concludes.

## 11.2 Background

### 11.2.1 Survey of Black Box Visualization

There is an extensive literature that attests to the superiority of black box machine learning algorithms in minimizing predictive error, both from a theoretical and an applied perspective. Breiman (2001b), summarizing, states “accuracy generally requires more complex prediction methods ...[and] simple and interpretable functions do not make the most accurate predictors.” Problematically, black box models offer little in the way of interpretability, unless the data is of very low dimension. When we are willing to compromise interpretability for improved predictive accuracy, any window into black box’s internals can be beneficial.

Authors have devised a variety of algorithm-specific techniques targeted at improving the interpretability of a particular statistical learning procedure’s output. Rao and Potts (1997) offers a technique for visualizing the decision boundary produced by bagging decision trees. Although applicable to high dimensional settings, their work primarily focuses on the low dimensional case of two covariates. Tzeng (2005) develops visualization of the layers of neural networks to understand dependencies between the inputs and model outputs and yields insight into classification uncertainty. Jakulin et al. (2005) improves the interpretability of support vector machines by using a device called “nomograms” which provide graphical representation of the contribution of variables to the model fit. Pre-specified interaction effects of interest can be displayed in the nomograms as well. Breiman (2001a) uses randomization of out-of-bag observations to compute a variable importance metric for Random Forests (RF). Those variables for which predictive performance degrades the most vis-a-vis the original model are considered the strongest contributors to forecasting accuracy. This method is also applicable to stochastic gradient boosting (Friedman, 2002). Plate et al. (2000) plots neural network predictions in a scatterplot for each

variable by sampling points from covariate space. Amongst the existing literature, this work is the most similar to ICE, but was only applied to neural networks and does not have a readily available implementation.

Other visualization proposals are model agnostic and can be applied to a host of supervised learning procedures. For instance, Strumbelj and Kononenko (2011) consider a game-theoretic approach to assess the contributions of different features to predictions that relies on an efficient approximation of the Shapley value. Jiang and Owen (2002) use quasi-regression estimation of black box functions. Here, the function is expanded into an orthonormal basis of coefficients which are approximated via Monte Carlo simulation. These estimated coefficients can then be used to determine which covariates influence the function and whether any interactions exist.

### 11.2.2 Friedman’s PDP

Another particularly useful model agnostic tool is Friedman (2001)’s PDP, which this paper extends. The PDP plots the change in the average predicted value as specified feature(s) vary over their marginal distribution. Many supervised learning models applied across a number of disciplines have been better understood thanks to PDPs. Green et al. (2010) use PDPs to understand the relationship between predictors and the conditional average treatment effect for a voter mobilization experiment, with the predictions being made by Bayesian Additive Regression Trees (BART, Chipman et al., 2010). Berk and Bleich (ress) demonstrate the advantage of using RF and the associated PDPs to accurately model predictor-response relationships under asymmetric classification costs that often arise in criminal justice settings. In the ecological literature, Elith et al. (2008), who rely on stochastic gradient boosting, use PDPs to understand how different environmental factors influence the distribution of a particular freshwater eel.

To formally define the PDP, let  $S \subset \{1, \dots, p\}$  and let  $C$  be the complement set of

$S$ . Here  $S$  and  $C$  index subsets of predictors; for example, if  $S = \{1, 2, 3\}$ , then  $\mathbf{x}_S$  refers to a  $3 \times 1$  vector containing the values of the first three coordinates of  $\mathbf{x}$ . Then the partial dependence function of  $f$  on  $\mathbf{x}_S$  is given by

$$f_S = \mathbb{E}_{\mathbf{x}_C} [f(\mathbf{x}_S, \mathbf{x}_C)] = \int f(\mathbf{x}_S, \mathbf{x}_C) dP(\mathbf{x}_C). \quad (11.1)$$

Each subset of predictors  $S$  has its own partial dependence function  $f_S$ , which gives the average value of  $f$  when  $\mathbf{x}_S$  is fixed and  $\mathbf{x}_C$  varies over its marginal distribution  $dP(\mathbf{x}_C)$ . As neither the true  $f$  nor  $dP(\mathbf{x}_C)$  are known, we estimate Equation 11.1 by computing

$$\hat{f}_S = \frac{1}{N} \sum_{i=1}^N \hat{f}(\mathbf{x}_S, \mathbf{x}_{Ci}) \quad (11.2)$$

where  $\{\mathbf{x}_{C1}, \dots, \mathbf{x}_{CN}\}$  represent the different values of  $\mathbf{x}_C$  that are observed in the training data. Note that the approximation here is twofold: we estimate the true model with  $\hat{f}$ , the output of a statistical learning algorithm, and we estimate the integral over  $\mathbf{x}_C$  by averaging over the  $N$   $\mathbf{x}_C$  values observed in the training set.

This is a visualization tool in the following sense: if  $\hat{f}_S$  is evaluated at the  $\mathbf{x}_S$  observed in the data, a set of  $N$  ordered pairs will result:  $\{(\mathbf{x}_{S\ell}, \hat{f}_{S\ell})\}_{\ell=1}^N$ , where  $\hat{f}_{S\ell}$  refers to the estimated partial dependence function evaluated at the  $\ell$ th coordinate of  $\mathbf{x}_S$ , denoted  $\mathbf{x}_{S\ell}$ . Then for one or two dimensional  $\mathbf{x}_S$ , Friedman (2001) proposes plotting the  $N$   $\mathbf{x}_{S\ell}$ 's versus their associated  $\hat{f}_{S\ell}$ 's, conventionally joined by lines. The resulting graphic, which is called a partial dependence plot, displays the average value of  $\hat{f}$  as a function of  $\mathbf{x}_S$ . For the remainder of the paper we consider a single predictor of interest at a time ( $|S| = 1$ ) and write  $x_S$  without boldface accordingly.

As an extended example, consider the following data generating process with a

simple interaction:

$$Y = 0.2X_1 - 5X_2 + 10X_2\mathbb{1}_{X_3 \geq 0} + \mathcal{E}, \quad (11.3)$$

$$\mathcal{E} \stackrel{iid}{\sim} \mathcal{N}(0, 1), \quad X_1, X_2, X_3 \stackrel{iid}{\sim} \text{U}(-1, 1).$$

We generate 1,000 observations from this model and fit a stochastic gradient boosting model (**SGB**) via the **R** package **gbm** (Ridgeway, 2013) where the number of trees is chosen via cross-validation and the interaction depth is set to 3. We now consider the association between predicted  $Y$  values and  $X_2$  ( $S = X_2$ ). In Figure 11.1a we plot  $X_2$  versus  $Y$  in our sample. Figure 11.1b displays the fitted model's partial dependence plot for predictor  $X_2$ . The PDP suggests that on average,  $X_2$  is not meaningfully associated with the predicted  $Y$ . In light of Figure 11.1a, this conclusion is plainly wrong. Clearly  $X_2$  is associated with  $Y$ ; it is simply that the averaging inherent in the PDP shields this discovery from view.

In fact, the original work introducing PDPs argues that the PDP can be a useful summary for the chosen subset of variables if their dependence on the remaining features is not too strong. When the dependence is strong, however – that is, when interactions are present – the PDP can be misleading. Nor is the PDP particularly effective at revealing extrapolations in  $\mathcal{X}$ -space. ICE plots are intended to address these issues.

## 11.3 The ICE Toolbox

### 11.3.1 The ICE Procedure

Visually, ICE plots disaggregate the output of classical PDPs. Rather than plot the target covariates' *average* partial effect on the predicted response, we instead plot the



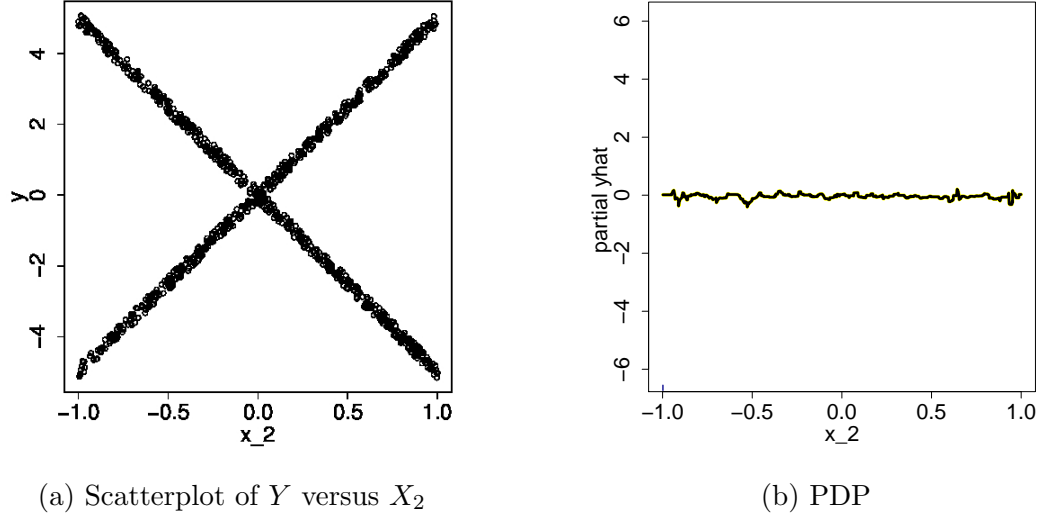


Figure 11.1: Scatterplot and PDP of  $X_2$  versus  $Y$  for a sample of size 1000 from the process described in Equation 11.3. In this example  $\hat{f}$  is fit using SGB. The PDP incorrectly suggests that there is no meaningful relationship between  $X_2$  and the predicted  $Y$ .

$N$  estimated conditional expectation curves: each reflects the predicted response as a function of covariate  $x_S$ , conditional on an observed  $\mathbf{x}_C$ .

Consider the observations  $\{(x_{Si}, \mathbf{x}_{Ci})\}_{i=1}^N$ , and the estimated response function  $\hat{f}$ . For each of the  $N$  observed and fixed values of  $\mathbf{x}_C$ , a curve  $\hat{f}_S^{(i)}$  is plotted against the observed values of  $x_S$ . Therefore, at each x-coordinate,  $x_S$  is fixed and the  $\mathbf{x}_C$  varies across  $N$  observations. Each curve defines the conditional relationship between  $x_S$  and  $\hat{f}$  at fixed values of  $\mathbf{x}_C$ . Thus, the ICE algorithm gives the user insight into the several variants of conditional relationships estimated by the black box.

The ICE algorithm is given in Algorithm 8 in Appendix A.5. Note that the PDP curve is the average of the  $N$  ICE curves and can thus be viewed as a form of post-processing. Although in this paper we focus on the case where  $|S| = 1$ , the pseudocode is general. All plots in this paper are produced using the R package `ICEbox`, available on CRAN.

Returning to the simulated data described by Equation 11.3, Figure 11.2 shows

the ICE plot for the **SGB** when  $S = X_2$ . In contrast to the PDP in Figure 11.1b, the ICE plot makes it clear that the fitted values *are* related to  $X_2$ . Specifically, the **SGB**'s predicted values are approximately linearly increasing or decreasing in  $X_2$  depending upon which region of  $\mathcal{X}$  an observation is in.

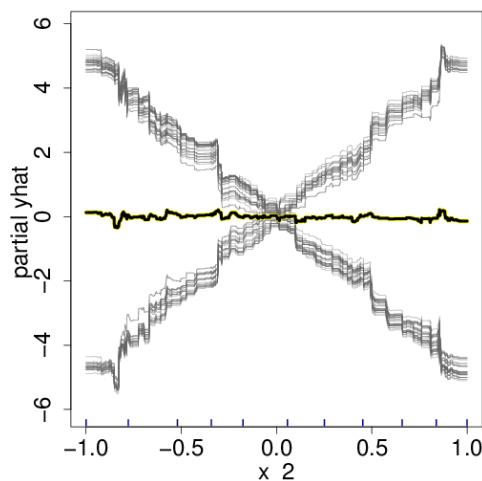


Figure 11.2: **SGB** ICE plot for  $X_2$  from 1000 realizations of the data generating process described by Equation 11.3. We see that the **SGB**'s fitted values are either approximately linearly increasing or decreasing in  $X_2$ .

Now consider the well known Boston Housing Data (BHD). The goal in this dataset is to predict a census tract's median home price using features of the census tract itself. It is important to note that the median home prices for the tracts are truncated at 50, and hence one may observe potential ceiling effects when analyzing the data. We use Random Forests (**RF**) implemented in **R** (Liaw and Wiener, 2002) to fit  $\hat{f}$ . The ICE plot in Figure 11.3 examines the association between the average age of homes in a census tract and the corresponding median home value for that tract ( $S = \text{age}$ ). The PDP is largely flat, perhaps displaying a slight decrease in predicted median home price as **age** increases. The ICE plot shows those observations for which increasing **age** is actually associated with higher predicted values, thereby describing how individual behavior departs from the average behavior.

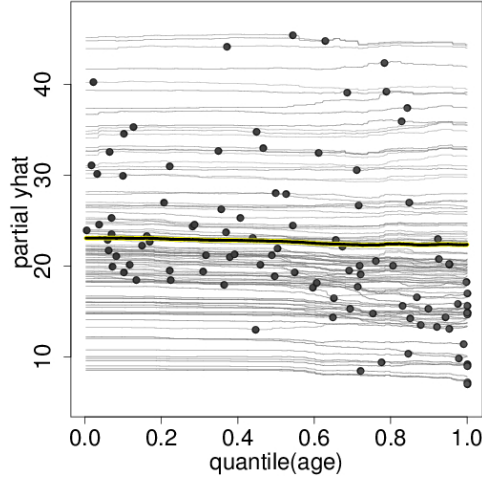


Figure 11.3: RF ICE plot for BHD for predictor `age`. The highlighted thick line is the PDP. For each curve, the location of its observed `age` is marked by a point. For some observations, higher `age` is associated with a higher predicted values. The upper set of tick marks on the horizontal axis indicate the observed deciles of `age`.

### 11.3.2 The Centered ICE Plot

When the curves have a wide range of intercepts and are consequently “stacked” on each other, heterogeneity in the model can be difficult to discern. In Figure 11.3, for example, the variation in effects between curves and cumulative effects are veiled. In such cases the “centered ICE” plot (the “c-ICE”), which removes level effects, is useful.

c-ICE works as follows. Choose a location  $x^*$  in the range of  $x_S$  and join or “pinch” all prediction lines at that point. We have found that choosing  $x^*$  as the minimum or the maximum observed value results in the most interpretable plots. For each curve  $\hat{f}^{(i)}$  in the ICE plot, the corresponding c-ICE curve is given by

$$\hat{f}_{\text{cent}}^{(i)} = \hat{f}^{(i)} - \mathbf{1}\hat{f}(x^*, \mathbf{x}_{Ci}), \quad (11.4)$$

where the unadorned  $\hat{f}$  denotes the fitted model and  $\mathbf{1}$  is a vector of 1’s of the

appropriate dimension. Hence the point  $(x^*, \hat{f}(x^*, \mathbf{x}_{Ci}))$  acts as a “base case” for each curve. If  $x^*$  is the minimum value of  $x_S$ , for example, this ensures that all curves originate at 0, thus removing the differences in level due to the different  $\mathbf{x}_{Ci}$ ’s. At the maximum  $x_S$  value, each centered curve’s level reflects the cumulative effect of  $x_S$  on  $\hat{f}$  relative to the base case. The result is a plot that better isolates the combined effect of  $x_S$  on  $\hat{f}$ , holding  $\mathbf{x}_C$  fixed.

Figure 11.4 shows a c-ICE plot for the predictor **age** of the BHD for the same RF model as examined previously. From the c-ICE plot we can now see clearly that the cumulative effect of **age** on predicted median value increases for some cases, and decreases for others. Such divergences of the centered curves suggest the existence of interactions between  $x_S$  and  $\mathbf{x}_C$  in the model. Also, the magnitude of the effect, as a fraction of the range of  $y$ , can be seen in the vertical axis displayed on the right of the graph.

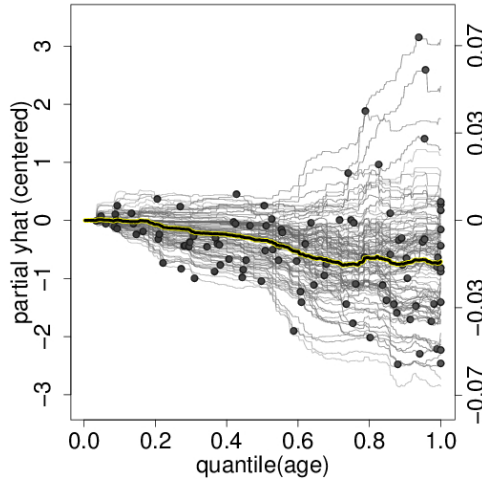


Figure 11.4: c-ICE plot for **age** with  $x^*$  set to the minimum value of **age**. The right vertical axis displays changes in  $\hat{f}$  over the baseline as a fraction of  $y$ ’s observed range. In this example, interactions between **age** and other predictors create cumulative differences in fitted values of up to about 14% of the range of  $y$ .

### 11.3.3 The Derivative ICE Plot

To further explore the presence of interaction effects, we develop plots of the partial derivative of  $\hat{f}$  with respect to  $x_S$ . To illustrate, consider the scenario in which  $x_S$  does not interact with the other predictors in the fitted model. This implies  $\hat{f}$  can be written as

$$\hat{f}(\mathbf{x}) = \hat{f}(x_S, \mathbf{x}_C) = g(x_S) + h(\mathbf{x}_C), \quad \text{so that} \quad \frac{\partial \hat{f}(\mathbf{x})}{\partial x_S} = g'(x_S), \quad (11.5)$$

meaning the relationship between  $x_S$  and  $\hat{f}$  does not depend on  $\mathbf{x}_C$ . Thus the ICE plot for  $x_S$  would display a set of  $N$  curves that share a single common shape but differ by level shifts according to the values of  $\mathbf{x}_C$ .

As it can be difficult to visually assess derivatives from ICE plots, it is useful to plot an estimate of the partial derivative directly. The details of this procedure are given in Algorithm 9 in Appendix A.5. We call this a “derivative ICE” plot, or “d-ICE.” When no interactions are present in the fitted model, all curves in the d-ICE plot are equivalent, and the plot shows a single line. When interactions do exist, the derivative lines will be heterogeneous.

As an example, consider the d-ICE plot for the RF model in Figure 11.5. The plot suggests that when **age** is below approximately 60,  $g' \approx 0$  for all observed values of  $\mathbf{x}_C$ . In contrast, when **age** is above 60 there are observations for which  $g' > 0$  and others for which  $g' < 0$ , suggesting an interaction between **age** and the other predictors. Also, the standard deviation of the partial derivatives at each point, plotted in the lower panel, serves as a useful summary to highlight regions of heterogeneity in the estimated derivatives (i.e., potential evidence of interactions in the fitted model).

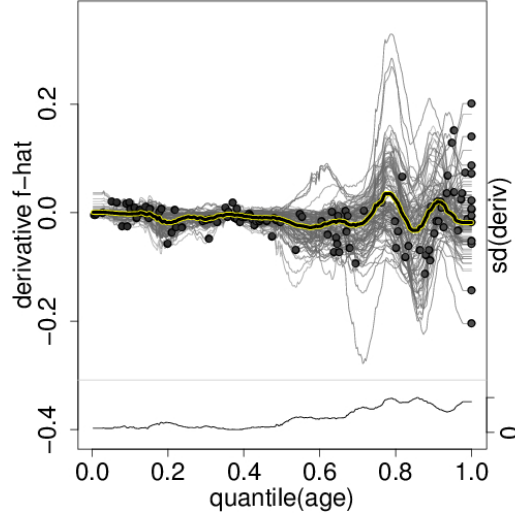


Figure 11.5: d-ICE plot for `age` in the BHD. The left vertical axis' scale gives the partial derivative of the fitted model. Below the d-ICE plot we plot the standard deviation of the derivative estimates at each value of `age`. The scale for this standard deviation plot is on the bottom of the right vertical axis.

### 11.3.4 Visualizing a Second Feature

Color allows overloading of ICE, c-ICE and d-ICE plots with information regarding a second predictor of interest  $x_k$ . Specifically, one can assess how the second predictor influences the relationship between  $x_S$  and  $\hat{f}$ . If  $x_k$  is categorical, we assign colors to its levels and plot each prediction line  $\hat{f}^{(i)}$  in the color of  $x_{ik}$ 's level. If  $x_k$  is continuous, we vary the color shade from light (low  $x_k$ ) to dark (high  $x_k$ ).

We replot the c-ICE from Figure 11.4 with lines colored by a newly constructed predictor,  $x = 1(\text{rm} > \text{median}(\text{rm}))$ . Lines are colored red if the average number of rooms in a census tract is greater than the median number of rooms across all all census tracts and are colored blue otherwise. Figure 11.6 suggests that for census tracts with a larger number of average rooms, predicted median home price value is positively associated with `age` and for census tracts with a lesser number of average rooms, the association is negative.

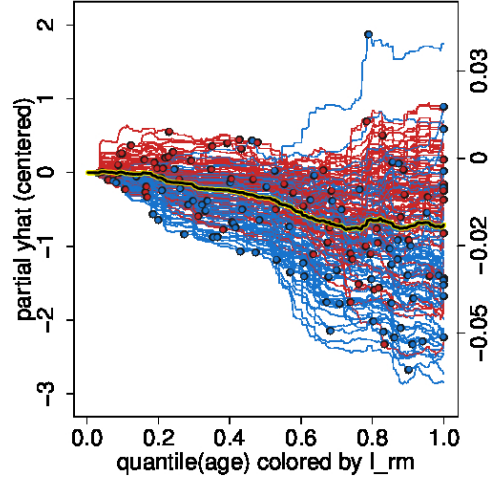


Figure 11.6: The c-ICE plot for `age` of Figure 11.4 in the BHD. Red lines correspond to observations with `rm` greater than the median `rm` and blue lines correspond to those with fewer.

## 11.4 Simulations

Each of the following examples is designed to emphasize a particular model characteristic that the ICE toolbox can detect. To more clearly demonstrate given scenarios with minimal interference from issues that one typically encounters in actual data, such as noise and model misspecification, the examples are purposely stylized.

### 11.4.1 Additivity Assessment

We begin by showing that ICE plots can be used as a diagnostic in evaluating the extent to which a fitted model  $\hat{f}$  fits an additive model.

Consider again the prediction task in which  $\hat{f}(\mathbf{x}) = g(x_S) + h(\mathbf{x}_C)$ . For arbitrary vectors  $\mathbf{x}_{C_i}$  and  $\mathbf{x}_{C_j}$ ,  $\hat{f}(x_S, \mathbf{x}_{C_i}) - \hat{f}(x_S, \mathbf{x}_{C_j}) = h(\mathbf{x}_{C_i}) - h(\mathbf{x}_{C_j})$  for all values of  $x_S$ . The term  $h(\mathbf{x}_{C_i}) - h(\mathbf{x}_{C_j})$  represents the shift in level due to the difference between  $\mathbf{x}_{C_i}$  and  $\mathbf{x}_{C_j}$  and is independent of the value of  $x_S$ . Thus the ICE plot for  $x_S$  will display a set of  $N$  curves that share a common shape but differ by level shifts

according to the unique values of  $\mathbf{x}_C$ .

As an illustration, consider the following additive data generating model

$$Y = X_1^2 + X_2 + \mathcal{E}, \quad X_1, X_2 \stackrel{iid}{\sim} \text{U}(-1, 1), \quad \mathcal{E} \stackrel{iid}{\sim} \mathcal{N}(0, 1).$$

We simulate 1000 independent  $(\mathbf{X}_i, Y_i)$  pairs according to the above and fit a generalized additive model (**GAM**, Hastie and Tibshirani, 1986) via the **R** package **gam** (Hastie, 2013). As we have specified it, the **GAM** assumes

$$f(\mathbf{X}) = f_1(X_1) + f_2(X_2) + f_3(X_1 X_2)$$

where  $f_1$ ,  $f_2$  and  $f_3$  are unknown functions estimated internally by the procedure using smoothing splines. Because  $f_3$  appears in the model specification but not in the data generating process, any interaction effects that **GAM** fits are spurious.<sup>1</sup> Here, ICE plots inform us of the degree to which interactions were fit. Were there no interaction in  $\hat{f}$  between  $X_1$  and  $X_2$ , the ICE plots for  $X_1$  would display a set of curves equivalent in shape but differing in level.

Figure 11.7a displays the ICE plots for  $X_1$  and indicates that this is indeed the case: all curves display a similar parabolic relationship between  $\hat{f}$  and  $X_1$ , shifted by a constant, and independent of the value of  $X_2$ . Accordingly, the associated d-ICE plot in Figure 11.7b displays little variation between curves. The ICE suite makes it apparent that  $f_3$  (correctly) contributes relatively little to the **GAM** model fit. Note that additive structure cannot be observed from the PDP alone in this example (or any other).

---

<sup>1</sup>If we were to eliminate  $f_3$  from the **GAM** then we would know a priori that  $\hat{f}$  would not display interaction effects.



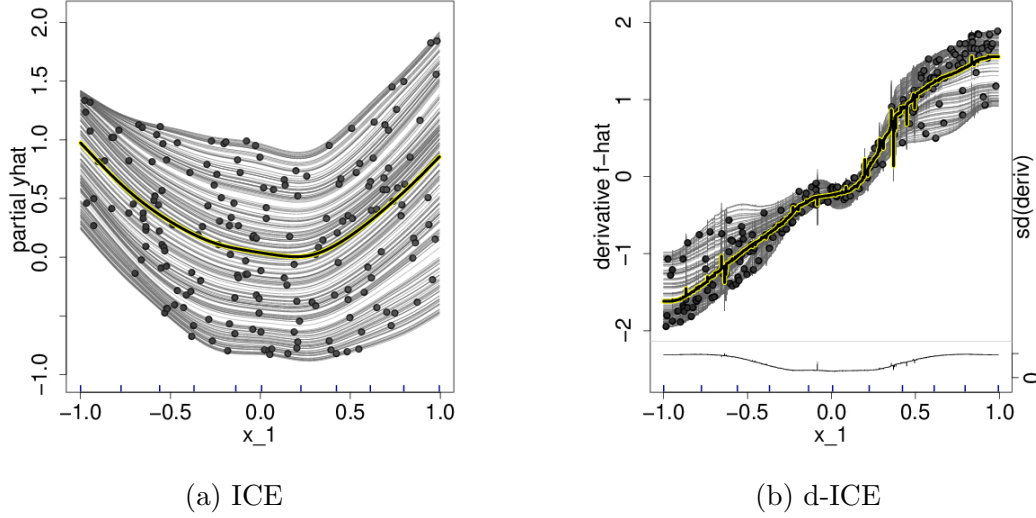


Figure 11.7: ICE and d-ICE plots for  $S = X_1$  when  $\hat{f}$  is a GAM with possible interaction effects between  $X_1$  and  $X_2$ . So as to keep the plot uncluttered we plot only a fraction of all 1000 curves. In the ICE plots the dots indicate the actual location of  $X_1$  for each curve.

#### 11.4.2 Finding interactions and regions of interactions

As noted in Friedman (2001), the PDP is most instructive when there are no interactions between  $x_S$  and the other features. In the presence of interaction effects, the averaging procedure in the PDP can obscure any heterogeneity in  $\hat{f}$ . Let us return to the simple interaction model

$$Y = 0.2X_1 - 5X_2 + 10X_2\mathbf{1}_{X_3 \geq 0} + \mathcal{E}, \quad (11.6)$$

$$\mathcal{E} \stackrel{iid}{\sim} \mathcal{N}(0, 1), \quad X_1, X_2, X_3 \stackrel{iid}{\sim} \text{U}(-1, 1)$$

to examine the relationship between SGB's  $\hat{f}$  and  $X_3$ . Figure 11.8a displays an ICE plot for  $X_3$ . Similar to the PDP we saw in Section 12.1, the plot suggests that averaged over  $X_1$  and  $X_2$ ,  $\hat{f}$  is not associated with  $X_3$ . By following the non-parallel ICE curves, however, it is clear that  $X_3$  modulates the fitted value through interactions

with  $X_1$  and  $X_2$ .

Where in the range of  $X_3$  do these interactions occur? The d-ICE plot of Figure 11.8b shows that interactions are in a neighborhood around  $X_3 \approx 0$ . This is expected; in the model given by Equation 11.6, being above or below  $X_3 = 0$  changes the response level. The plot suggests that the fitted model’s interactions are concentrated in  $X_3 \in [-0.025, 0.025]$  which we call the “region of interaction” (ROI).

Generally, ROIs are identified by noting where the derivative lines are variable. In our example, the lines have highly variable derivatives (both positive and negative) in  $[-0.025, 0.025]$ . The more heterogeneity in these derivative lines, the larger the effect of the interaction between  $x_S$  and  $\mathbf{x}_C$  on the model fit. ROIs can be seen most easily by plotting the standard deviation of the derivative lines at each  $x_S$  value. In this example, the standard deviation function is plotted in the bottom pane of Figure 11.8b and demonstrates that fitted interactions peak at  $X_3 \approx 0$ .

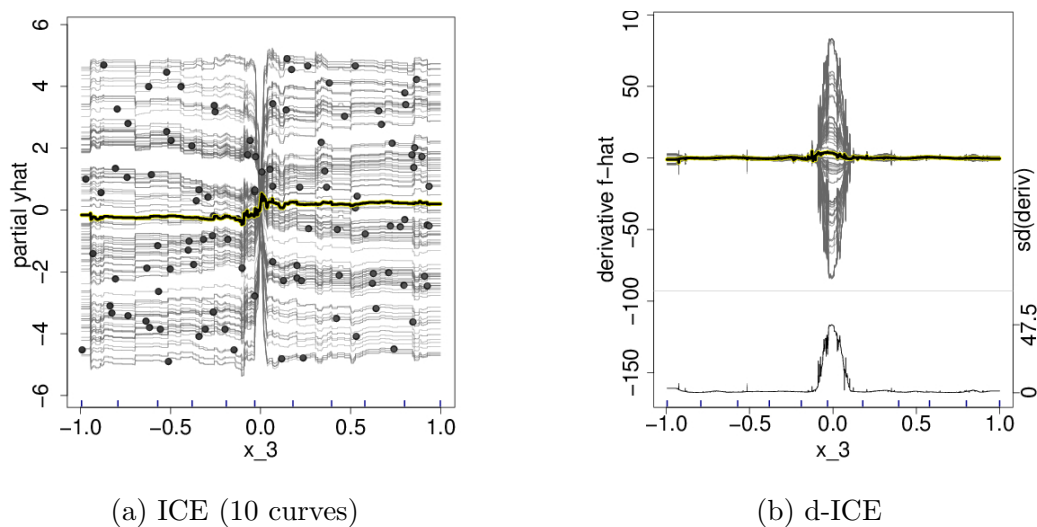


Figure 11.8: ICE plots for an SGB fit to the simple interaction model of Equation 11.6.

### 11.4.3 Extrapolation Detection

As the number of predictors  $p$  increases, the sample vectors  $\mathbf{x}_1, \dots, \mathbf{x}_N$  are increasingly sparse in the feature space  $\mathcal{X}$ . A consequence of this curse of dimensionality is that for many  $\mathbf{x} \in \mathcal{X}$ ,  $\hat{f}(\mathbf{x})$  represents an extrapolation rather than an interpolation (see Hastie et al., 2009 for a more complete discussion).

Extrapolation may be of particular concern when using a black-box algorithm to forecast  $\mathbf{x}_{\text{new}}$ . Not only may  $\hat{f}(\mathbf{x}_{\text{new}})$  be an extrapolation of the  $(\mathbf{x}, y)$  relationship observed in the training data, but the black-box nature of  $\hat{f}$  precludes us from gaining any insight into what the extrapolation might look like. Fortunately, ICE plots can cast light into these extrapolations.

Recall that each curve in the ICE plot includes the fitted value  $\hat{f}(x_{Si}, \mathbf{x}_{Ci})$  where  $x_{Si}$  is actually observed in the training data for the  $i$ th observation. The other points on this curve represent extrapolations in  $\mathcal{X}$ . Marking each curve in the ICE plot at the observed point helps us assess the presence and nature of  $\hat{f}$ 's hypothesized extrapolations in  $\mathcal{X}$ .

Consider the following model:

$$Y = 10X_1^2 + \mathbf{1}_{X_2 \geq 0} + \mathcal{E}, \quad (11.7)$$

$$\mathcal{E} \stackrel{iid}{\sim} \mathcal{N}(0, .1^2), \quad \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \sim \begin{cases} \text{U}(-1, 0), \text{U}(-1, 0) & \text{w.p. } \frac{1}{3} \\ \text{U}(0, 1), \text{U}(-1, 0) & \text{w.p. } \frac{1}{3} \\ \text{U}(-1, 0), \text{U}(0, 1) & \text{w.p. } \frac{1}{3} \\ \text{U}(0, 1), \text{U}(0, 1) & \text{w.p. } 0. \end{cases}$$

Notice  $\mathbb{P}(X_1 > 0, X_2 > 0) = 0$ , leaving the quadrant  $[0, 1] \times [0, 1]$  empty. We simulate 1000 observations and fit a RF model to the data. The ICE plot for  $x_1$  is displayed in Figure 11.9a with the points corresponding to the 1000 observed  $(x_1, x_2)$

values marked by dots. We highlight observations with  $x_2 < 0$  in red and those with  $x_2 \geq 0$  in blue. The two subsets are plotted separately in Figures 11.9b and 11.9c.

The absence on the blue curves of points where both  $x_1, x_2 > 0$  confirms that the probability of  $X_1 > 0$  and  $X_2 > 0$  equals zero. From Figure 11.9c, we see that in this region of  $\mathcal{X}$ ,  $\hat{f}$  increases roughly in proportion with  $x_1^2$  even though no data exists. Ostensibly the RF model has extrapolated the polynomial relationship from the observed  $\mathcal{X}$ -space to where both  $x_1 > 0$  and  $x_2 > 0$ .

Whether it is desirable for  $\hat{f}$  to display such behavior in unknown regions of  $\mathcal{X}$  is dependent on the character of the extrapolations in conjunction with the application at hand. Moreover, different algorithms will likely give different extrapolations. Examining the ICE plots can reveal the nature of these extrapolations and guide the user to a suitable choice.

## 11.5 Real Data

We now demonstrate the ICE toolbox on three real data examples. We emphasize features of  $\hat{f}$  that might otherwise have been overlooked.

### 11.5.1 Depression Clinical Trial

The first dataset comes from a depression clinical trial (DeRubeis et al., 2013). The response variable is the Hamilton Depression Rating Scale (a common composite score of symptoms of depression where lower scores correspond to being less depressed) after 15 weeks of treatment. The treatments are placebo, cognitive therapy (a type of one-on-one counseling), and paroxetine (an anti-depressant medication). The study also collected 37 covariates which are demographic (e.g. age, gender, income) or related to the medical history of the subject (e.g. prior medications and whether the subject was previously treated). For this illustration, we drop the placebo subjects to focus

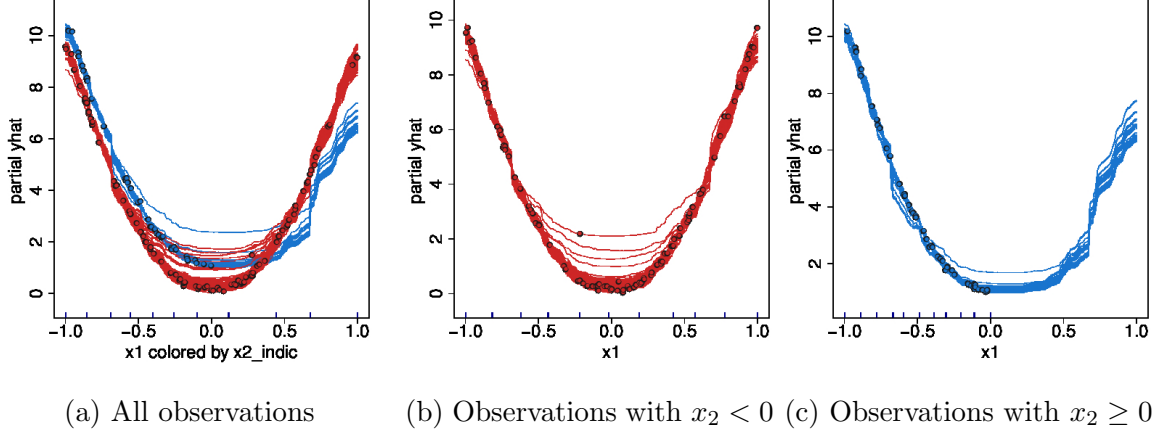


Figure 11.9: ICE plots for  $S = x_1$  of a RF model fit to Equation 11.7. The left plot shows the ICE plot for the entire dataset where  $x_2 < 0$  is colored red and  $x_2 \geq 0$  in blue. The middle plot shows only the red curves and the right only the blue. Recall that there is no training data in the quadrant  $[0, 1] \times [0, 1]$ , and so Figure 11.9c contains no points for observed values when  $x_1 > 0$  (when both  $x_1$  and  $x_2$  are positive). Nevertheless, from Figure 11.9c’s ICE curves it is apparent that the fitted values are increasing in  $x_1$  for values above 0. Here, the ICE plot elucidates the existence and nature of the RF’s extrapolation outside the observed  $\mathcal{X}$ -space.

on the 156 subjects who received either of the two active treatments.

The goal of the analysis in DeRubeis et al. (2013) is to understand how different subjects respond to different treatments, conditional on their *personal* covariates. The difference between the two active treatments, assuming the classic linear (and additive) model for treatment, was found to be statistically insignificant. If the clinician believes that the treatment effect is heterogeneous and the relationship between the covariates and response is complex, then flexible nonparametric models could be an attractive exploratory tool.

Using the ICE toolbox, one can visualize the impact of the treatment variable on an  $\hat{f}$  given by a black box algorithm. Note that extrapolations in the treatment indicator (i.e. predicting at 0 for an observed 1 or vice versa) correspond to coun-

terfactuals in a clinical setting, allowing the researcher to see how the same patient might have responded to a different treatment.

We first modeled the response as a function of the 37 covariates as well as treatment to obtain the best fit of the functional relationship using the black-box algorithm **BART** (implemented by Kapelner and Bleich, 2013) and obtained an in-sample  $R^2 \approx 0.40$ .

Figure 11.10a displays an ICE plot of the binary treatment variable, with cognitive therapy coded as “0” and paroxetine coded as “1”, colored by marital status (blue if married and red if unmarried). The plot shows a flat PDP, demonstrating no relationship between the predicted response and treatment when averaging over the effects of other covariates. However, the crossing of ICE curves indicates the presence of interactions in  $\hat{f}$ , which is confirmed by the c-ICE plot in Figure 11.10b. After centering, it becomes clear that the flat PDP obscures a complex relationship: the model predicts between -3 and +3 points on the Hamilton scale, which is a highly clinically significant range (and almost 20% of the observed response’s range). Further, we can see that **BART** fits an interaction between treatment and marital status: married subjects are generally predicted to do better on cognitive therapy and unmarried subjects are predicted to do better with paroxetine.

### 11.5.2 White Wine

The second data set concerns 5,000 white wines produced in the *vinto verde* region of Portugal obtained from the UCI repository (Bache and Lichman, 2013). The response variable is a wine quality metric, taken to be the median preference score of three blind tasters on a scale of 1-10, treated as continuous. The 11 covariates are physicochemical metrics that are commonly collected for wine quality control such as citric acid content, sulphates, etc. The model is fit with a neural network (NN) using the R package **nnet** (Venables and Ripley, 2002). We fit a NN with 3 hidden

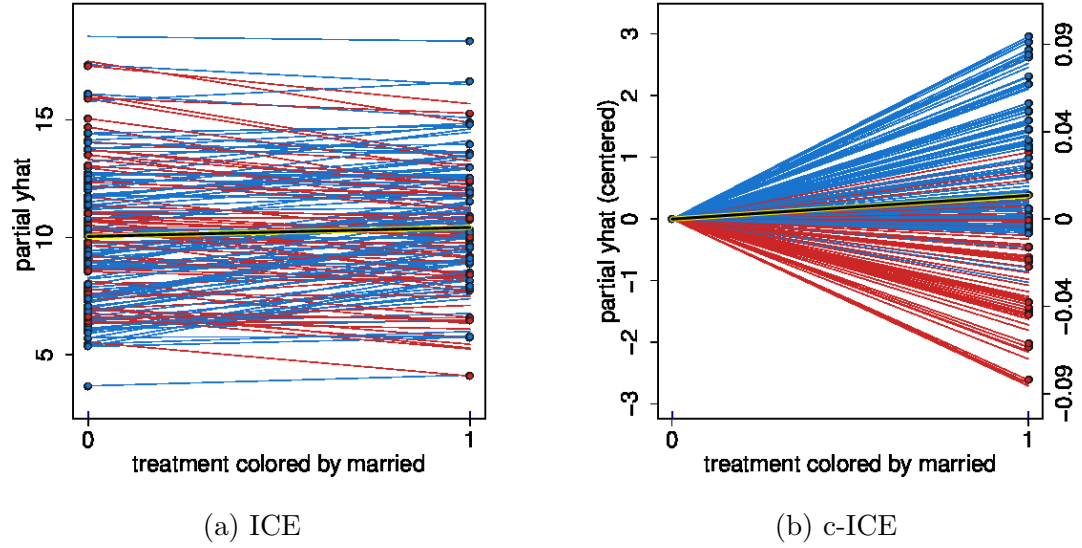


Figure 11.10: ICE plots of a BART model for the effect of treatment on depression score after 15 weeks. Married subjects are colored in blue and unmarried subjects are colored in red.

units and a small parameter value for weight decay<sup>2</sup> and achieved an in-sample  $R^2$  of approximately 0.37.

We find the covariate `pH` to be the most illustrative. The c-ICE plot is displayed in Figure 11.11a. Wines with high alcohol content are colored blue and wines with low alcohol content are colored red. Note that the PDP shows a linear trend, indicating that on average, higher `pH` is associated with higher fitted preference scores. While this is the general trend for wines with higher alcohol content, the ICE plots reveal that interaction effects are present in  $\hat{f}$ . For many white wines with low alcohol content, the illustration suggests a nonlinear and cumulatively *negative* association. For these wines, the predicted preference score is actually negatively associated with `pH` for low values of `pH` and then begins to increase — a severe departure from what the PDP suggests. However, the area of increase contains no data points, signifying

---

<sup>2</sup>Note that NN models are highly sensitive to the number of hidden units and weight decay parameter. We therefore offer the following results as merely representative of the type of plots which NN models can generate.

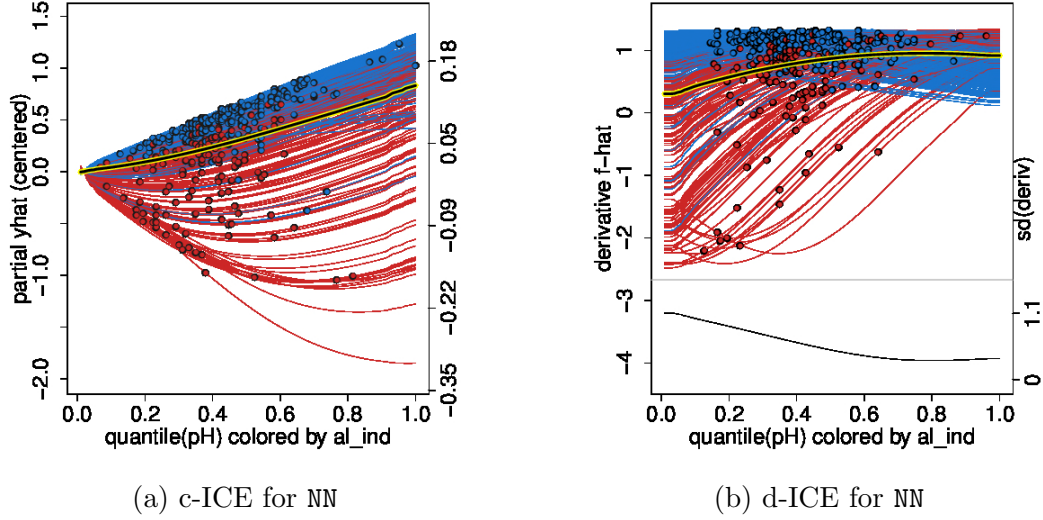


Figure 11.11: ICE plots of NN model for wine ratings versus pH of white wine colored by whether the alcohol content is high (blue) or low (red). To prevent cluttering, only a fraction of the 5,000 observations are plotted.

that the increase is merely an extrapolation likely driven by the positive trend of the high alcohol wines. Overall, the ICE plots indicate that for more alcoholic wines, the predicted score is increasing in pH while the opposite is true for wines with low alcohol content. Also, the difference in cumulative effect is meaningful; when varied from the minimum to maximum values of pH, white wine scores vary by roughly 40% of the range of the response variable.

Examining the derivative plot of Figure 11.11b confirms the observations made above. The NN model suggests interactions exist for lower values of pH in particular. Wines with high alcohol content have mostly positive derivatives while those with low alcohol content have mostly negative derivatives. As pH increases, the standard deviation of the derivatives decreases, suggesting that interactions are less prevalent at higher levels of pH.



### 11.5.3 Diabetes Classification in Pima Indians

The last dataset consists of 332 Pima Indians (Smith and Everhart, 1988) obtained from the R library `MASS`. Of the 332 subjects, 109 were diagnosed with diabetes, the binary response variable which was fit using seven predictors (with body metrics such as blood pressure, glucose concentration, etc.). We model the data using a RF and achieve an out-of-bag misclassification rate of 22%.

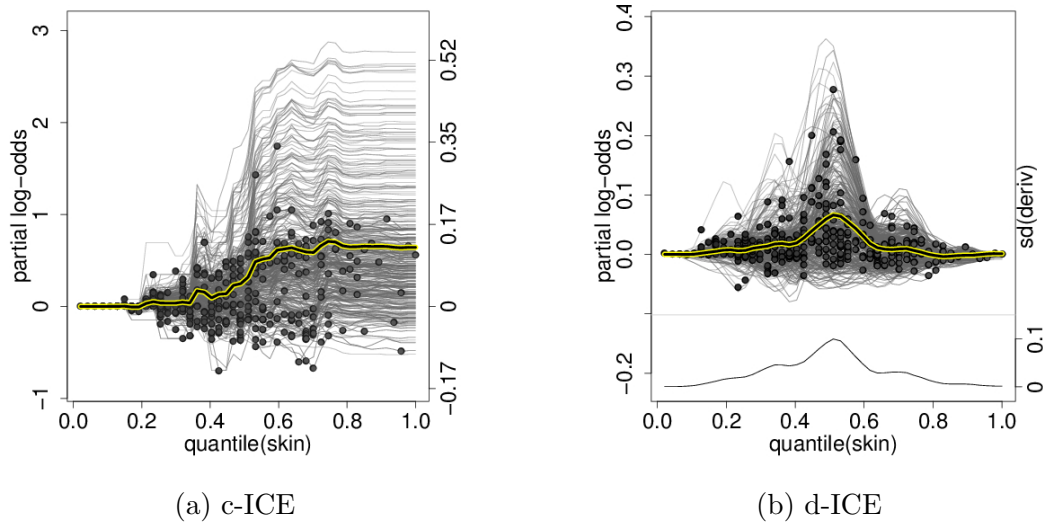


Figure 11.12: ICE plots of a RF model for estimated centered logit of the probability of contracting diabetes versus `skin` colored by subject `age`.

Once again, ICE plots offer the practitioner a more comprehensive view of the output of the black box. For example, the covariate `skin` thickness about the triceps is plotted as a c-ICE in Figure 11.12a. The PDP clearly shows an increase in the predicted centered log odds of contracting diabetes. This is expected given that `skin` is a proxy for obesity, a major risk factor for diabetes. However, the ICE plot illustrates a more elaborate model fit. Many subjects with high `skin` have a flat risk of diabetes according to  $\hat{f}$ ; others with comparable thickness exhibit a much larger centered log-odds increase.<sup>3</sup> Figure 11.12b shows that the RF model fits interactions

<sup>3</sup>The curves at the top of the figure mainly correspond to younger people. Their estimated effect

across the range of `skin` with the largest heterogeneity in effect occurring when `skin` is slightly above 30. This can be seen in the standard deviation of the derivative in the bottom pane of Figure 11.12b.

## 11.6 A Visual Test for Additivity

Thus far we have used the ICE toolbox to explore the output of black box models. We have explored whether  $\hat{f}$  has additive structure or if interactions exist, and also examined  $\hat{f}$ 's extrapolations in  $\mathcal{X}$ -space. To better visualize interactions, we plotted individual curves in colors according to the value of a second predictor  $x_k$ . We have *not* asked whether these findings are reflective of phenomena in any underlying model.

When heterogeneity in ICE plots is observed, the researcher can adopt two mindsets. When one considers  $\hat{f}$  to be the fitted model used for subsequent predictions, the heterogeneity is of interest because it determines future fitted values. This is the mindset we have considered thus far. Separately, it might be interesting to ascertain whether interactions between  $x_S$  and  $\mathbf{x}_C$  exist in the data generating model, denoted  $f$ . This question exists for other discoveries made using ICE plots, but we focus here on interactions.

The problem of assessing the statistical validity of discoveries made by examining plots is addressed in Buja et al. (2009) and Wickham et al. (2010). The central idea in these papers is to insert the observed plot randomly into a lineup of null plots generated from data sampled under a null distribution. If the single real plot is correctly identified amongst 19 null plots, for example, then “the discovery can be assigned a  $p$ -value of 0.05” (Buja et al., 2009). A benefit of this approach is that the procedure is valid despite the fact that we have not specified the form of the alternative distribution — the simple instruction “find the plot that appears different”

---

of high thickness is seen to be an extrapolation.

is sufficient.

### 11.6.1 Procedure

We adapt this framework to the specific problem of using ICE plots to evaluate additivity in a statistically rigorous manner. For the exposition in this section, suppose that the response  $y$  is continuous, the covariates  $\mathbf{x}$  are fixed, and  $y = f(\mathbf{x}) + \mathcal{E}$ . Further assume  $\mathbb{E}[\mathcal{E}] = 0$  and

$$f(\mathbf{x}) = g(x_S) + h(\mathbf{x}_C), \quad (11.8)$$

meaning the true  $\mathbf{x}$ -conditional expectation of  $y$  is additive in functions of  $x_S$  and  $\mathbf{x}_C$ . Let  $F$  be the distribution of  $\hat{f}$  when Equation 11.8 holds and  $f$  is additive. We wish to test  $H_0: \hat{f} \sim F$  versus  $H_a: H_0$  is false.

Recall that ICE plots displaying non-parallel curves suggest that  $\hat{f}$  is not additive in functions of  $x_S$  and  $\mathbf{x}_C$ . Thus if we can correctly identify a plot displaying such features amongst  $K - 1$  null plots generated under  $F$ , the discovery is valid at  $\alpha = 1/K$ .

We sample from  $F$  by using backfitting (Breiman and Friedman, 1985) to generate  $g^*$  and  $h^*$ , estimates of  $g$  and  $h$ , and then bootstrapping the residuals. Both  $g^*$  and  $h^*$  can be obtained via any supervised learning procedures. The general procedure for  $|S| = 1$  proceeds is as follows.

- 1 Using backfitting, obtain  $g^*$  and  $h^*$ . Then compute a vector of fitted values  $\hat{\mathbf{y}}^* = g^*(x_S) + h^*(\mathbf{x}_C)$  and a vector of residuals  $\mathbf{r}^* := y - \hat{\mathbf{y}}^*$ .
- 2 Let  $\mathbf{r}_b$  be a random resampling of  $\mathbf{r}^*$ . If heteroscedasticity is of concern, one can keep  $\mathbf{r}^*$ 's absolute values fixed and let  $\mathbf{r}_b$  be a permutation of  $\mathbf{r}^*$ 's signs. Define  $\mathbf{y}_b := \hat{\mathbf{y}}^* + \mathbf{r}_b$ . Note that  $\mathbb{E}[\mathbf{y}_b \mid \mathbf{x}]$  is additive in  $g^*(x_S)$  and  $h^*(\mathbf{x}_C)$ .

- 3 Fit  $\mathbf{y}_b$  to  $\mathbf{X}$  using the same learning algorithm that generated the original ICE (c-ICE or d-ICE) plot to produce  $\hat{f}_b$ . This yields a potentially non-additive approximation to null data generated using an additive model.
- 4 Display an ICE (or c-ICE or d-ICE) plot for  $\hat{f}_b$ . Deviations from additivity observed in this plot must be due to sources other than interactions between  $x_S$  and  $\mathbf{x}_C$  in the underlying data.
- 5 Repeat steps (2) - (4)  $K - 1$  times, then randomly insert the true plot amongst these  $K - 1$  null plots.
- 6 If the viewer can correctly identify the true plot amongst all  $K$  plots, the discovery is valid for level  $\alpha = 1/K$ . Note that the discovery is conditional on the procedures for generating  $g^*$  and  $h^*$ .

### 11.6.2 Examples

An application of this visual test where  $g$  is taken to be the “supersmoother” (Friedman, 1984) and  $h$  is a BART model is illustrated using the depression data of Section 11.5.1. We sample  $\mathbf{r}_b$  by permuting signs. The data analyst might be curious if the ICE plot is consistent with the treatment being additive in the model. We employ the additivity lineup test in Figure 11.13 using 20 images. We reject the null hypothesis of additivity of the treatment effect at  $\alpha = 1/20 = 0.05$  since the true plot (row 2, column 2) is clearly identifiable. This procedure can be a useful test in clinical settings when the treatment effect is commonly considered linear and additive and can alert the practitioner that interactions should be investigated.

Another application of this visual test where  $g$  is taken to be the supersmoother and  $h$  is a NN model is illustrated using the wine data of Section 11.5.2. Here again we sample  $\mathbf{r}_b$  by permuting signs. The data analyst may want to know if the fitted model

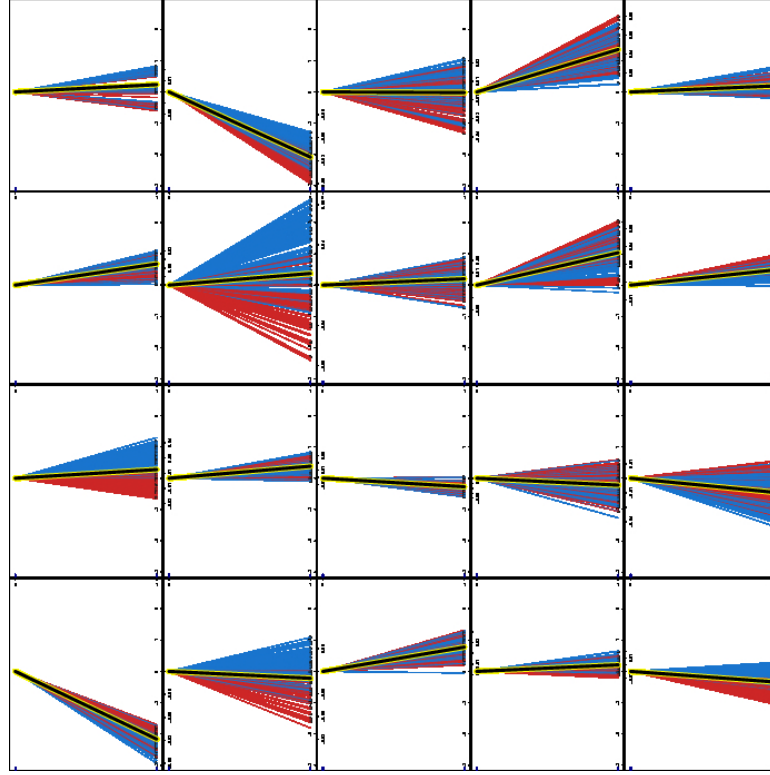


Figure 11.13: Additivity lineup test for the predictor `treatment` in the depression clinical trial dataset of Section 11.5.1.

is suggestive of interactions between `pH` and the remaining features in the underlying model. We employ the additivity lineup test in Figure 11.14, again using 20 images.

Looking closely one sees that the first and third plots in the last row have the largest range of cumulative effects and exhibit more curvature in individual curves than most of the other plots, making them the most extreme violations of the null. Readers that singled out the first plot in the last row would have a valid discovery at  $\alpha = .05$ , but clearly the evidence of non-additivity is much weaker here than in the previous example. Whereas Figure 11.13 suggests the real plot is identifiable amongst more than 20 images, it would be easy to confuse Figure 11.14's true plot with the one in row 4, column 3. Hence there is only modest evidence that `pH`'s impact on  $\hat{f}$  is different from what a NN might generate if there were no interactions between `pH` and the other predictors.

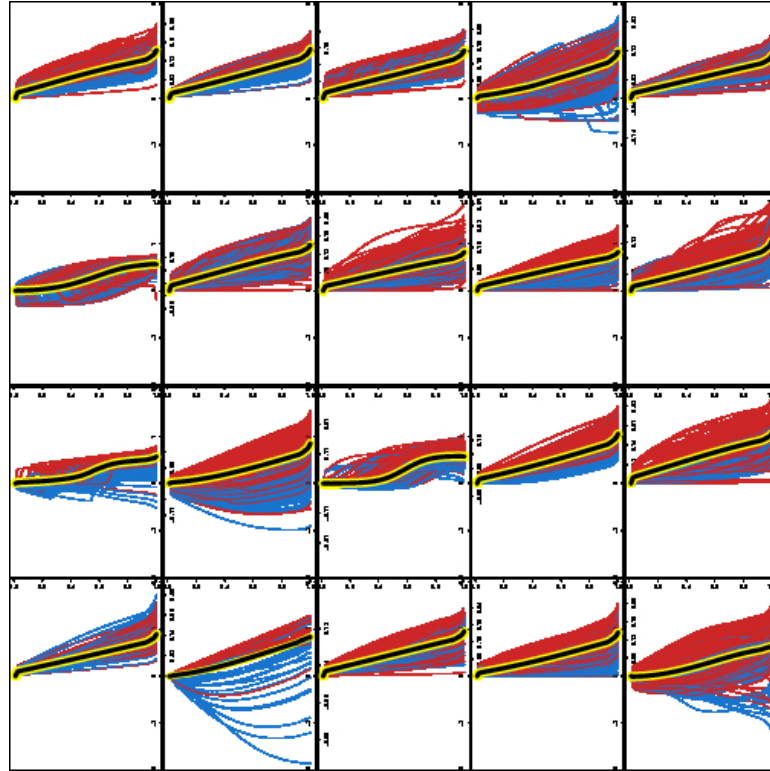


Figure 11.14: Additivity lineup test for the predictor pH in the white wine dataset of Section 11.5.2.

## 11.7 Discussion

We developed a suite of tools for visualizing the fitted values generated by an arbitrary supervised learning procedure. Our work extends the classical partial dependence plot (PDP), which has rightfully become a very popular visualization tool for black-box machine learning output. The partial functional relationship, however, often varies conditionally on the values of the other variables. The PDP offers the average of these relationships and thus individual conditional relationships are consequently masked, unseen by the researcher. These individual conditional relationships can now be visualized, giving researchers additional insight into how a given black box learning algorithm makes use of covariates to generate predictions.

The ICE plot, our primary innovation, plots an entire distribution of individual

conditional expectation functions for a variable  $x_S$ . Through simulations and real data examples, we illustrated much of what can be learned about the estimated model  $\hat{f}$  with the help of ICE. For instance, when the remaining features  $\mathbf{x}_C$  do not influence the association between  $x_S$  and  $\hat{f}$ , all ICE curves lie on top of another. When  $\hat{f}$  is additive in functions of  $\mathbf{x}_C$  and  $x_S$ , the curves lie parallel to each other. And when the partial effect of  $x_S$  on  $\hat{f}$  is influenced by  $\mathbf{x}_C$ , the curves will differ from each other in shape. Additionally, by marking each curve at the  $x_S$  value observed in the training data, one can better understand  $\hat{f}$ 's extrapolations. Sometimes these properties are more easily distinguished in the complementary “centered ICE” (c-ICE) and “derivative ICE” (d-ICE) plots. In sum, the suite of ICE plots provides a tool for visualizing an arbitrary fitted model’s map between predictors and predicted values.

The ICE suite has a number of possible uses that were not explored in this work. While we illustrate ICE plots using the same data as was used to fit  $\hat{f}$ , out-of-sample ICE plots could also be valuable. For instance, ICE plots generated from random vectors in  $\mathbb{R}^p$  can be used to explore other parts of  $\mathcal{X}$  space, an idea advocated by Plate et al. (2000). Further, for a single out-of-sample observation, plotting an ICE curve for each predictor can illustrate the sensitivity of the fitted value to changes in each predictor for this particular observation, which is the goal of the “contribution plots” of Strumbelj and Kononenko (2011). Additionally, investigating ICE plots from  $\hat{f}$ 's produced by multiple statistical learning algorithms can help the researcher compare models. Exploring other functionality offered by the ICEbox package, such as the ability to cluster ICE curves, is similarly left for subsequent research.

The tools summarized thus far pertain to *exploratory* analysis. Many times the ICE toolbox provides evidence of interactions, but how does this evidence compare to what these plots would have looked like if no interactions existed? Section 11.6 proposed a *testing* methodology. By generating additive models from a null distri-

bution and introducing the actual ICE plot into the lineup, interaction effects can be distinguished from noise, providing a test at a known level of significance. Future work will extend the testing methodology to other null hypotheses of interest.

## Supplementary Materials

The procedures outlined in Section 11.3 are implemented in the R package `ICEbox` available on `CRAN`. Simulated results, tables, and figures specific to this paper can be replicated via the script included in the supplementary materials and located at <https://github.com/kapelner/ICEbox>. The depression data of Section 11.5.1 cannot be released due to privacy concerns.

## Acknowledgements

We thank Richard Berk for insightful comments on multiple drafts and suggesting color overloading. We thank Andreas Buja for helping conceive the testing methodology. We thank Abba Krieger for his helpful suggestions. We also wish to thank Zachary Cohen for the depression data of Section 11.5.1 and helpful comments. Alex Goldstein acknowledges support from the Simons Foundation Autism Research Initiative. Adam Kapelner acknowledges support from the National Science Foundation's Graduate Research Fellowship Program.



## Detecting Heterogeneous Effects via Crowdsourcing

By Adam Kapelner

### Abstract

We attempt to detect heterogeneous treatment effects between an experimental population that chooses to be in a study versus those who are randomly assigned to a study. We call this “selection-into-experiment bias” and we test its effects in three experiments run on Amazon Mechanical Turk (MTurk). We find no differential average treatment effects in the studies, but we were also underpowered to do so. We feel that external validity is not a concern when making inference to the MTurk population at large for the studies illustrated here.

### 12.1 Introduction

Imagine a typical study with subjects randomized into treatment and control. Assuming a linear model,

$$\mathbf{Y} = \beta_0 + \beta_T \mathbf{1}_T + \boldsymbol{\mathcal{E}}, \quad \boldsymbol{\mathcal{E}} \sim \mathcal{N}_n(0, \sigma^2 \mathbf{I}_n)$$

the investigator can estimate the “average treatment effect,”  $\beta_T$  and they’ll most likely write in their paper that:

“We found on average,  $b_T = 1.23$  with a  $p_{\text{val}} = 0.0456$  and **thus we recommend treatment  $T$  to the general population.**”

The text in red above may be over ambitious. The investigators are essentially vouching for their results’ *external validity* which means the truth of their results in this limited experimental setting can be extended to some larger population of interest. Pictured in figure 12.1, the population of interest is usually humanity at large and the sample is ususally a convenience sample and the context of the study is a laboratory.

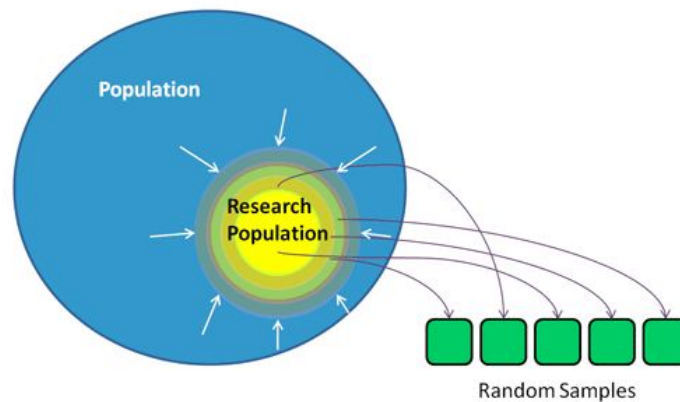


Figure 12.1: Experimental samples are drawn from the “research population” which may be different than the whole population of interest.

External validity is a well-known problem; it’s a fundamental problem in RCT inference and has been noted since the times of Fisher. It is well discussed in many fields: statistics, philosophy, political science, economics, and medicine.

An article in the Lancet (Rothwell, 2005) discusses how RCT’s were designed to have great internal validity, but the pioneers knew about the problem of applicability / generalizability, a problem which has now seemingly been forgotten. He bemoans

the ignorance and lack of guidelines for researchers and sites interesting studies where the treatment effect can vary even by small changes in latitude of the patient.

A recent article (Druckman and Kam, 2011) discusses how using undergraduates as subjects may not be a problem for generalizing experimental results; they argue that undergraduates in fact *do* reflect the American population at-large. Of course they don't speak about the *additional* selection problem that undergraduates are generally paid or given course credit, and those that participate are not even representative of their peers. The problem is open to speculation.

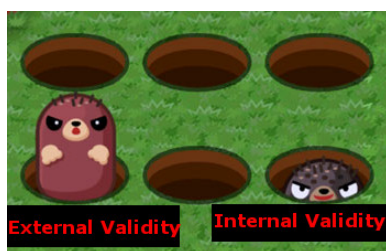


Figure 12.2: External validity can be had at the expense of internal validity and vice versa.

Philosophically, Cartwright (2007) argues that wringing truth out of nature is sort of like an arcade game of whack-a-mole illustrated in figure 12.2. “Clinching” methods, such as RCTs and theory, prove the conclusion but are narrow in scope; “vouching” methods, such as qualitative comparative analysis hint at the conclusion without unequivocal proof, but are more broader in application.

### 12.1.1 Heterogeneous Effects

External validity is suspect when it can be proven that there exists both:

- (a) *selection bias* into the study itself, meaning that the “research population” is different than the “total population” on one or many demographic variables,  $\mathbf{X} := X_1, \dots, X_p$ .

(b) *an interaction* between treatment and demographic variables.

If the true model were as follows, selection bias into the study itself would be unproblematic.

$$\mathbf{Y} = \beta_T \mathbf{1}_T + f(\mathbf{X}) + \mathcal{E}$$

As we move from the research population to the true population, the  $f$  function will change, but the ATE,  $\mathbb{E}[Y \mid \mathbf{1}_T = 1] - \mathbb{E}[Y \mid \mathbf{1}_T = 0] = \beta_T$ , would remain unchanged.

However, if there are interactions between the treatment and the demographic variables,

$$\mathbf{Y} = \beta_T \mathbf{1}_T + f(\mathbf{1}_T, \mathbf{X}) + \mathcal{E}$$

there is no longer an “average treatment effect” unless we either slice the population on fixed levels of  $\mathbf{X}$  or expectorate over the population covariate distribution, since:

$$\mathbb{E}[Y \mid \mathbf{1}_T = 1] - \mathbb{E}[Y \mid \mathbf{1}_T = 0] = \beta_T + f(1, \mathbf{X}) - f(0, \mathbf{X})$$

whose behavior as subjects move around  $\mathbf{X}$  space may vary considerably due to the unknown properties of the  $f$  function.

### 12.1.2 Goals of the Study and Outline of this Report

This study attempts to detect if external validity is compromised in *behavioral* studies which involve text manipulation run on Amazon’s Mechanical Turk (MTurk). We pick

three studies that are designed to elicit cognitive biases. The first measures the price of beer between two groups that experience differential framing from a luxury hotel or a run-down grocery store. The second measures whether people decide to cooperate with one another after being primed with religious text. The third measures the sunk cost fallacy, we ask whether or not people are likely to go to a theatre event after paying dearly for tickets or being given them for free.

We measure if there is a difference if people are administered a study at random or if they are allowed to select which study they want to participate in when they are told the general idea of the study. This “selection” wing is designed to create different research populations for each study.

For the reader unfamiliar with MTurk and/or natural field experiments, we suggest reading Chandler and Kapelner (2013) where it is described how one major external validity concern that is manifest in most studies, laboratory bias, is *not* characteristic of carefully designed MTurk experiments. We explain the experiment in detail in section 13.2, analyze the data in 12.3, and conclude in section 12.4.

## 12.2 Experimental Methods and Design

We create a HIT on MTurk paying \$0.05 posing as a major marketing company asking for consumer opinions (see figure 12.3). Subjects were not told they were part of a study in order to create a real-effort task in a *natural field experiment*. We limited experimental subjects to be from America because the experimental manipulations relied on fluency in English.

Once the HIT was accepted, we move them to a demographic survey page pictured in figure 12.4 (for a list of questions asked, refer to appendix A.6.1). To avoid any systematic bias, we randomized the order of these questions as well as their answer choices (within reason).

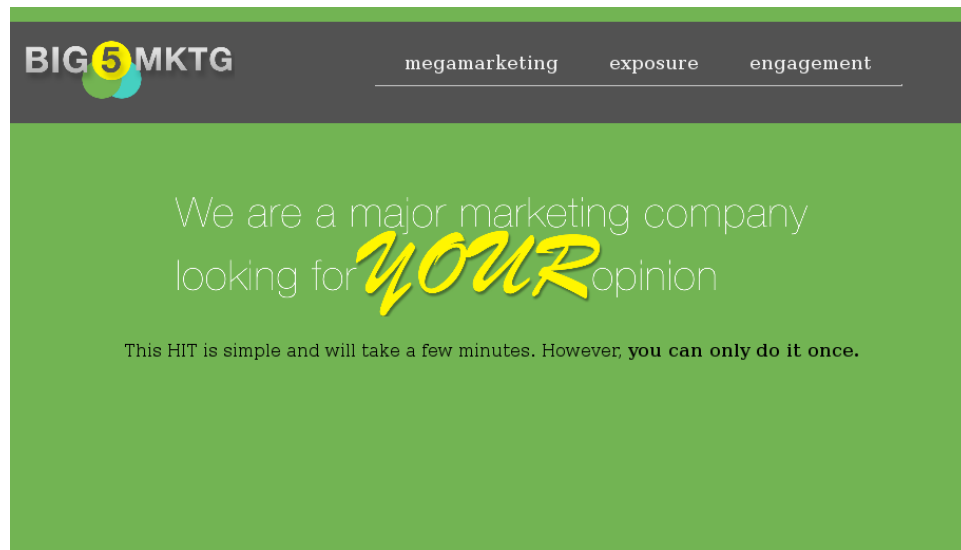


Figure 12.3: The splash page shown to a Turker viewing our HIT for the first time. The Turker will make their decision whether or not to participate in the task based on this screen and the wage offered.

We first need to know a bit about you.

1. What is your gender?
  - ☐ Male
  - ☐ Female
2. I am more...
  - ☐ Friendly and compassionate
  - ☐ Logical and cold
3. Were you born in the United States?
  - ☐ No
  - ☐ Yes
4. Do you prefer your life to be filled with puzzles you must solve?
  - ☐ Yes
  - ☐ No
5. How many children do you have?
  - ☐ 0
  - ☐ 1
  - ☐ 2
  - ☐ 3
  - ☐ 4

Figure 12.4: Part of the survey page which collects demographic covariates about the subjects.

Upon completion of this survey, they were randomized into the “random study” wing, which we’ll call “R,” or the “select study” wing which we’ll denote “S.”<sup>1</sup> If they were placed into the “R” wing, they were then randomized into one of the three studies described in the next subsection. If they were placed into the “S” wing, the subjects were allowed to choose which study they wanted to participate in. This selection is described more fully in a following subsection. Once they were assigned one of the three studies, they were further randomized evenly between treatment and control particular to the study.<sup>2</sup> The experimental flowchart illustrating this description is pictured in figure 12.5.

### 12.2.1 The Three Studies

We are trying to detect sampling bias into experiments on MTurk for behavioral experiments whose experimental manipulations involve text. To make any general claim, studies would ideally be sampled from the space of all possible studies that fit this criteria.

Our initial thought would be to sample from cognitive bias studies, of which there are many, and they are currently of great interest to behavioral scientists. We selected three studies investigating three different cognitive biases: framing, priming, and sunk cost. Each of the studies’ experimental manipulation consisted of *subtle* text changes between the control and the treatment. We briefly describe the studies below. Full text of the experimental screenst can be found in appendix A.6.3.

---

<sup>1</sup>For this randomization, we used complete randomization with a standard bernoulli. This would prevent any systematic bias in the order in which the participants showed up but would most likely result in imbalance. Since we planned on collecting a large number of subjects, an imbalance here was not a concern.

<sup>2</sup>Once again, we employed complete randomization. We admittedly should’ve used a block design here to obtain higher power because any bias from time admitted is most likely taken care of by the R/S randomization.

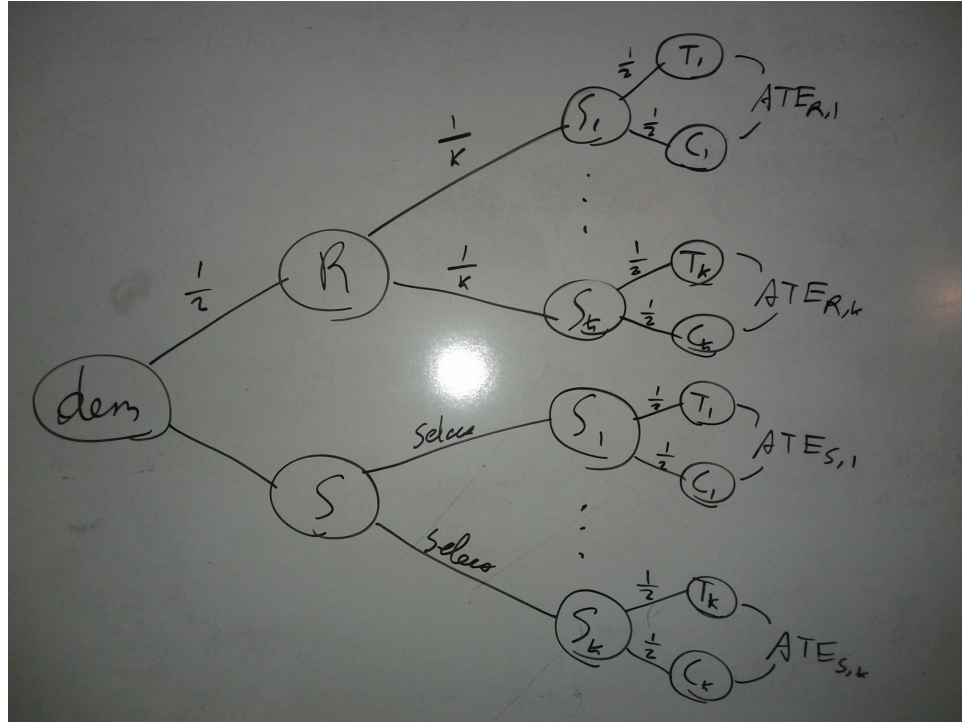


Figure 12.5: The experimental flow: wings and randomizations. The above illustration is for any number of studies. Here  $k = 3$ .

People are known to differentially pay higher or lower prices for identical items based on how the item is “framed.” We take a verbatim study from Thaler (1985) where participants are asked how much they would pay for a beer which comes from either a fancy resort hotel (treatment) or a run-down grocery store (control). The outcome measure would be a real number greater than zero.

It is known that people differentially change their behavior if they were “primed” before with religious ideas. We test whether subjects differentially choose to cooperate in a prisoner’s dilemma if they were given an excerpt from the bible about charity (treatment) or an excerpt of an encyclopedia page about fish (control). This experiment is a replication of one found in Horton and Chilton (2010) with minor modifications. The outcome measure is cooperate or not cooperate.

People are also known to differentially make decisions to consume if based on a cost that has previously been sunk. According to economic theory, the cost already sunk



should not affect the amount of utility, or enjoyment, one should get from a product. We design a study where the subject is hypothetically going to a theatre production on a night with extreme weather. We test whether the subjects are more likely to indicate a strong desire to go when they paid dearly for the tickets (treatment) or received tickets for free (control). The response was measured on a Likert scale from zero to nine.

Since the studies rely on subtly text manipulations, we employ the anti-satisficing technique of fading in each word of the experiment text at a high school reading page to ensure respondents pay more attention to the text (Kapelner and Chandler, 2010).

### 12.2.2 The Selection Wing

How are subjects to choose which study to participate in when they are randomized into the “S” wing? The purpose of the experiment is to detect selection bias into the experiments; thus, we ideally want to bias the subjects *as much as possible* into their study of preference to create the most conspicuous heterogeneous effects. We attempt to do this *without* giving away what the study is trying to measure (which would raise concerns about the studies’ internal validity).

We chose to give subjects the screen shown in figure 12.6. They know the marketing question-of-interest has something to do with beer, religion, or theatre, but they don’t know what the treatment manipulations are. This can create heterogeneous effects in two ways (1) having knowledge about the question at-hand may bias their responses and induce a different ATE vis-a-vis subjects who did not have prior knowledge (2) subjects who are more interested in “beer” may represent a different subpopulation than those who are interested in “religion” and likewise those who are interested in “theatre.”



Figure 12.6: The screen shown to subjects in the “S” wing immediately after completing the demographic survey.

### 12.2.3 Formal Experimental Design

We are trying to demonstrate heterogeneous effects which amount to a differential ATE between the “R” and “S” wings. This can be tested via the following model (its design matrix is illustrated in figure 12.7). For study  $k$ ,

$$\mathbf{Y}_k = \mathbf{X}\boldsymbol{\beta} + \beta_{T,k}\mathbf{1}_{T_k} + \beta_{S,k}\mathbf{1}_S + \beta_{ST,k}\mathbf{1}_{T_k}\mathbf{1}_S + \boldsymbol{\varepsilon}, \quad k \in \{1, 2, 3\}$$

The  $\boldsymbol{\beta}$  represents the nuisance parameters for the slopes of the demographic variables. The  $\beta_{T,k}$ ’s are the ATE’s for each study in the “R” wing which is of ancillary interest — we only wish to check these to ensure the study worked as designed to arrive at an estimate to the causal effect of the manipulations. The  $\beta_{S,k}$ ’s are offsets for belonging to the “S” wing. Being allowed to select the study may shift your level

$$\mathbf{X} = \begin{bmatrix} X_{\cdot 1} & \dots & X_{\cdot p} & \mathbf{1}_S & \mathbf{1}_T & \mathbf{1}_S \mathbf{1}_T \\ \vdots & \dots & \vdots & \mathbf{1}_{n_S} & \mathbf{1}_{n_{ST}} & \mathbf{1}_{n_{ST}} \\ \vdots & \dots & \vdots & \vdots & \mathbf{0}_{n_S - n_{ST}} & \mathbf{0}_{n_S - n_{ST}} \\ \vdots & \dots & \vdots & \mathbf{0}_{N - n_S} & \mathbf{1}_{n_{RT}} & \mathbf{0}_{N - n_S} \\ \vdots & \dots & \vdots & \vdots & \mathbf{0}_{N - n_S - n_{RT}} & \vdots \end{bmatrix}$$

Figure 12.7: The design matrix for each of the  $k$  studies.

of response; this is interesting to check, but still ancillary to our question of interest. The  $\beta_{ST,k}$  are the parameters of interest. They model the differential ATE between the “R” and “S” wings which is the average heterogeneous effect induced by having two different research populations participate in the same study. Based on our hypothesis, we wish to reject these being zero:

$$H_0 : \beta_{ST,1} = 0, \quad H_0 : \beta_{ST,2} = 0, \quad H_0 : \beta_{ST,3} = 0$$

For each of the studies, we have a  $2 \times 2$  factorial design with  $n_S$  random (thus it cannot be balanced).

#### 12.2.4 Pilot Study to Estimate Power

We first ran a pilot study of  $N = 100$  to inform ourselves about power for the full study. For the framing study,  $Y$  is continuous, but for the sunk cost study,  $Y$  is ordinal, and for the priming study,  $Y$  is dichotomous, so the following calculation is for the framing effect study at  $\alpha = 5\%$  since power calculations for non-continuous responses are more difficult.

We estimate MSE of the pilot study regression to be 11.42 and we assume this to

be our variance of the experimental error term going forward.<sup>3</sup> We wish to calculate power for many possible values of  $\beta_{ST}$  which is the interaction effect in our factorial design. Letting  $r$  be the number of duplicates in each wing-treatment cell,  $N$  be the total sample size for the study, power can be calculated following Morris (2011, page 159):

$$F^* := F(95\%, 1, 4(r-1)), \quad W \sim F\left(1, 4(r-1), \frac{4r\beta_{ST}^2}{\sigma^2}\right), \quad \text{POW} := \mathbb{P}(W > F^*)$$

Power for a range of possible  $r$  and  $\beta_{ST}$  values appears below in table 12.1.

$N_{\text{study}}$	$r$	$\beta_{ST}$							
		0.01	0.05	0.1	0.15	0.2	0.25	0.5	0.75
400	100	0.05	0.06	0.09	0.14	0.22	0.31	<b>0.84</b>	<b>0.99</b>
800	200	0.05	0.07	0.13	0.24	0.39	0.55	<b>0.99</b>	<b>1.00</b>
1200	300	0.05	0.08	0.18	0.34	0.53	0.73	<b>1.00</b>	<b>1.00</b>
1600	400	0.05	0.09	0.22	0.43	0.66	<b>0.84</b>	<b>1.00</b>	<b>1.00</b>
2000	500	0.05	0.10	0.26	0.51	0.75	<b>0.91</b>	<b>1.00</b>	<b>1.00</b>
2400	600	0.05	0.11	0.30	0.58	<b>0.83</b>	<b>0.95</b>	<b>1.00</b>	<b>1.00</b>
2800	700	0.05	0.12	0.35	0.65	<b>0.88</b>	<b>0.97</b>	<b>1.00</b>	<b>1.00</b>

Table 12.1: Power by number of duplicates and interaction effect size.

## 12.3 Results

We hired 2112 total Turkers as subjects with total cost of about \$116. Sample sizes are reported for each wing and treatment cell for all studies in table 12.2. Note that

---

<sup>3</sup>The variance found after the experiment was complete was around 14, so this was a good estimate.

$r$ , the cell size, varies between 140 and 214.

Arm	Study	Treatment	Control	Total	
R	Framing	169	172	341	
	Priming	171	176	347	1053
	Sunk Cost	199	166	365	
S	Framing	167	145	312	
	Priming	203	214	417	1059
	Sunk Cost	169	161	330	

Table 12.2: Experimental sample sizes.

We analyzed data from all studies using ordinary least squares<sup>4</sup> to estimate parameters noted in figure 12.7. The nuisance parameters include the demographic variables discussed in appendix A.6.1 as well as other experimental covariates we collected from subject behavior which we describe in appendix A.6.2. We summarize our main results in table 12.3, results for the framing study in figure 12.8, results for the priming study in figure 12.9, and results for the sunk cost study in figure 12.10.

	Framing Study			Priming Study			Sunk Cost Study		
	$b_T$	$s_{b_T}$	$p_{val}$	$b_T$	$s_{b_T}$	$p_{val}$	$b_T$	$s_{b_T}$	$p_{val}$
ATE in “R” wing	0.69	0.43	0.107	0.06	0.03	0.078	0.19	0.29	0.507
Offset for “S” wing	0.32	0.44	0.475	-0.01	0.03	0.662	-0.20	0.31	0.525
Heterogeneous Effect	-0.19	0.61	0.752	-0.04	0.04	0.369	-0.06	0.42	0.891

Table 12.3: Experimental estimates of ATE, selection offset, and average heterogeneous effects for all studies.

<sup>4</sup>Technically, this is inappropriate for the priming study which has dichotomous response and the sunk cost study which has ordinal response. Since the data turned out too noisy for effect detection, we did not try other models because we feel they would have been equally unsuccessful.

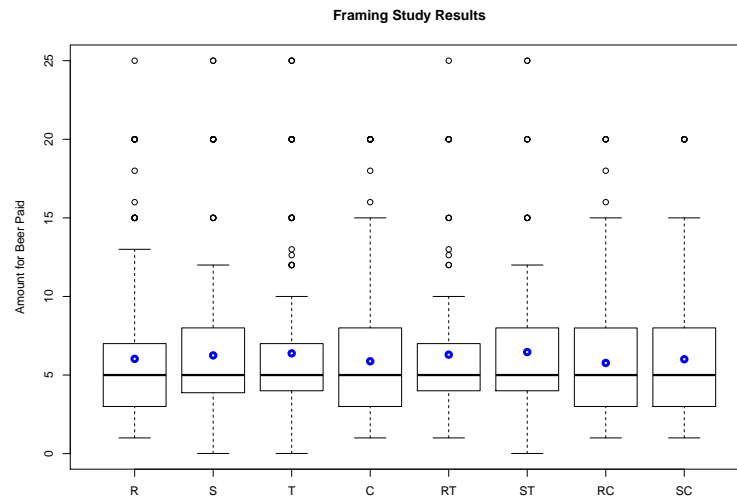


Figure 12.8: Box and whisker plots for the framing study by row, column, and cell in the factorial design. Blue dots indicate sample averages.

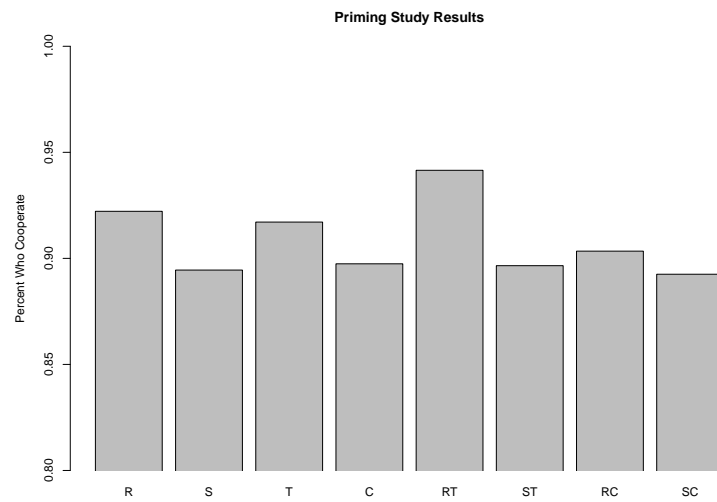


Figure 12.9: Sample proportions of cooperate in the priming study by row, column, and cell in the factorial design.

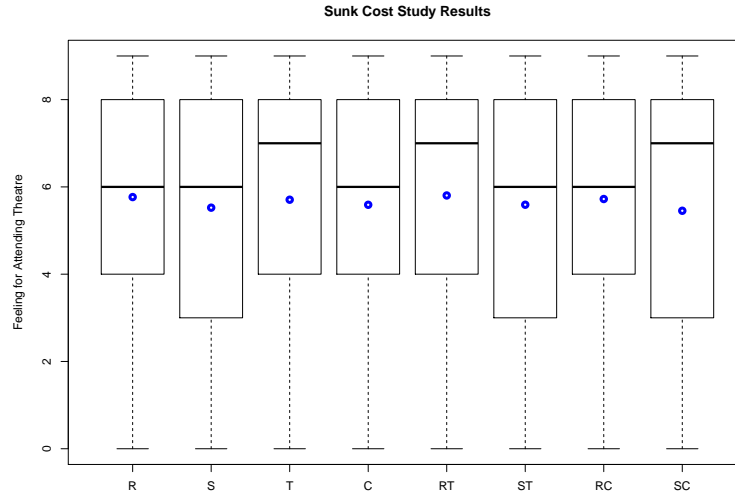


Figure 12.10: Box and whisker plots for the sunk cost study by row, column, and cell in the factorial design. Blue dots indicate sample averages.

In short, we did not detect significant heterogeneous effects in any of the three studies. Referencing table 12.1, we note that we only have sufficient power to detect interaction effects above \$0.50 in the framing study.

## 12.4 Conclusions and Future Directions

We ran a natural field experiment on MTurk to detect selection-into-experiment bias for three behavioral studies. We did so by placing half of the participants into a random study and allowing the other half to pick the study they wished to complete. We then tried to measure selection-into-experiment bias by calculating differential ATE estimates between the random study participants and the select-your-own study participants.

We did not find significant heterogeneous treatment effects for any of the studies between the two populations either because they don't exist or because we were underpowered to find them. If we were to make a recommendation to investigators

using MTurk to study behavioral phenomena using experiments that involve text manipulations, we would still recommend to not let them know anything about the study, but if they had to inform the participants it most likely would not affect the results.

We would like to run the study again with better design. First, each of the three studies should be designed to elicit a larger ATE. This would allow more room for heterogeneous effects. Second, we should employ techniques to reduce noise in the data. We are currently developing sequential matching methods for this very purpose. Third, the selection mechanism should be better designed.

## Acknowledgements

Thanks to Dean Foster, Abba Krieger, Mary Putt and Paul Rosenbaum for helpful discussions.



## Matching on-the-fly in Sequential Experiments

By Adam Kapelner and Abba Krieger

### Abstract

We propose a dynamic allocation procedure that increases power and efficiency when measuring an average treatment effect in fixed sample size sequential randomized trials. Subjects arrive iteratively and are either randomized *or* paired via a matching criterion to a previously randomized subject and administered the alternate treatment. We develop estimators for the average treatment effect that combine information from both the matched pairs and unmatched subjects as well as an exact test. Simulations illustrate the method's higher efficiency and power over competing allocation procedures in both controlled scenarios and historical clinical trial data.

### 13.1 Introduction

Randomization in experimentation is likely to work well with large sample size. With small sample size, the empirical distributions of relevant covariates can be different across treatments possibly masking an effect by creating bias and inflating variance. Some improvements over completely randomized design are rerandomization (Mor-

gan and Rubin, 2012), *a priori* matching (Raudenbush et al., 2007), and *adaptive design* which involves any change to the experiment or statistical procedures while the experiment is underway (Chow and Chang, 2008).

We limit our focus to *sequential experiments*, where subjects enter iteratively over time and the experimental condition is administered upon entrance, but the outcome does not necessarily have to be assessed before the next subject is enrolled. We develop a new adaptive design for sequential experiments whose goal is to elucidate an average treatment effect (ATE) between two arms, which we call treatment (T) and control (C). Sequential experiments are very popular in both clinical trials and recently, crowdsourced-Internet experimentation (Horton et al., 2011; Chandler and Kapelner, 2013).

Our design is a new type of *dynamic allocation*, a means of assigning T/C to newly-arrived subjects based on decisions about previous assignments, covariates, and/or responses (Hu and Rosenberger, 2006). Proposals of dynamic allocation began with Efron (1971)’s biased coin design. Here, a coin is biased in favor of the treatment with the fewest subjects, hence leading to better balance in treatment allocation. However, this procedure ignores subjects’ covariates.

The first line of defense to balance covariates is stratification (or “blocking”), a classic strategy dating back to Fisher’s agricultural experiments. Stratification becomes quickly impractical when the number of total covariate levels is large relative to sample size. Taves (1974) and others tackle these shortcomings by “minimizing” the imbalance between the treatments among all levels of covariates present. The most popular and widely implemented among these methods is found in Pocock and Simon (1975) whose procedure involves picking a few arbitrary functions to tailor the imbalances. The selection of these functions is still an ongoing area of research (for instance, see Han et al., 2009). Concerned by this arbitrariness, Atkinson (1982) posits a method solidly rooted in linear model theory using  $D_A$  optimality.

If the end goal of experimentation is to find an effect when one is present, then the primary concern is estimator efficiency and test power. Stratification and minimization methods rely on the logic that greater balance among the covariates implies greater efficiency which is mathematically true *only* in homoskedastic linear models (Rosenberger and Sverdlov, 2008).  $D_A$  optimality iteratively maximizes efficiency assuming the linear model without explicitly focusing on balance. Thus, we see one of the fundamental problems in previous allocation procedures is the reliance on the homoskedastic linear model, an assumption that is rarely true in practice. We wish to develop a dynamic allocation which is robust when the covariates combine non-linearly and with interactions to produce the response and no worse than current methods when the linear model holds.

The seminal guidebook, Cook and Campbell (1979), states “whenever possible, it is recommended to minimize error terms.” They then recommend *matching* subjects before randomization on covariates to create better stratification. It was not a novel idea; Student (1931) commented on the  $n = 20,000$  children Lanarkshire Milk Experiment proposing the experiment should be performed exclusively on 50 identical twin pairs which would be randomly assigned T/C.

We propose matching iteratively, *on-the-fly*. As subjects walk in the door or engage a survey online they should be matched with people “similar” to them who came in previously, a procedure which Whitehead (1997) believes to be “especially difficult.” Imagine the following scenario of a trial testing whether a pill decreases blood pressure. The investigators determine that age, height, weight, and race should be collected as covariates as they are known to be related to blood pressure. Bob, 28, 5’10”, 180lb and White enters and is determined to fit the requirements of the study. By the flip of a coin, he receives the pill. The next day, Grace, 45, 5’2”, 105lb and Asian enters. Based on inspection of this demographic data, she is clearly different from Bob; thus she is also randomized. Soon after, Joe, 29, 5’11”, 185lb and White

enters. We determine that he is similar to Bob, pair them, and deterministically administer to him the placebo. The trial continues and Grace would then await someone to be matched with, which may or may not occur.

The algorithm is simple: incoming subjects are either randomized and placed in a holding pool, called the “reservoir,” or if they’re found to match a subject already in the reservoir, they’re matched and given their match’s alternate treatment. The matches and the reservoir form two independent samples yielding two different estimators which are combined to estimate the ATE.

The closest idea we find in the literature is in Raghavarao (1980) who computes the Mahalanobis distances between a new subject’s covariates and the average covariates in the different treatment groups. The allocation to the treatment is then made via a biased coin with probabilities proportional to these distances. We use the idea of Mahalanobis distance which creates robustness to collinearity in the covariates, but we use it to match individual subjects together in pairs.

We layout our scenario assumptions, explain our algorithm and develop testing procedures in Section 13.2. We demonstrate our procedure’s improvements over previous procedures via simulations in Section 13.3. Our method performs particularly well in the case where the model is non-linear, performs respectably with linear models, and also performs respectively when the covariates do not inform the response. We then demonstrate higher efficiency using historical data from randomized controlled trials (RCT’s) in Section 13.4, where the covariate-response model was unknown but most likely non-linear. We discuss and describe future directions in Section 13.5.

## 13.2 The Algorithm, Estimation, and Testing

### 13.2.1 Problem Formulation

Subjects arrive sequentially and their covariates, denoted by  $\mathbf{x}_i := [x_{i1}, \dots, x_{ip}]$  which are either continuous or binary, are immediately recorded. The subjects must then be assigned to a treatment on-the-spot. We develop our method for allocating one of two treatments, T or C, to subject  $i$  denoted by the treatment indicator  $\mathbf{1}_{T,i}$ . The response  $y_i$  can be collected at any time after allocation. We assume the following model with independent observations, an additive treatment effect, a possibly non-linear covariate effect, normal and homoskedastic errors, fixed covariate design, and sample size  $n$  fixed in advance:

$$\begin{aligned} Y_i &= \beta_T \mathbf{1}_{T,i} + z_i + \mathcal{E}_i, & z_i &:= f(\mathbf{x}_i), \\ \mathcal{E}_i &\stackrel{iid}{\sim} \mathcal{N}(0, \sigma_e^2), & i &\in \{1, \dots, n\}. \end{aligned} \tag{13.1}$$

We wish to develop a dynamic allocation method followed by an unbiased estimator for  $\beta_T$  with higher efficiency and thereby more powerful when testing classic null hypotheses compared to popular approaches.

### 13.2.2 The Algorithm

The first  $n_0$  subjects enter the experiment and are randomized to T or C with the flip of a coin. These subjects comprise the initial “reservoir.” After a certain point, we would like to potentially match an incoming subject with subjects in the reservoir. We would like to match them on  $f(\mathbf{x})$ , which is latent, so we match on what we consider is the next best thing, the  $\mathbf{x}$ ’s themselves. We hope that  $\mathbf{x}_1 \approx \mathbf{x}_2$  implies  $f(\mathbf{x}_1) \approx f(\mathbf{x}_2)$  which is true if the function is sufficiently smooth.

We match using squared Mahalanobis distance which gives a convenient scalar distance between points in  $\mathbb{R}^p$  adjusting for collinearities. This metric has a long implementation history in matching applications dating back to Rubin (1979). Matching using Mahalanobis distance and then randomizing the pairs to T/C has been demonstrated to result in better balance and higher power (Greevy et al., 2004). Further, the assumption of normal covariates seems to work well with real data even when the covariates are non-normal (see Section 13.4). After matches are produced, we do not make use of the normality assumption of the  $\mathbf{x}$ 's further in the development of the estimator. Improvements to the matching machinery that may be more robust to real-world covariate distributions are also discussed in Section 13.5.

Thus, the new subject enters and the squared Mahalanobis distance between its covariate vector,  $\mathbf{x}_{\text{new}}$ , and each of the previous subject covariate vectors, the  $\mathbf{x}_{\text{old}}$ 's, are calculated. Denote  $\mathbf{S}$  as the covariates' sample variance-covariance matrix calculated with all subjects including the new subject. Assuming normal covariates, the squared Mahalanobis distance then has a scaled  $F$  distribution given below:

$$D^2 := (\mathbf{X}_{\text{new}} - \mathbf{X}_{\text{old}})^\top \mathbf{S}^{-1} (\mathbf{X}_{\text{new}} - \mathbf{X}_{\text{old}}), \quad (13.2)$$

$$\frac{n-p}{2p(n-1)} D^2 \stackrel{\text{approx}}{\sim} F_{p, n-p}.$$

We then take the minimum of the squared Mahalanobis distances between the new observation and each observation in the reservoir and calculate its probability. Let the minimum distance squared come from the previous subject,  $\mathbf{x}_{\text{old}}^*$ . If the probability is less than  $\lambda$ , a pre-specified hyperparameter, then  $\mathbf{x}_{\text{new}}$  and  $\mathbf{x}_{\text{old}}^*$  are matched together; if it's not,  $\mathbf{x}_{\text{new}}$  is randomized and added to the reservoir. If  $\mathbf{x}_{\text{new}}$  is matched, it is not added to the reservoir and  $\mathbf{x}_{\text{old}}^*$  is removed from the reservoir.  $\mathbb{1}_{T, \text{new}}$  is then assigned to be  $1 - \mathbb{1}_{T, \text{old}^*}$ , i.e.  $\mathbf{x}_{\text{old}}^*$ 's opposite treatment. The process is repeated until the  $n^{\text{th}}$  entrant. We left out other implementation details in this discussion but

make them explicit in algorithm 3. Note that our proposed procedure is considered a form of *covariate-adaptive randomization* (Rosenberger and Sverdlov, 2008, Section 2) because we are using the covariates to determine the dynamic allocation.

Note that upon matching, the treatment indicator is assigned *deterministically* to be the opposite of its match. This can cause selection bias if the investigator is not properly blinded. In defense of our decision to make allocation deterministic for a little less than half the subjects, note that our algorithm is sufficiently complicated that a duplicitous investigator would not be able to guess whether the entering subject will be assigned T or C based on previous information during a clinical trial. McEntegart (2003) discusses how this type of machination is unrealistic even in multi-center block permuted designs, which is much simpler than the allocation strategy proposed here. Further, if the procedure is implemented in an Internet-based experiment, the algorithm would be hard-coded and would not be subject to human tampering.

### 13.2.3 Estimation and Hypothesis Testing

We assume the covariate design matrix is fixed, the classic regression assumption. If so, the subjects ultimately matched and the subjects ultimately found in the reservoir are fixed as well. Thus conditioning on the design is equivalent to conditioning on the sigma field given below:

$$\mathcal{F} = \sigma\left(\underbrace{\langle \mathbf{x}_{T,1}, \mathbf{x}_{C,1} \rangle, \langle \mathbf{x}_{T,2}, \mathbf{x}_{C,2} \rangle, \dots, \langle \mathbf{x}_{T,m}, \mathbf{x}_{C,m} \rangle}_{\text{matched pairs}}, \underbrace{\mathbf{x}_{R,1}, \mathbf{x}_{R,2}, \dots, \mathbf{x}_{R,n_R}}_{\text{reservoir}}\right).$$

Upon completion of the experiment, there are  $m$  matched pairs and  $n_R$  subjects in the reservoir, both quantities fixed since the sample size and the design are fixed ( $n = 2m + n_R$ ). In our development of estimators and testing procedures, we always

---

**Algorithm 3** The sequential matching algorithm for subjects entering the experiment. The algorithm requires  $\lambda$  and  $n_0$  to be prespecified, which controls the ease of creating matches.

---

```

1: for  $t \leftarrow \{1, \dots, n\}$  do                                 $\triangleright n$  is the total sample size, fixed a priori
2:   if  $t \leq n_0$  or reservoir empty then
3:      $\mathbb{1}_{T,t} \leftarrow \text{Bern}\left(\frac{1}{2}\right)$  and subject  $t$  is added to the reservoir       $\triangleright$  randomize
4:   else
5:      $\mathbf{S}_t^{-1}$  is calculated using  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_t$   $\triangleright$  Estimate the true var-cov matrix
6:     Lookup  $F_{\lambda,p,t-p}^*$   $\triangleright F^*$  is the critical cutoff from the  $F$  distribution quantile
7:     for all  $\mathbf{x}_r$  in the reservoir do
8:        $F_r \leftarrow \frac{t-p}{2p(n-1)}(\mathbf{x}_t - \mathbf{x}_r)^\top \mathbf{S}_t^{-1}(\mathbf{x}_t - \mathbf{x}_r)$ 
9:     end for
10:     $F_{r^*} \leftarrow \min_r \{F_r\}$ ,  $r^* \leftarrow \arg \min_r \{F_r\}$   $\triangleright$  among ties, randomly select
11:    if  $F_{r^*} \leq F_{\lambda,p,t-p}^*$  then                                 $\triangleright$  a match is found
12:       $\mathbb{1}_{T,t} \leftarrow 1 - \mathbb{1}_{T,r^*}$        $\triangleright$  assign subject  $t$  the opposite of  $r^*$ 's assignment
13:       $[\mathbf{x}_{r^*}, \mathbb{1}_{T,r^*}]$  is removed from the reservoir
14:      record  $\langle \mathbf{x}_t, \mathbf{x}_{r^*} \rangle$  as a new match
15:    else                                                         $\triangleright$  a match is not found
16:       $\mathbb{1}_{T,t} \leftarrow \text{Bern}\left(\frac{1}{2}\right)$  and subject  $t$  is added to the reservoir       $\triangleright$  randomize
17:    end if
18:  end if
19: end for

```

---



assume conditioning on  $\mathcal{F}$ , thus this notation is withheld going forward.

We focus on testing the classic hypotheses  $H_0 : \beta_T = \beta_0$  versus  $H_a : \beta_T \neq \beta_0$ . We consider testing under three model assumptions (a) the response has normal noise and may possibly depend on covariates but we do not wish to model their effect (b) the response has normal noise and depends linearly on covariates (c) the response has mean-centered noise and depends on covariates through an unknown model. We develop testing procedures for each of these models: (a) a modification to the classic  $\bar{Y}_T - \bar{Y}_C$  in Section 13.2.3.1, (b) a modification to ordinary least squares regression in Section 13.2.3.2 and (c) an exact permutation test in Section 13.2.3.3.

### 13.2.3.1 The Classic Test

We define  $\bar{D}$  as the estimator for the average of the differences of the  $m$  matched pairs (treatment response minus control response) and  $\bar{R} := \bar{Y}_{R,T} - \bar{Y}_{R,C}$  as the difference in averages of the treatments and controls in the reservoir. We combine these two estimators of the treatment effect using a weight parameter,  $w\bar{D} + (1 - w)\bar{R}$ . We can find a  $w^*$  to minimize variance thereby obtaining the estimator

$$B_T = \frac{\sigma_R^2 \bar{D} + \sigma_D^2 \bar{R}}{\sigma_R^2 + \sigma_D^2} \quad \text{with} \quad \text{Var}[B_T] = \frac{\sigma_R^2 \sigma_D^2}{\sigma_R^2 + \sigma_D^2}. \quad (13.3)$$

$B_T$  is unbiased because under the assumption of fixed covariates, taking the expectation over treatment allocation and noise yields  $\mathbb{E}[\mathbb{E}[\bar{D}]] = \mathbb{E}[\mathbb{E}[\bar{R}]] = \beta_T$ . Standardizing  $B_T$  gives a standard normal due to the assumption of normal noise. To create a usable test statistic, note that the true variances are unknown, so we plugin  $S_D^2$ , the matched pairs sample variance estimator, and  $S_R^2$ , the pooled two-sample reservoir difference variance estimator. Note that the standard calculation of  $S_R^2$  assumes the number of treatments and controls in the reservoir is fixed in advance but in our algorithm, these quantities are random. A more careful calculation could

include this randomness.

Equation 13.4 shows the resulting statistic which has an asymptotically standard normal distribution.

$$\frac{B_T - \beta_0}{\text{SE}[B_T]} \approx \frac{\frac{S_R^2 \bar{D} + S_D^2 \bar{R}}{S_R^2 + S_D^2} - \beta_0}{\sqrt{\frac{S_R^2 S_D^2}{S_R^2 + S_D^2}}} \xrightarrow{\mathcal{D}} \mathcal{N}(0, 1) \quad (13.4)$$

This asymptotic result holds if both  $m \rightarrow \infty$  and  $n_R \rightarrow \infty$  since  $S_D^2 \xrightarrow{p} \sigma_D^2$  and  $S_R^2 \xrightarrow{p} \sigma_R^2$ . The results also holds if  $m \rightarrow \infty$  and  $n_R$  is bounded which would occur in practice as the covariates are typically categorical or have compact support. In this case,  $S_R^2/(S_R^2 + S_D^2) \xrightarrow{a.s.} 1$  which reduces Equation 13.4 above to  $(\bar{D} - \beta_0)/S_D \xrightarrow{\mathcal{D}} \mathcal{N}(0, 1)$ . Also, by the assumption of additive and normal noise, the estimator is unbiased for finite  $n$ .

Note that in the case where there are no matched pairs, we default to the classic estimator and in the case where there are fewer than two treatments or controls in the reservoir, we default to the matched pairs estimator.

When is this estimator more efficient than the standard classic estimator,  $\Delta \bar{Y} := \bar{Y}_T - \bar{Y}_C$ ? In other words, when is  $\sigma_{\Delta \bar{Y}}^2/\sigma_D^2 > 1$ ? Assuming perfect balance in its treatment allocation ( $n_T = n_C = \frac{n}{2}$ ) for the classic estimator and taking the expectation over both noise and treatment allocation, it can be shown that the variances are:

$$\begin{aligned} \sigma_D^2 &= \frac{1}{m^2} \sum_{k=1}^m (z_{T,k} - z_{C,k})^2 + \frac{2}{m} \sigma_e^2, \\ \sigma_{\Delta \bar{Y}}^2 &\approx \frac{4}{n^2} \sum_{i=1}^n z_i^2 + \frac{4}{n} \sigma_e^2. \end{aligned} \quad (13.5)$$

This means that the better the matching, the smaller  $\sum_{k=1}^m (z_{T,k} - z_{C,k})^2$  will be, the smaller the variance becomes, and the higher the power. If we further allow  $n_R = 0$  (*all* the subjects matched), then it's clear that  $\sigma_{\Delta_Y}^2 / \sigma_{\bar{D}}^2 > 1$  if and only if  $\sum_{k=1}^m z_{T,k} z_{C,k} > 0$ . Note that the approximation in the last expression is due to ignoring covariance terms which do not exist when conditioning on  $n_T$  and  $n_C$ .

### 13.2.3.2 The Least Squares Test

To construct a test when the response is linear in the covariates or when we wish to make linear adjustments, we extend the idea in the previous section where we combined an effect estimate from the matched pairs data and an effect estimate from the reservoir data to regression models. Consider the model for the response differences among the matched pairs:  $D_k = \beta_{0,D} + \beta_{1,D}\Delta x_{1,k} + \dots + \beta_{p,D}\Delta x_{p,k} + \mathcal{E}_{k,D}$  where  $\mathcal{E}_{k,D}$  is  $\overset{iid}{\sim}$  normal noise and  $k$  ranges from 1 to  $m$ . The parameter of interest is the intercept,  $\beta_{0,D}$ , with OLS estimator  $B_{0,D}$ , the analogue of  $\bar{D}$  in the previous section.  $\Delta x_{1,k}, \dots, \Delta x_{p,k}$  are the differences between treatment and control within match  $k$  for each of the  $p$  covariates respectively.  $\beta_{1,D}, \dots, \beta_{p,D}$  are nuisance parameters that adjust for linear imbalances in the covariate differences not accounted for in the matching procedure.

For the responses in the reservoir, consider the classic model:  $Y_i = \beta_{T,R}\mathbb{1}_{T,R,i} + \beta_{0,R} + \beta_{1,R}x_{1,i} + \dots + \beta_{p,R}x_{p,i} + \mathcal{E}_{i,R}$  where  $\mathcal{E}_{i,R}$  is  $\overset{iid}{\sim}$  normal noise and  $i$  ranges from 1 to  $n_R$ . The parameter of interest is the additive effect of the treatment,  $\beta_{T,R}$ , with OLS estimator  $B_{T,R}$ , the analogue of  $\bar{R}$  in the previous section.  $\beta_{0,R}, \beta_{1,R}, \dots, \beta_{p,R}$  are nuisance parameters that adjust for a mean offset and linear imbalances in the covariates.

Using the parallel construction in Equations 13.3 and 13.4, our modified OLS estimator has the form

$$\frac{\frac{S_{B_{T,R}}^2 B_{0,D} + S_{B_{0,D}}^2 B_{T,R}}{S_{B_{T,R}}^2 + S_{B_{0,D}}^2} - \beta_0}{\sqrt{\frac{S_{B_{T,R}}^2 S_{B_{0,D}}^2}{S_{B_{T,R}}^2 + S_{B_{0,D}}^2}}} \xrightarrow{\mathcal{D}} \mathcal{N}(0, 1) \quad (13.6)$$

with the same asymptotics described in the last section below Equation 13.4.  $S_{B_{T,R}}^2$  is the sample variance of  $B_{T,R}$  and  $S_{B_{0,D}}^2$  is the sample variance of  $B_{0,D}$ .

### 13.2.3.3 The Permutation Test

An application of Fisher's exact test is straightforward. For the matched pairs component of the data, we examine the  $2^m$  configurations (each match can have T-C or C-T) to compute all  $\bar{d}$ 's. For the reservoir portion of the estimator, we condition on  $n_{R,T}$  and examine every possible arrangement of the treatment vector to compute every  $\bar{y}_{R,T} - \bar{y}_{R,C}$ . For each arrangement, we also compute  $s_D^2$  and  $s_R^2$  to create values of the test statistic in Equation 13.3. In practice, the 2-sided  $p$ -value is approximated by comparing the observed  $|b_T|$  from the true sample data to absolute values of Monte-Carlo samples from the space of all possible test statistics. A similar exact test is available using the modified OLS estimates but we do not explore it in this study.

### 13.2.4 Properties of the Reservoir Size

We wish to gain insight about how  $\lambda$ ,  $n_0$  and  $n$  affect  $n_R$ . Assume for now that we only have one covariate,  $x$  (which may also be the largest principal component of a collection of covariates) and we allow immediate matching ( $n_0 = 1$ ). Mahalanobis distance matches on standardized distance. For this illustration, assume we match if the two  $x$ 's sample quantiles are within  $\lambda$  of each other. For example, the latest

subject in the experiment had a sample quantile of 0.96, they would be matched to the closest subject in the reservoir with quantile between 0.91 and 1 at  $\lambda = 0.10$ .

Consider dividing the unit interval into  $K := 1/\lambda$  intervals of equal length. Two items in one interval qualify to be matched. Assume that  $K$  is even (similar results follow for  $K$  odd). Consider the Markov process that transitions after each pair of subjects. Let  $s$  be the state that  $2m$  of the  $K$  cells are occupied for  $s \in \{0, 1, \dots, K/2\}$ . It is straightforward that  $P_{i,j}$ , the transition probability of pairs from state  $i$  to  $j$ , satisfies:

$$P_{s,j} = \begin{cases} \frac{2s(2s-1)}{K^2}, & j = s - 1, \quad s \neq 0 \\ \frac{K(4s+1)-8s^2}{K^2}, & j = s \\ \frac{K^2-K(4s+1)+2s(2s+1)}{K^2}, & j = s + 1, \quad s \neq \frac{K}{2}. \end{cases}$$

Note the inherent symmetry:  $P_{s,j} = P_{K/2-s, K/2-j}$ . Hence, the steady-state probabilities are symmetric about  $s = K/4$ . Therefore, the mean number of items in the reservoir goes to  $K/2 = (2\lambda)^{-1}$  as  $n$  grows arbitrarily large. For example,  $\lim_{n \rightarrow \infty} \mathbb{E}[N_R \mid \lambda = 0.10] = 5$ .

### 13.3 Simulation Studies

We demonstrate our method's performance by simulating in three scenarios: covariates affect the response non-linearly (the “NL” scenario), covariates affect the response linearly (the “LI” scenario) and covariates do not affect the response (the “ZE” scenario). These scenarios were simulated via the settings found in Table 13.1. In practice, we simulated many settings for the NL and LI scenarios with similar results.

We set the treatment effect to be  $\beta_T = 1$ .  $n$  was varied over  $\{50, 100, 200\}$ ,  $n_0$  was

Scenario	$Y_i$
NL	$\beta_T \mathbf{1}_{T,i} + x_{1,i} + x_{2,i} + x_{1,i}^2 + x_{2,i}^2 + x_{1,i}x_{2,i} + \mathcal{E}_i$
LI	$\beta_T \mathbf{1}_{T,i} + 2x_{1,i} + 2x_{2,i} + \mathcal{E}_i$
ZE	$\beta_T \mathbf{1}_{T,i} + \mathcal{E}_i$

Table 13.1: The response models for the three scenarios proposed. The covariates were  $X_{1,i} \stackrel{iid}{\sim} \mathcal{N}(0, 1)$  and  $X_{2,i} \stackrel{iid}{\sim} \mathcal{N}(0, 1)$  and the errors were  $\mathcal{E}_i \stackrel{iid}{\sim} \mathcal{N}(0, \sigma_e^2)$ .

set to the minimum to admit calculation of  $\mathbf{S}^{-1}$  (at  $n_0 = p = 2$ ) and  $\lambda$  was varied over nine settings between 0.01 and 0.75. We then used  $\sigma_e^2$  to modulate the resolution in our comparisons. We chose  $\sigma_e^2 = 3$  to be a good balance because even at  $n = 200$  comparisons were clear.

In choosing which competitor dynamic allocation methods to simulate against, we wanted to pick methods that are in use in sequential trials. According to Scott et al. (2002), stratification is very popular and Efron’s biased coin has been used in a few studies. Most popular is minimization which has been used in over 1,000 trials (McEntegart, 2003).

Thus, we choose to compare our method against (1) complete randomization (CR). (2) stratification: both  $x_1$  and  $x_2$  were stratified into three levels based on the tertiles of the standard normal distributions, creating  $3 \times 3 = 9$  blocks. Within blocks we alternate T / C in order to coerce  $n_T \approx n_C$  so no power is lost on allocation imbalance. (3) Efron’s biased coin design (BCD): we use the bias parameter of  $\alpha = 2/3$  which is Efron’s “personal favorite.” (4) Minimization: we used the same blocks as stratification, the “ $D$ ” function (Begg and Iglewicz, 1980 found the variance method performed slightly better than alternatives), the “ $G$ ” function, and we set  $p = 1$  for deterministic assignments in order to force  $n_T \approx n_C$  so no power is lost on allocation imbalance. (5) stratification as above followed by post-matching which we detail in Web Appendix A in the Supplementary Materials.

Note that power for our stratification and minimization simulations may be higher

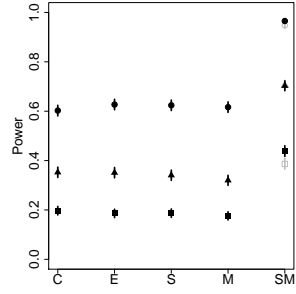
than what is expected in practice due to two reasons. First, we use deterministic assignments. Second and more importantly, real world covariates are collinear which reduces the effectiveness of stratification and minimization. Here, we generate covariates independently. Note that collinearity is not an issue for our sequential matching proposal due to the use of Mahalanobis distance.

There are three scenarios (NL, LI, and ZE) and four competitors. Naturally, we want to gauge performance if we assumed the correct underlying model, but we also want to ensure we are robust if the model is misspecified. Therefore, we simulate each of these under the three model assumptions discussed in Section 13.2.3. For the classic estimator, all competitors employed  $\bar{Y}_T - \bar{Y}_C$ ; for the linear estimator, all competitors employed OLS; and for the exact test, all competitors employed the standard conditional permutation test.

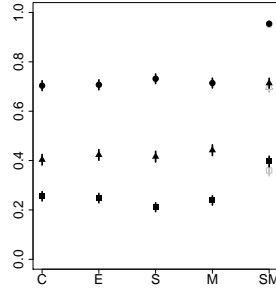
We hypothesize that in the case of no effects (the ZE scenario), we will slightly underperform against competitors under all three testing procedures because of the loss of power due to the lower effective sample size when analyzing paired differences. If the effects are linear, we hypothesize to do slightly worse against the OLS procedures due to lower effective sample size. Under all other scenarios and models, we expect to do better.

We simulated each scenario 2,000 times and for exact testing, we Monte-Carlo sampled 1,000 times within each simulation.

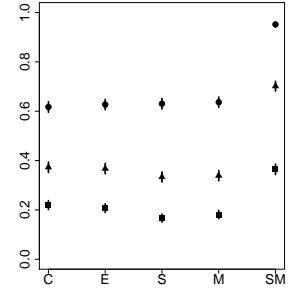
Our main metric for comparison is power, the proportion of the times the null was rejected under the Type I error rate of  $\alpha = 5\%$ . We also record standard error of the estimate (when the estimator was parametric) as well as balance (the maximum standardized difference in the averages of covariates between treatment and control samples:  $\max_{j \in \{1,2\}} \{(\bar{x}_{j,T} - \bar{x}_{j,C})/s_j\}$ ). Results for power under  $\lambda = 10\%$  against the null of no treatment effect are illustrated in Figure A.17c and results for balance and relative efficiency vis-a-vis other methods are found in Table 13.2. Power results for



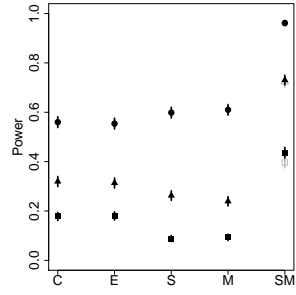
(a) Scen NL, Classic Test



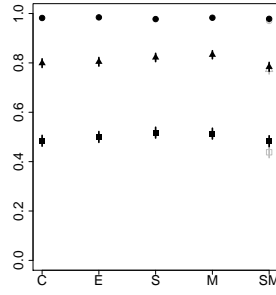
(b) Scen NL, Linear Test



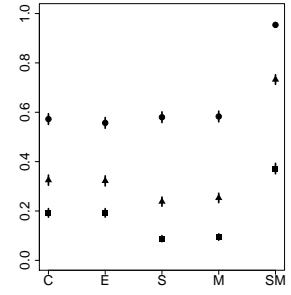
(c) Scen NL, Exact Test



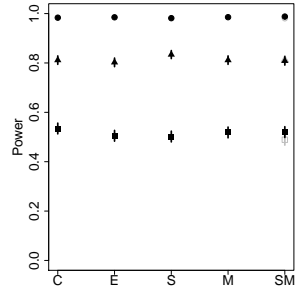
(d) Scen LI, Classic Test



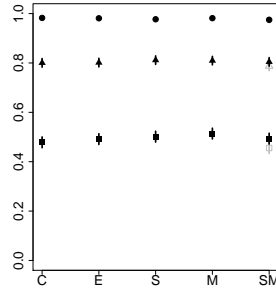
(e) Scen LI, Linear Test



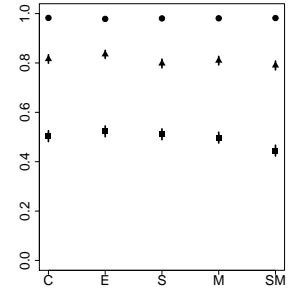
(f) Scen LI, Exact Test



(g) Scen ZE, Classic Test



(h) Scen ZE, Linear Test



(i) Scen ZE, Exact Test

Figure 13.1: Power at  $\alpha = 0.05$  illustrated for matching parameter  $\lambda = 10\%$  for the three scenarios by the three testing procedures, all five allocation methods (C: Complete Randomization, E: Efron's BCD, S: Stratification, M: Minimization and SM: our sequential matching algorithm) and sample sizes (squares illustrates results for  $n = 50$ , triangles for  $n = 100$  and circles for  $n = 200$ ). Plotted points represent the sample proportion of null hypothesis rejections in 2,000 simulations and segments represent 95% confidence intervals. The grey points and segments plot the  $T$  approximation to the finite distribution of Equations 13.4 and 13.6.



$n$	Allocation Method	Balance	Sample Relative Efficiency Over Competitors					
			Scenario NL		Scenario LI		Scenario ZE	
			Classic	Linear	Classic	Linear	Classic	Linear
50	CR	0.812	<b>2.233</b>	<b>1.645</b>	<b>2.543</b>	<b>0.814</b>	<b>0.896</b>	<b>0.779</b>
	Efron's BCD	0.810	<b>2.134</b>	<b>1.744</b>	<b>2.358</b>	<b>0.805</b>	<b>0.854</b>	<b>0.825</b>
	Stratification	0.422	<b>1.664</b>	<b>1.282</b>	<b>1.125</b>	<b>0.780</b>	<b>0.798</b>	<b>0.717</b>
	Minimization	0.397	<b>1.750</b>	<b>1.478</b>	<b>1.163</b>	<b>0.778</b>	<b>0.842</b>	<b>0.718</b>
	Seq. Matching	0.506	—	—	—	—	—	—
100	CR	0.804	<b>2.610</b>	<b>1.762</b>	<b>2.637</b>	<b>0.898</b>	0.944	<b>0.885</b>
	Efron's BCD	0.802	<b>2.500</b>	<b>1.688</b>	<b>2.709</b>	<b>0.855</b>	0.967	<b>0.885</b>
	Stratification	0.392	<b>1.785</b>	<b>1.401</b>	<b>1.224</b>	<b>0.820</b>	0.928	<b>0.879</b>
	Minimization	0.376	<b>2.975</b>	<b>1.532</b>	<b>1.132</b>	<b>0.842</b>	0.932	<b>0.879</b>
	Seq. Matching	0.430	—	—	—	—	—	—
200	CR	0.801	<b>3.139</b>	<b>2.011</b>	<b>3.183</b>	0.969	0.991	<b>0.851</b>
	Efron's BCD	0.796	<b>2.695</b>	<b>2.034</b>	<b>3.187</b>	0.979	1.005	<b>0.866</b>
	Stratification	0.378	<b>2.084</b>	<b>1.586</b>	<b>1.400</b>	0.977	1.002	0.917
	Minimization	0.370	<b>2.241</b>	<b>1.757</b>	<b>1.289</b>	0.938	0.964	<b>0.883</b>
	Seq. Matching	0.361	—	—	—	—	—	—

Table 13.2: Balance results and relative sample efficiency results (the ratio of the two unbiased sample variances) of sequential matching ( $\lambda = 0.10$ ,  $Z$  approximation) versus competitors by scenario and testing procedure. Balance results are averages over all scenarios and model assumptions (60,000 simulations for SM and 36,000 for other allocations). Efficiencies in **gray** indicate our algorithm performed worse than a competitor and efficiencies in **bold** indicate our algorithm performed better (as measured by an F-test with 1% significance level unadjusted for multiple comparisons). Exact tests are not shown because they do not admit a standard error calculation.

other values of  $\lambda$  appear as Web Figures 1 to 8 in the Supplementary Materials. The levels of  $\lambda$  have little effect the comparisons against the four competitors. A table of empirical bias results appears as Web Table 1 in the Supplementary Materials.

In the NL scenario, our sequential matching procedure dominates competitors in power and efficiency, sometimes doubling power and nearly tripling efficiency. Even at  $n = 200$ , the gains are large. Regression adjustment helps the competitors, but it cannot adjust for the non-linear portion of the quadratic terms and interaction term; they will appear as higher noise.

In the LI scenario, sequential matching dominates competitors in the classic and the exact test because the competitors do not use the covariate information. In the linear assumption, sequential matching performs similarly in power but has a lower efficiency across all  $n$ 's. This loss is due to a lower effective sample size when using matched pairs. The gap decreases as  $n$  increases because there are benefits to matching even when employing regression adjustment. As Greevy et al. (2004) explain, better balance reduces collinearity resulting in a smaller standard error for the estimate. Balance is improved over competitors that do not allocate based on the covariates and this better balance implies higher power and efficiency. Parenthetically, we note that as  $n$  increases, it appears as if balance is approaching levels observed in both stratification and minimization. This is expected and is an added bonus of our procedure.

In the ZE scenario, our approach is most severely impacted by lower effective sample size. However, power is not as low as expected. Efficiency seems to be lost for all simulated  $n$ 's but most significantly when  $n = 50$  as expected.

All in all, sequential matching shines in the case of non-linear covariate models which is the most realistic case in practice. If the covariate model is truly linear, sequential matching does worse when OLS is employed but our relative inefficiency is only observed for small sample sizes. In the case when covariates do not matter at

all, we begin to perform about equally with competitors when  $n \geq 100$ . This is an important result in practice because investigators sometimes choose useless covariates which do not affect the outcome measure.

A possible criticism of the high power achieved is we assume our  $n$  was large enough for the estimators in Equations 13.4 and 13.6 to converge. Via unshown simulations, we have reason to believe the true density to be similar to a  $T$  with degrees of freedom  $\max(m - 1, \min(n_{R,T}, n_{R,C}))$  where  $n_{R,T}$  and  $n_{R,C}$  are the number of treatments and controls left in the reservoir respectively. We illustrate power results for this putative  $T$  distribution in gray in Figure A.17c. To further investigate the convergence, we also simulated the size of the tests in Table 13.3. For the classic estimator, at  $n = 50$ , the size of the  $Z$  approximation test is generally larger than the Type I error rate of 5% even with the  $T$  procedure (in most cases). At  $n = 100$  and 200, the  $Z$  approximation is appropriately sized for most scenario-model situations, and the  $T$  procedure is appropriately sized for almost all cases. Other anomalies observed in this table are discussed in Section 13.5.

## 13.4 Demonstration Using Real Data

### 13.4.1 Behavioral Experiment

Kapelner and Chandler (2010) (Chapter 3 of this document) ran experiments using the Amazon Mechanical Turk platform, a global outsourcing website for small one-off tasks that can be completed anonymously on the Internet. They focused on measuring subjects' stated preference for a beer price when the beer came from different purchasing locations (an online replication of Thaler, 1985's demonstration of the "framing effect," a cognitive bias). The treatment involved subtle text manipulations: the same beer came from either a *fancy resort* or a *run-down grocery store*. In

$n$	Allocation Method	Scenario NL			Scenario LI			Scenario ZE		
		Classic	Linear	Exact	Classic	Linear	Exact	Classic	Linear	Exact
50	CR	0.047	0.047	0.052	0.046	0.046	0.056	0.053	0.050	0.048
	Efron's BCD	0.050	0.046	0.040*	0.054	0.055	0.046	0.053	0.051	0.046
	Stratification	0.016**	0.031**	0.018**	0.006**	0.051	0.006**	0.050	0.049	0.046
	Minimization	0.021**	0.041	0.024	0.004**	0.046	0.005**	0.046	0.055	0.050
	Seq. Match (Z)	0.078**	0.072**	0.052	0.077**	0.089**	0.046	0.087**	0.093**	0.044
	Seq. Match (T)	0.056	0.052		0.061*	0.077*		0.076**	0.073**	
100	CR	0.050	0.047	0.050	0.042	0.053	0.054	0.054	0.049	0.048
	Efron's BCD	0.053	0.050	0.048	0.054	0.046	0.046	0.052	0.054	0.050
	Stratification	0.034*	0.034*	0.023**	0.004**	0.046	0.004**	0.048	0.050	0.046
	Minimization	0.032**	0.040*	0.043	0.004**	0.048	0.061*	0.046	0.046	0.050
	Seq. Match (Z)	0.054	0.054	0.051	0.054	0.062*	0.050	0.063*	0.061**	0.048
	Seq. Match (T)	0.047	0.052		0.050	0.063*		0.060*	0.070**	
200	CR	0.052	0.043	0.051	0.051	0.053	0.050	0.043	0.053	0.060*
	Efron's BCD	0.052	0.050	0.050	0.044	0.052	0.048	0.052	0.051	0.050
	Stratification	0.018**	0.027**	0.014**	0.004**	0.054	0.003**	0.051	0.050	0.048
	Minimization	0.018**	0.032*	0.024**	0.004**	0.053	0.001**	0.054	0.046	0.052
	Seq. Match (Z)	0.053	0.056	0.049	0.052	0.049	0.048	0.054	0.062*	0.054
	Seq. Match (T)	0.049	0.060		0.054	0.058		0.050	0.058	

Table 13.3: Simulated size of tests for all scenarios, competitors, and all tests at  $\lambda = 10\%$ . Numbers followed by a \*\* indicate they are different from the purported 5% size at a Bonferroni-corrected significance level (162 comparisons). Numbers followed by a \* indicate they are different from the purported 5% size without Bonferroni correction.

their control wing ( $n = 168$ ), no tricks were employed to ensure the subjects were paying attention to the text. Thus, in this wing, the subtle text manipulation did not seem to affect the subjects' stated beer prices. The effect may have been real, but the data was either too noisy or there was insufficient sample size to find it. We demonstrate here that if our sequential matching procedure was employed, the effect estimator would have been more efficient.

For matching, we first used most of the covariates found in the original dataset: age, gender, level of earnings, number of weekly hours spent doing one-off tasks, level of multitasking when performing tasks, stated motivation level, passing the “instructional manipulation check” (Oppenheimer et al., 2009) and a survey gauging the subject's “need for cognition” (Cacioppo and Petty, 1982).

We note that  $R^2$  under OLS was about 18.7%. We then run two off-the-shelf machine learning algorithms that are designed to find interactions and non-linearities

in the response function. The in-sample pseudo- $R^2$  using Chipman et al. (2010)’s Bayesian Additive Regressive Trees (BART) was 42.4% and Breiman (2001b)’s Random Forests (RF) was 70.4%. Although this is not a formal test, it is pretty compelling evidence that the covariates do not combine strictly linearly to inform beer price. Thus, as demonstrated in figure A.17a and column 4 of table 13.2, our method should be more powerful and more efficient than using previous dynamic allocation strategies with a classic estimator. The results for 200 simulations at  $\lambda = 0.10$  are shown in table 13.4a. Many of the covariates are binary. Thus, the variance-covariance matrix was *not* invertible in line 5 of algorithm 3 for many of the early iterations, so we used the Moore-Penrose generalized inverse instead.

	purported $n$	average actual $n$	average efficiency	approx. sample size reduction
(a)	50	37.8	1.84	45.7%
	100	71.9	1.23	16.9%
	168 (all)	116.1	1.06	5.4%
(b)	50	34.9	2.01	50.1%
	100	67.8	1.60	37.3%
	168 (all)	112.1	1.57	36.3%

Table 13.4: Results for 200 simulations of the sequential matching procedure over many values of  $n$  and  $\lambda = 0.10$ . (a) all covariates matched on (b) four cherry-picked covariates are matched on (OLS has an  $R^2 = 20.8\%$ , BART, 32.1% and RF, 26.5%). Note that many of the covariates were binary, and thus the variance-covariance matrix was *not* invertible in line 5 of algorithm 3 for many of the early iterations of these simulations, so we used the Moore-Penrose generalized inverse.

We now match on four selected covariates that come out most significant in an OLS regression on the full dataset: age, level of earnings, level of multitasking when performing tasks and one question from the survey gauging the subject’s “need for cognition.” The results are found in table 13.4b. Note that the efficiencies are higher and do not drop off as quickly when  $n$  increases. Thus, matching on *relevant* covariates

yields a performance enhancement in our procedure.

### 13.4.2 Clinical Trial

We now examine sequential clinical trial data from Foster et al. (2010), a twelve-week, multicenter, double-blind, placebo-controlled clinical trial studying whether amitriptyline, an anti-depressant drug, can effectively treat painful bladder syndrome. The study measured many outcomes, including change in pain after 12 weeks (difference in Likert scale scores). The confidence interval for the ATE between pill and placebo for this outcome measure was  $[-1.00, 0.30]$  with a  $p$  value of 0.29 (Table 2, row 1, page 1856). Such results indicate the effect may have been real but there wasn't enough power to detect it due to low sample size or a high error variance.

For matching, we first use most of the covariates found in the original dataset: age, gender, race (White / Hispanic), level of education, level of employment, living with a partner, presence of sexually transmitted diseases and urinary tract infection, as well as baseline measures of pain, urination frequency and urgency, quality of life, anxiety and depression as well as syndrome symptom levels.

$R^2$  was about 25.2% in an OLS regression. We then run two off-the-shelf machine learning algorithms that are designed to find interactions and non-linearities in the response function. The in-sample pseudo- $R^2$  using Chipman et al. (2010)'s Bayesian Additive Regressive Trees (BART) was 42.6% and Breiman (2001b)'s Random Forests (RF) was 82.4%. Although this is not a formal test, it is pretty compelling evidence that the covariates do not combine strictly linearly to inform change in pain after 12 weeks. Thus, as demonstrated in Figure A.17a and column 4 of Table 13.2, our method should be more powerful and more efficient than using previous dynamic allocation strategies with a classic estimator.

We simulate the subjects being dynamically allocated using the sequential matching procedure by first assuming the entering subjects do not exhibit any time trend;

this will allow us to permute their order. During the iterative procedure, all subjects assigned to the reservoir keep whichever assignment they had during the experiment. During matching, if the subject happened to have been assigned the treatment which the sequential matching procedure allocated, they are kept in the subject pool; if not, they are discarded (this is illustrated in Figure 13.2). Thus, during our simulations, we result in a *subset* of the data we began with. Note that we only show results for the classic estimator versus the modified estimator in Equation 13.4, not the OLS modified estimator whose results we suspect to be similar.

```

nsim: 1 .....o.xx....o....xxx.oxxx.o.xxooxooxx.xo. (37)
nsim: 2 .....xxx..x.xxoo.xxx.o.o.oo..ox.xxxxx.o (35)
nsim: 3 .....x.x..oo.xxxo.x.xxxxxxo.x.ox.oxx.o (34)
nsim: 4 .....o..o..ox.o.o.xo.ox.o....xxooox.oxx.xo (42)
nsim: 5 .....x.xoo.o.o..xxoxx.o.x.x.oo.o.xoxx.. (39)
nsim: 6 .....x..xx.x..xx.oo.xoxoxxx.oxxo.o.xxxo (35)
nsim: 7 .....x.o..oo..oxx.x..o.oxo.xxox.o.xoxxxoxo (37)
nsim: 8 .....xx.oxxoxo..x.xxx.x..oxoo.o.xx.x..xox.o (34)
nsim: 9 .....o...xxoo.xo.ooo.....o.oxxox.oxx..o.x (42)

```

Figure 13.2: Running an  $n = 50$  subset of historical data through the sequential procedure. The dots represent a subject being placed into the reservoir. The “o” signifies that the subject was matched and that their treatment allocation was *opposite* of their matching partner. The “x” signifies that the subject was matched but their treatment allocation was the *same* as their matching partner, resulting in the subject being discarded. The number in parenthesis at the end of the line is the sample size retained of the purported 50.

The results for 200 simulations are shown in Table 13.5a. We also match on the top four covariates which are the most significant in an OLS of the full dataset: living with a partner, baseline pain, frequency of pain and syndrome symptom levels. The results are found in Table 13.5b. Note that these efficiencies are higher when compared to matching on all covariates and they do not suffer the steep drop off as  $n$  increases.

	purported $n$	average actual $n$	average efficiency	approx. sample size reduction
(a)	50	38.9	1.30	23.0%
	100	75.2	1.10	9.2%
	150	111.3	1.05	4.9%
	224 (all)	165.5	1.07	6.7%
(b)	50	38.0	1.27	21.2%
	100	72.9	1.23	18.8%
	150	108.6	1.15	13.2%
	224 (all)	160.3	1.13	11.3%

Table 13.5: Results for 200 simulations of the sequential matching procedure over many values of  $n$  and  $\lambda = 0.10$ . (a) all covariates matched on (b) four cherry-picked covariates are matched on (OLS has an  $R^2 = 18.9\%$ , BART, 35.1% and RF, 45.0%). Note that many of the covariates were binary, and thus the variance-covariance matrix was *not* invertible in line 5 of algorithm 3 for many of the early iterations of these simulations, so we used the Moore-Penrose generalized inverse.

## 13.5 Discussion

Estimation in sequential experiments, of which many are clinical trials, can have higher power and efficiency if the covariate information is leveraged. We present a dynamic allocation of treatment and control that matches subjects on-the-fly via a novel algorithm and present modified estimators of classic approaches: average difference, linear regression, and permutation testing. We simulate under different scenarios and illustrate higher power in scenarios where competing methods cannot make proper use of covariate information. We underperform only in the case of low sample size when the covariate model is linear or non-existent. In simulations with real clinical data, we find the efficiency of our method over complete randomization increases as the covariates become more important. This is most likely due to the fact that real-world response functions, such as this one in the clinical setting, combine covariates non-linearly, and this is when our procedure is most advantageous.



We note that “analysis assumptions may be compromised due to the ‘pseudo’-random allocation” (Scott et al., 2002) and would like to address this criticism which can be made about our procedure. Note that in Table 13.3, the size of the tests under stratification and minimization are less than 5%. One should not use the classic estimator in these cases because one implicitly tried to balance on the covariates, but then did not include the covariates in the model (as seen by the poor sizes using the classic model in scenario NL using the classic model in scenario LI in Table 13.3). Simon (1979) and Senn (2000) have very good discussions about this issue and recommend using regression adjustment (as seen in the competitors in scenario LI, linear response model in the Table 13.3 for  $n > 50$ ). As for exact testing under dynamic allocation, Kalish and Begg (1987) and many others warn that permutation distributions in stratification and minimization are incorrect unless the investigator permuted the treatment allocation according to how the allocation was determined by the covariates. This is not straightforward in practice for stratification and even less straightforward for minimization.

In contrast, the sizes of the test for our approach seem to be correct in Table 13.3, especially when making use of the low-sample  $T$  approximation. Usage of our classic estimator seems ill-advised for the same reasons that the classic estimator is not recommended under stratification and minimization. However, using linear regression on the covariates is also ill-advised when the model is non-linear or otherwise does not satisfy the OLS model assumptions (Freedman, 2008) although Rubin (1979) finds that covariance adjustment of matched pair differences is robust to model misspecification. Thus, since our permutation test performs well in the non-linear case, properly sized and can avoid many of the above issues, we recommend the permutation test in practice. We permuted according to the structure of our dynamic matching allocation, thus our permutation tests are valid.

### 13.5.1 Further Developments

We view this contribution as a step forward in covariate-adaptive randomization in sequential experiments but it is far from complete. We list extensions below which we believe are in difficulty order.

**Sequential Analysis** Although we assume fixed  $n$  in our construction, it is relatively straightforward to adapt to a fully or group sequential design whose methods can be found in Jennison and Turnbull (2000). It would be hard to tabulate values when our estimator has unknown convergence properties, thus it would probably have to be done by waiting until the estimator most likely converged, and then using standard sequential analysis software.

**Multi-armed designs** This paper is devoted to studies where there are only two arms. If the study is multi-armed, the approach can be modified. Upon matching, the newly matched subject could be randomly assigned to a random treatment (an arm other than the treatment of the reservoir subject) or perhaps, alternatively, assigned to one of the other arms to best achieve balance. Analogous estimators can then be created for contrasts of interest.

**Non-continuous outcomes** We believe with adjustment of the estimators, our method can apply beyond continuous responses to binary, ordinal, or count responses.

**Imbalance protection** Complete randomization when allocating subjects in the reservoir may result in treatment imbalances in the reservoir after the experiment is completed. In clinical trials, block permutations are popularly employed to avoid imbalance. In our case, blocks for reservoir allocation would not work since, upon matching, the block sequences would be broken. A solution would be to use a biased coin allocation scheme (Efron, 1971) or an urn model. This would involve modification

of the exact test of Section 13.2.3.3 to reflect the restricted randomization scheme.

**Multi-centered designs** Clinical trial recruitment is frequently spread across multiple centers and center-center variation is expected. Our procedure can be modified in two ways to be of use in multiple centers. First, if large enrollment is expected in each center, the procedure can run independently in each center and resulting estimates from each center can be combined upon study completion. Second, if small enrollment is expected in each center, the subjects can be matched across centers and the least squares estimator of section 13.2.3.2 can be used to adjust for center-center variation.

**Better matching distance** We feel that the most significant improvement would be better matching. Mahalanobis distance is logical, but prone to strange behavior with departures from the normality assumption. Another natural extension is to bootstrap the distribution of the nominal metric in Equation 13.2 as to not rely on probabilities from the scaled  $F$  distribution. Also, practitioners may want to weight the variables in the matching as well as force some variables to always match (see Rosenbaum, 2010, chapter 8).

**On-the-fly variable selection** Of course, our procedure suffers from the central issue of all matching: selecting the variables to match on. A poor choice makes a big difference as evidenced by the simulations on clinical trial data (Table 13.5a vs. 13.5b). There may be a way to *iteratively* match and differentially weight by covariates that are found to be important, so the set of perceived important covariates is updated during the sequential experiment.

**Reservoir optimization** We have begun to consider how large  $n_R$  grows asymptotically as a function of  $n_0$  and  $\lambda$  (Section 13.2.4). To understand the optimal tradeoff of

$n_0$  and  $\lambda$  to maximize estimator efficiency as functions of  $n$ ,  $p$ , the variance-covariance matrix of the covariates, and how strong the signal of  $f$  is to the noise will involve a lot of theory. Also, perhaps a variable rule for  $\lambda$  would be effective: if the sample size is large, the algorithm can afford to be conservative about the matches during the beginning of the experiment, but then become less conservative as time passes. Waiting until  $n_0$  is almost half  $n$  can be another strategy.

## Supplementary Materials

Original data and source code for reproduction can be found at [github.com/kapelner/sequential\\_matching\\_simulations\\_2](https://github.com/kapelner/sequential_matching_simulations_2).

## Acknowledgements

We wish to thank Larry Brown, Dean Foster, John Horton, Stephen Kapelner, Katherine Propert, Paul Rosenbaum, Andrea Troxel, and the reviewers for helpful suggestions and Hugh MacMullan for help with grid computing. Adam Kapelner also acknowledges the National Science Foundation for his graduate research fellowship.

## List of Figures

2.1	Main task portal for a subject in the meaningful treatment . . . . .	16
2.2	Example preferences as a function of task meaningfulness . . . . .	19
3.1	Display of Question B for all treatments . . . . .	39
3.2	Screenshot of the <i>instructional manipulation check</i> . . . . .	40
4.1	The <b>DistributeEyes</b> Components and their interactions . . . . .	62
4.2	<b>DistributeEyes</b> Project Settings Dialog Window . . . . .	63
4.3	Worker submission check window . . . . .	64
4.4	The training video and example quiz questions . . . . .	66
4.5	The <b>HTML</b> training application within MTurk . . . . .	67
4.6	Cropped selections from the project images . . . . .	74
4.7	Putative beta fit and Q-Q for the beta distribution . . . . .	77
4.8	Posterior of $N$ for each project (truth assumed unknown) . . . . .	79
4.9	Posterior of $N$ for each project (truth assumed known) . . . . .	82
4.10	Accuracy regressed on training path length and time spent training .	83
5.1	An example of the WSD task that appears inside an MTurk HIT . .	90
5.2	Predicted accuracy vs. number of senses for a sample of the words . .	94
5.3	Accuracy of all 595 Turkers in this study . . . . .	96
6.1	Depiction of the cascaded message-to-user prediction model . . . . .	108
6.2	Correlation and 2 topics negatively correlated with $SWL$ . . . . .	116
7.1	Model creation times as a function of sample size . . . . .	130
7.2	Summary for the <b>bartMachine</b> model on the automobile data . . . . .	136
7.3	Out-of-sample predictive performance by number of trees . . . . .	137
7.4	Test of normality of errors . . . . .	139
7.5	Convergence diagnostics for the cross-validated <b>bartMachine</b> model .	140
7.6	Fitted versus actual response values for the automobile dataset . . . .	143
7.7	Average variable inclusion proportions in the automobile data . . . . .	144

7.8	Tests of covariate importance in the automobile dataset . . . . .	146
7.9	Partial Dependence Plots for two features in the automobile dataset .	147
7.10	Visualization of the three variable selection procedures . . . . .	151
7.11	The top 10 average variable interaction counts . . . . .	155
7.12	Test of covariate importance for a Diabetes predictor . . . . .	158
7.13	Partial dependence plot for a Diabetes predictor . . . . .	159
8.1	Posterior draws for a variety of new observations . . . . .	176
8.2	Simulation results of the response model for the three MDM's . . . .	180
8.3	Simulations of oosRMSE by different probabilities of missingness . . .	184
9.1	Variable inclusion proportions in the null setting . . . . .	197
9.2	Variance decomposition simulation experiment results . . . . .	200
9.3	The fifteen largest variable inclusion proportions for a gene at $\alpha = .05$	204
9.4	$F_1$ results for the variable selection approaches in the linear model . .	212
9.5	$F_1$ results for the variable selection approaches in the non-linear model	215
9.6	$F_1$ measures under three different prior choices . . . . .	218
9.7	The number of predictor variables selected for each method . . . . .	223
9.8	RMSE reduction per predictor for each method . . . . .	224
9.9	Number of genes for which each TF was selected . . . . .	226
10.1	BART's and HBART's posterior mean estimates . . . . .	240
10.2	Estimates of the conditional mean function $f$ for HBART and BART . .	243
10.3	Distribution of out-of-sample RMSE for BART and HBART . . . . .	244
10.4	Posterior mean estimates for BART and HBART (Lidar data) . . . . .	245
10.5	Posterior predictive intervals for many methods (motorcycle data) . .	247
11.1	Scatterplot and PDP of $X_2$ versus $Y$ for the DGP of Equation 11.3 .	255
11.2	SGB ICE plot for $X_2$ from the DGP of Eq. 11.3 . . . . .	256
11.3	RF ICE plot for BHD for predictor <b>age</b> . . . . .	257
11.4	c-ICE plot for <b>age</b> ( $x^*$ set to the minimum value of <b>age</b> ) . . . . .	258
11.5	d-ICE plot for predictor <b>age</b> . . . . .	260
11.6	c-ICE plot for <b>age</b> colored by <b>rm</b> . . . . .	261
11.7	ICE and d-ICE plots for $X_1$ when $\hat{f}$ is a GAM . . . . .	263
11.8	SGB ICE plots for the DPG of Eq. 11.6 . . . . .	264
11.9	RF ICE plots for $x_1$ of the DPG in Eq. 11.7 . . . . .	267
11.10	BART ICE plots of <b>treatment</b> on <b>depression</b> colored by <b>married</b> . .	269
11.11	NN ICE plots of <b>pH</b> on <b>rating</b> colored by <b>alcohol</b> . . . . .	270
11.12	RF ICE plots of <b>skin</b> on probability of diabetes versus colored by <b>age</b> .	271
11.13	Additivity lineup test for <b>treatment</b> in the depression dataset . . . .	275
11.14	Additivity lineup test for <b>pH</b> in the white wine dataset . . . . .	276
12.1	Illustration of samples being drawn from the population . . . . .	280
12.2	External validity versus internal validity tradeoff . . . . .	281
12.3	Splash page of the experimental HIT . . . . .	284
12.4	Part of the survey page in the experimental HIT . . . . .	284

12.5	Diagram of the multi-stage experiment layout . . . . .	286
12.6	The screen shown to subjects in the “S” wing post survey . . . . .	288
12.7	The design matrix for each of the $k$ studies. . . . .	289
12.8	Box and whisker plots for the framing study . . . . .	292
12.9	Sample proportions of cooperate in the priming study . . . . .	292
12.10	Box and whisker plots for the sunk cost . . . . .	293
13.1	Results for power at $\alpha = 0.05$ with matching propensity $\lambda = 10\%$ . .	310
13.2	Illustration of the sequential procedure on historical data . . . . .	317
A.1	The HIT as initially encountered on MTurk . . . . .	328
A.2	The colorblindness test . . . . .	329
A.3	Opening screen of training video . . . . .	330
A.4	Examples of meaningful cues . . . . .	331
A.5	Describing the training process . . . . .	331
A.6	The quiz after watching the training video . . . . .	333
A.7	The training interface in all treatments . . . . .	334
A.8	The landing page after a labeling task is completed . . . . .	335
A.9	The survey after completion of the HIT . . . . .	336
A.10	Power illustrated for matching parameter $\lambda = 1\%$ . . . . .	371
A.11	Power illustrated for matching parameter $\lambda = 2.5\%$ . . . . .	372
A.12	Power illustrated for matching parameter $\lambda = 5\%$ . . . . .	373
A.13	Power illustrated for matching parameter $\lambda = 7.5\%$ . . . . .	374
A.14	Power illustrated for matching parameter $\lambda = 10\%$ . . . . .	375
A.15	Power illustrated for matching parameter $\lambda = 20\%$ . . . . .	376
A.16	Power illustrated for matching parameter $\lambda = 35\%$ . . . . .	377
A.17	Power illustrated for matching parameter $\lambda = 50\%$ . . . . .	378

## List of Tables

2.1	Summary statistics for response variables and demographics . . . . .	20
2.2	A heatmap illustration of our results . . . . .	21
2.3	Results for the treatment effects on quantity of images . . . . .	23
2.4	Results for the treatment effects on quality of labeling . . . . .	26
3.1	Overview of treatments and how they improve data quality . . . . .	35
3.2	Summary statistics by treatment . . . . .	43
3.3	Attrition by treatment . . . . .	44
3.4	IMC pass rate by treatment (no covariates) . . . . .	45
3.5	IMC pass rate by treatment (with covariates) . . . . .	46
3.6	Results for Question A by treatment . . . . .	48
3.7	Results for Question B by treatment . . . . .	53
4.1	Averages and standard errors of $F1$ scores over all workers . . . . .	75
4.2	Results for true positives being beta-distributed . . . . .	78
4.3	The raw features used to discriminate a cluster . . . . .	81
5.1	The OntoNotes sample of 89 words used in this study . . . . .	89
5.2	Regression results of correctness on word, context, and senses features	93
5.3	Accuracy using plurality voting for different numbers of Turkers . . .	97
6.1	Examples of statuses with contrastive values . . . . .	109
6.2	User-level prediction scores as the Pearson correlation coefficient . . .	113
6.3	Message-level prediction scores across PERMA and SWL . . . . .	115
8.1	MDM models in the context of statistical learning . . . . .	166
8.2	Missingness scenarios for the BHD simulations . . . . .	182
9.1	The Three Nested Variances . . . . .	198
9.2	Distribution of prior influence values $c$ used across the 6026 genes. . .	225



12.1	Power by number of duplicates and interaction effect size . . . . .	290
12.2	Experimental sample sizes . . . . .	291
12.3	Experimental estimates of ATE . . . . .	291
13.1	Response models for the three scenarios proposed . . . . .	308
13.2	Balance and relative sample efficiency results . . . . .	311
13.3	Simulated sizes for all scenarios, competitors, and tests . . . . .	314
13.4	Results for the sequential matching procedure for the behavioral data	315
13.5	Results for the sequential matching procedure for the clinical trial data	318
A.1	Out-of-sample RMSE values for 9 datasets . . . . .	349
A.2	Average absolute bias of sequential matching and post matching . . .	370

## A.1 Supplement for Chapter 2

### A.1.1 Detailed Experimental Design

This section details exact screens shown to users in the experimental groups. The worker begins by encountering the HIT on the MTurk platform (see Figure A.1).



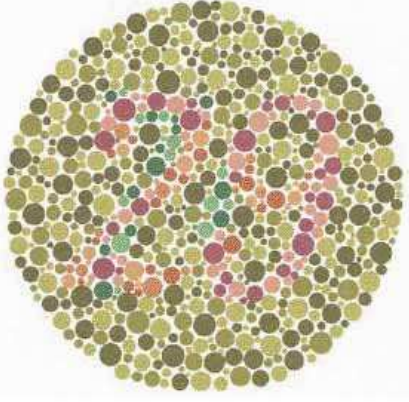
Figure A.1: The HIT as initially encountered on MTurk. Note: we used an alias in order to appear as a non-corporate and non-institutional employer.

The worker can then click on the HIT and they see the “preview screen” which describes the HIT (not shown) with text. In retrospect, a flashy image enticing the worker into the HIT would most likely have increased throughput. If the worker

chooses to accept, they are immediately directed to a multi-purpose page which hosts a colorblindness test, demographic survey, and an audio test for functioning speakers (see Figure A.2). Although many tasks require workers to answer questions before working, we avoided asking too many survey-like questions to avoid appearing as an experiment.

**Since the tasks you will perform require you to be able to differentiate color, we have to ask you a few questions that will determine if you may be colorblind.**

1. Look at the below image.



Do you see a number? If so, enter it into this box:

2. Are you male or female?  
☐ Male ☐ Female

3. Have you ever had trouble differentiating between reds and greens?  
☐ Yes ☐ No

4. Have you ever had trouble differentiating between blues and yellows?  
☐ Yes ☐ No

5. How old are you?


6. Listen to the following sound clip (  ) and enter the word below:

Figure A.2: The colorblindness test.

At this point, the worker is randomized into one of the three treatments and

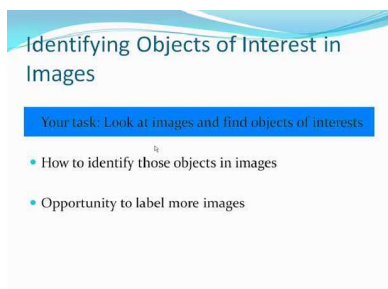
transitioned to the “qualification test.” The page displays an instructional video varying by treatment which they cannot fast-forward. Screenshots of the video are shown in Figures A.3, A.4, and A.5.<sup>1</sup>

We include the verbatim script for the videos below. Text that differs between treatments is typeset in square brackets separated by a slash. The text before the slash in red belongs to the meaningful treatment and the text following the slash in blue belongs to both the zero-context and shredded treatments.

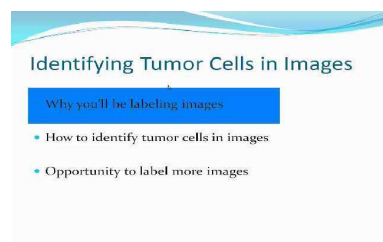
Thanks for participating in this task. [Your job will be to help identify tumor cells in images and we appreciate your help. / In this task, you’ll look at images and find objects of interest.]

In this video tutorial, we’ll explain [three / two] things:

[First, why you’re labeling the images, which is to help researchers identify tumorous cancer cells. Next, we’ll show you how to identify those tumor cells. / First, we’ll show you how to identify objects of interest in images.] [Finally, / Then,] we’ll explain how after labeling your first image you’ll have a chance to label some more.



(a) Zero-context / Shredded treatments



(b) Meaningful treatment

Figure A.3: Opening screen of training video.

Now we’re ready to learn how to identify [tumor cells / objects of interest] in images. Some example pictures of the [tumor cells / objects of interest] you’ll be identifying can be found at the bottom left. Each [tumor cell / object of interest] is blue and circular and surrounded by a red border.

When you begin each image, the magnification will be set to the lowest resolution. This gives you an overview of all points on the image, but you’ll need to zoom in and out in order to make the most precise clicks in the center of the [tumor cells / objects of interest].

Let’s scroll through the image and find some [tumor cells / objects of interest] to identify.

---

<sup>1</sup>We thank Rob Cohen who did an excellent job narrating both scripts.

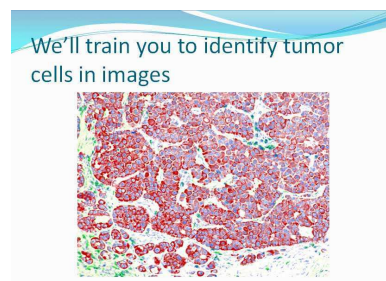
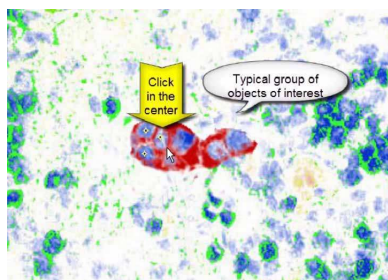
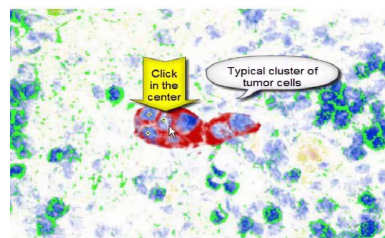


Figure A.4: Examples of meaningful cues which are not present in the Zero-context and Shredded treatment instructional video.



(a) Zero-context / Shredded treatments



(b) Meaningful treatment

Figure A.5: Describing the training process.

Here's a large cluster of [tumor cells / objects of interest]. To identify them, it is very important to click as closely to the center as possible on each [cell / object] . If I make a mistake and don't click in the center, I can undo the point by right-clicking.

Notice that this [cell / point] isn't entirely surrounded by red, [probably because the cell broke off]. Even though it's not entirely surrounded by red, we still want to identify it as a [tumor cell / object of interest].

In order to ensure that you've located all [tumor cells / objects of interest], you should use the thumbnail view in the top right. You can also use the magnification buttons to zoom out.

It looks like we missed a cluster of [tumor cells / objects of interest] at the bottom. Let's go identify those points.

Remember once again, that if you click on something that is not a [tumor cell / object of interest], you can unclick by right-clicking.

Using the scroll bars, we'll navigate to the other points ... and here's some more to the left ... Now that we think we've identified all points, let's zoom out to be sure and scroll around.

Before submitting, we should be sure of three things: (1) That we've identified all [tumor cells / objects of interest] (2) That we've clicked in the center of each one (3) That we haven't clicked on anything that's not a [tumor cell / object of interest].

Once we've done that, we're ready to submit.

Finally, after you complete your first image, you'll have an opportunity to label additional images as part of this HIT.

The first images you label will pay more to compensate for training.

After that, as part of this HIT you'll have the chance to identify as many additional images as you like as long as you aren't taking more than 15 minutes per image.

Although you can label unlimited images in this HIT, you won't be able to accept more HITs. This is to give a variety of turkers an opportunity to identify the images.

[Thank you for your time and effort. Advances in the field of cancer and treatment prevention rely on the selfless contributions of countless individuals such as yourself.]

Then, workers must take a quiz (see Figure A.6). During the quiz, they can watch the video freely (which was rarely done).

Upon passing, they began labeling their first image (see Figure A.7). The training interface includes the master training window where workers can create and delete points and scroll across the entire image. To the left, there is a small image displaying example tumor cells. Above the master window, they have zoom in / out buttons. And on the top right there is a thumbnail view of the overall image.

Participants were given 15 minutes to mark an image. Above the training window, we displayed a countdown timer that indicated the amount of time left. The participant's total earnings was also prominently displayed atop. On the very top, we provided a submit button that allowed the worker to submit results at any time.

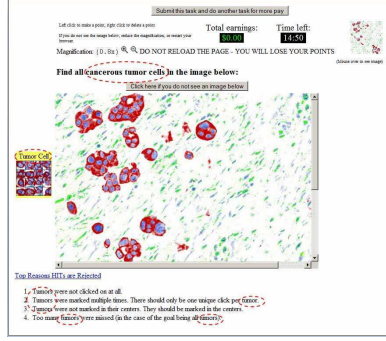
Each image had the same 90 cells from various-sized clusters. The cell clusters were selected for their unambiguous examples of cells, thereby eliminating the difficulty of training the difficult-to-identify tumor cells. In each image, the same clusters were arranged and rotated haphazardly, then pasted on one of five different believable backgrounds using Adobe Photoshop. Those clusters were then further rotated to create a set of ten images. This setup guarantees that the difficulty was relatively the same image-image. Images were displayed in random order for each worker, repeating after each set of ten (repetition was not an issue since it was rare for a participant to

Please answer the below questions. Once you answer these questions, you will be qualified to help identify tumor cells.

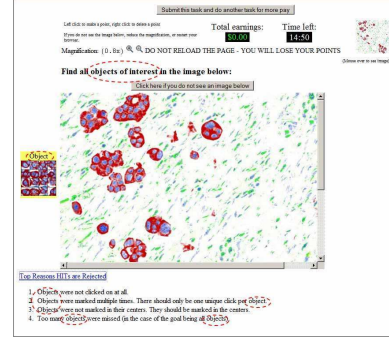
- 1 - You should adjust the magnification in order to...
  - ☐ Make a prettier picture
  - ☐ Make the tumors exactly 10 pixels across
  - ☒ Find tumors and make clicks as close to the center as possible
- 2 - When you are clicking on tumor cells, how many times do you click on the tumor?
  - ☐ Once
  - ☒ Twice
  - ☐ As many dots as you can fit inside the tumor
- 3 - When you incorrectly click on an area, you should...
  - ☐ Give up the HIT
  - ☐ Reload the page
  - ☐ Use the right mouse button
- 4 - What will happen if you don't accurately click on the tumor?
  - ☐ Scientists who are depending on you to identify tumors will not have accurate results
  - ☐ Your HIT may be rejected
  - ☐ You will not be allowed to do additional HITs with us
  - ☐ All of the above.
- 5 - Your HIT will be rejected if...
  - ☐ You do not click on all the tumors
  - ☐ You don't click in the center of the tumor
  - ☐ You click multiple times on the same tumor or on things that are NOT tumors
  - ☐ All of the above.
- 6 - As part of this HIT, how many images will you have the chance to identify assuming you correctly label the tumors?
  - ☐ As many as you want within a 4hr period (as long as you complete each image within 15 minutes)
  - ☐ You can label up to 4 more images
  - ☐ You must submit since you can only label one image

Begin Task

Figure A.6: The quiz after watching the training video for the meaningful treatment. In the zero-context and shredded treatments, all mention of “tumor cells” are replaced by “objects of interest.” The shredded treatment has an additional question asking them to acknowledge that they are working on a test system and their work will be discarded. Green indicates a correct response; red indicates an incorrect response.



(a) Zero-context / Shredded treatments



(b) Meaningful treatment

Figure A.7: The training interface as seen by workers. The meaningful interface reminds the subjects in 8 places that they are identifying tumor cells. The zero-context interface only says “objects of interest” and the shredded condition in addition has a message in red indicating that their points will not be saved (unshown). The circles around each point were *not* visible to participants. We display them to illustrate the size of a 10-pixel radius.

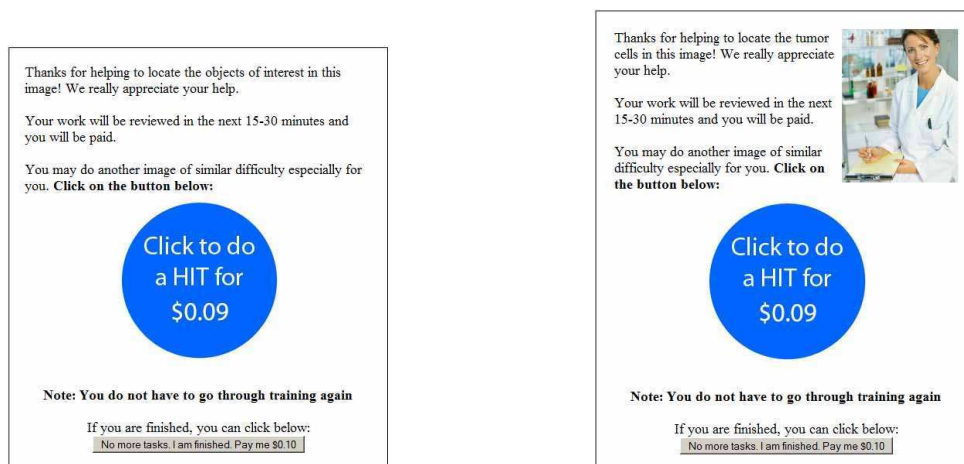
label more than ten).

After the worker is finished labeling, the worker presses submit and they are led to an intermediate page which asks if they would like to label another image and the new wage is prominently displayed (see Figure A.8). In the meaningful treatment, we add one last cue of meaning — a stock photo of a researcher to emphasize the purpose of the task. In the shredded treatment, we append the text “NONE of your points will be saved because we are testing our system, but you will still be paid.” If the worker wishes to continue, they are led to another labeling task; otherwise, they are directed to the post manipulation check survey shown in figure A.9.

The program ensures that the worker is being honest. We require them to find more than 20% of the cells (the workers were unaware that we were able to monitor their accuracy). If they are found cheating on three images, they are deemed *fraudulent* and not allowed to train more images. Since payment is automatic, this is to



protect us from a worker depleting our research account. In practice, this happened rarely and was not correlated with treatment.



(a) Zero-context / Shredded treatments

(b) Meaningful treatment

Figure A.8: The landing page after a labeling task is completed. At this point, workers are asked if they'd like to label another image or quit and be paid what they've earned so far.

The majority of data (70%) was collected over three days (January 5-8, 2010). However, since Indians in our sample had high attrition, we collected the remaining 30% from January 29 - February 2. In order to be balanced on time-of-day, we posted tasks so that the local time in the US and India would be as similar as possible. TK -we don't need this anymore, it's not common to talk about dates of data collection anyway.

Thanks for all of your work. In order to improve our HIT, please complete the following OPTIONAL feedback form.

If you do not want to fill in the survey, please click here:

[Submit to MTurk.](#)

Please rate how much you agree/disagree with the following statements (where 1-strongly disagree, 5-strongly agree):

The task was fun/enjoyable

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

I liked that the task seemed to be useful and had a good purpose

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

I felt good completing the task

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

The task seemed a lot more meaningful than the average MTurk HITs

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

The task was well-designed and respected my efforts and work more than the average MTurk HITs

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Any other comments:

[Submit my survey](#)

Figure A.9: The survey a subject fills out upon completion of the task.

## **A.1.2 A Technical Guide to Running Field Experiments on Mechanical Turk**

### **Institutional Review Board (IRB) Requirements**

This study requires the use of deception in order to observe social preferences in a natural environment and thus is not exempted under category 2's survey procedures. The issue is you cannot give the subjects an initial consent form indicating that they are part of an experiment.

Upon waiving the requirement of consent, the IRB will most likely require you to issue a debrief statement to your subjects stating that they were part of an experiment, a blurb about the purpose of the experiment, and contact information to your institution's IRB. In order for the experiment to work properly, you can only issue the debriefing *after* data collection is completed. Otherwise, MTurk subjects can communicate to each other that this is a study and this may be a problem for internal validity.

### **Engineering Required**

The implementation of an experiment of similar scale to the one we describe in the paper requires no more than two weeks of full-time work for an experienced software engineer.

It is critical that the engineer be fluent in "front-end" design. The front-end is what your subjects will use throughout the experiment and it can be highly dynamic, responding to the individual participant's actions. MTurk tasks are rendered in HTML and CSS. Javascript controls the dynamism and AJAX provides smooth client-server communication. Although not used in this study, the front-end can become even more fancy by implementing Adobe Flash or Microsoft Silverlight applets.

An MTurk experiment also requires a back-end web stack consisting of an http

server, a database, and a server-side platform. The back-end's function is to render webpages and store the experiment's data. We recommend Ruby on Rails 3.1 because of its rapid development speed, inexpensive and instant deployment to rented space in "the Cloud," and because it is free and open-source.

Since the server must communicate with MTurk, it is convenient for the engineer to also have experience using the Amazon web services application programming interface (MTurk's API) as well as CRON jobs on Linux.

We recommend setting up two cron jobs for effective experimentation:

- (a) A CRON that creates HITs every 15 minutes. In order for subjects to see your task, it must remain fresh on Amazon's main page. Creating tasks every 15 min with short expiration times (an hour or so) will allow for maximum throughput.
- (b) Another CRON that automatically pays workers. We recommend doing this every hour or so. It's important that this be done automatically including the bonuses and rejections otherwise it will be cumbersome for the experimenter to check through each and every assignment.

Apart from technical skills, the engineer should also have knowledge of the principles of experimental design.

### **A.1.3 Instructions for Replication of this Study**

- (a) Obtain the source from [http://github.com/kapelner/breaking\\_monotony](http://github.com/kapelner/breaking_monotony).
- (b) Create a Ruby on Rails hosting account on EngineYard cloud at <http://cloud.engineyard.com> and configure for Rails 3.1.
- (c) Modify the `PersonalInformation.rb` file by adding your MTurk account information, server IP address, and the administrator login and password.

- (d) Deploy the application and setup the two CRON jobs recommended in the previous section.

## A.2 Supplement for Chapter 7

### A.2.1 Sampling New Trees

This section provides details on the implementation of Equation A.14 (steps 1, 3, ...,  $2m - 1$ ), the Metropolis-Hastings step for sampling new trees. Recall from Section 7.2.2 that trees can be altered via growing new daughter nodes from an existing terminal node, pruning two terminal nodes such that their parent becomes terminal, or changing the splitting rule in a node.

Below is the Metropolis ratio (Gelman et al., 2004, p.291) where the parameter sampled is the tree and the data is the responses unexplained by other trees denoted by  $\mathbf{R}$ . We denote the new, proposal tree with an asterisk and the original tree without the asterisk.

$$r = \frac{\mathbb{P}(\mathfrak{T}_* \rightarrow \mathfrak{T}) \mathbb{P}(\mathfrak{T}_* \mid \mathbf{R}, \sigma^2)}{\mathbb{P}(\mathfrak{T} \rightarrow \mathfrak{T}_*) \mathbb{P}(\mathfrak{T} \mid \mathbf{R}, \sigma^2)} \quad (\text{A.1})$$

We accept a draw from the posterior distribution of trees if a draw from a standard uniform distribution is less than the value of  $r$ . Immediately we note that it is difficult (if not impossible) to calculate the posterior probabilities for the trees themselves. Instead, we employ Bayes' Rule,

$$\mathbb{P}(\mathfrak{T} \mid \mathbf{R}, \sigma^2) = \frac{\mathbb{P}(\mathbf{R} \mid \mathfrak{T}, \sigma^2) \mathbb{P}(\mathfrak{T} \mid \sigma^2)}{\mathbb{P}(\mathbf{R} \mid \sigma^2)},$$

and plug the result into Equation A.22 to obtain:

$$r = \underbrace{\frac{\mathbb{P}\left(\mathfrak{T}_* \rightarrow \mathfrak{T}\right)}{\mathbb{P}\left(\mathfrak{T} \rightarrow \mathfrak{T}_*\right)}}_{\text{transition ratio}} \times \underbrace{\frac{\mathbb{P}\left(\mathbf{R} \mid \mathfrak{T}_*, \sigma^2\right)}{\mathbb{P}\left(\mathbf{R} \mid \mathfrak{T}, \sigma^2\right)}}_{\text{likelihood ratio}} \times \underbrace{\frac{\mathbb{P}\left(\mathfrak{T}_*\right)}{\mathbb{P}\left(\mathfrak{T}\right)}}_{\text{tree structure ratio}}.$$

Note that the probability of the tree structure is independent of  $\sigma^2$ .

The goal of this section is to explicitly calculate  $r$  for all possible tree proposals — GROW, PRUNE and CHANGE. For each proposal, the calculations are organized into separate sections detailing each of the three ratios — transition, likelihood and tree structure. Note that our actual implementation uses the following expressions in log form for numerical accuracy.

## A.2.2 Grow Proposal

### Transition Ratio

Transitioning from the original tree to a new tree involves growing two daughter nodes from a current terminal node:

$$\begin{aligned} \mathbb{P}\left(\mathfrak{T} \rightarrow \mathfrak{T}_*\right) &= \mathbb{P}(\text{GROW}) \mathbb{P}(\text{selecting } \eta \text{ to grow from}) \times \\ &\quad \mathbb{P}(\text{selecting the } j\text{th attribute to split on}) \times \\ &\quad \mathbb{P}(\text{selecting the } i\text{th value to split on}) \\ &= \mathbb{P}(\text{GROW}) \frac{1}{b} \frac{1}{p_{\text{adj}}(\eta)} \frac{1}{n_{j \cdot \text{adj}}(\eta)}. \end{aligned} \tag{A.2}$$

We chose one of the current  $b$  terminal nodes which we denote the  $\eta$ th node, and then we pick an attribute and split point.  $p_{\text{adj}}(\eta)$  denotes the number of predictors left available to split on. This can be less than  $p$  if certain predictors do not have

two or more unique values once the data reaches the  $\eta$ th node. For example, this regularly occurs if a dummy variable was split on in some node higher up in the lineage.  $n_{j\cdot\text{adj}}(\eta)$  denotes the number of *unique* values left in the  $p$ th attribute after adjusting for parents' splits.

Transitioning from the new tree back to the original tree involves pruning that node:

$$\mathbb{P}\left(\mathfrak{T}_* \rightarrow \mathfrak{T}\right) = \mathbb{P}(\text{PRUNE}) \mathbb{P}(\text{selecting } \eta \text{ to prune from}) = \mathbb{P}(\text{PRUNE}) \frac{1}{w_2^*}$$

where  $w_2^*$  denotes the number of second generation internal nodes (nodes with two terminal daughter nodes) in the new tree. Thus, the full transition ratio is:

$$\frac{\mathbb{P}\left(\mathfrak{T}_* \rightarrow \mathfrak{T}\right)}{\mathbb{P}\left(\mathfrak{T} \rightarrow \mathfrak{T}_*\right)} = \frac{\mathbb{P}(\text{PRUNE})}{\mathbb{P}(\text{GROW})} \frac{b \, p_{\text{adj}}(\eta) \, n_{j\cdot\text{adj}}(\eta)}{w_2^*}.$$

Note that when there are no variables with more two or more unique values, the probability of GROW is set to zero and the step will be automatically rejected.

## Likelihood Ratio

To calculate the likelihood, the tree structure determines which responses fall into which of the  $b$  terminal nodes. Thus,

$$\mathbb{P}\left(R_1, \dots, R_n \mid \mathfrak{T}, \sigma^2\right) = \prod_{\ell=1}^b \mathbb{P}\left(R_{\ell_1}, \dots, R_{\ell_{n_\ell}} \mid \sigma^2\right)$$

where each term on the right hand side is the probability of responses in one of the  $b$  terminal nodes, which are independent by assumption. The  $R_\ell$ 's denote the data

in the  $\ell$ th terminal node and where  $n_\ell$  denotes how many observations are in each terminal node and  $n = \sum_{\ell=1}^b n_\ell$ .

We now find an analytic expression for the node likelihood term. Remember, if the mean in each terminal node, which we denote  $\mu_\ell$ , was known, then we would have  $R_{\ell_1}, \dots, R_{\ell_{n_\ell}} \mid \mu_\ell, \sigma^2 \stackrel{iid}{\sim} \mathcal{N}(\mu_\ell, \sigma^2)$ . BART requires  $\mu_\ell$  to be margined out, allowing the Gibbs sampler in Equation A.14 to avoid dealing with jumping between continuous spaces of varying dimensions (Chipman et al., 2010, page 275). Recall that one of the BART model assumptions is a prior on the average value of  $\mu \sim \mathcal{N}(0, \sigma_\mu^2)$  and thus,

$$\mathbb{P}(R_{\ell_1}, \dots, R_{\ell_{n_\ell}} \mid \sigma^2) = \int_{\mathbb{R}} \mathbb{P}(R_{\ell_1}, \dots, R_{\ell_{n_\ell}} \mid \mu_\ell, \sigma^2) \mathbb{P}(\mu_\ell; \sigma_\mu^2) d\mu_\ell$$

which can be shown via completion of the square or convolution to be

$$\begin{aligned} \mathbb{P}(R_{\ell_1}, \dots, R_{\ell_{n_\ell}} \mid \sigma^2) &= \frac{1}{(2\pi\sigma^2)^{n_\ell/2}} \sqrt{\frac{\sigma^2}{\sigma^2 + n_\ell\sigma_\mu^2}} \times \\ &\exp\left(-\frac{1}{2\sigma^2} \left(\sum_{i=1}^{n_\ell} (R_{\ell_i} - \bar{R}_\ell)^2 - \frac{\bar{R}_\ell^2 n_\ell^2}{n_\ell + \frac{\sigma^2}{\sigma_\mu^2}} + n_\ell \bar{R}_\ell^2\right)\right) \end{aligned} \quad (\text{A.3})$$

where  $\bar{R}_\ell$  denotes the mean response in the node and  $R_{\ell_i}$  denotes the observations  $i = 1 \dots n_\ell$  in the node.

Since the likelihoods are solely determined by the terminal nodes, the proposal tree differs from the original tree by only the selected node to be grown, denoted by  $\ell$ , which becomes two daughters after the GROW step denoted by  $\ell_L$  and  $\ell_R$ . Hence, the likelihood ratio becomes:

$$\frac{\mathbb{P}(\mathbf{R} \mid \mathfrak{F}_*, \sigma^2)}{\mathbb{P}(\mathbf{R} \mid \mathfrak{F}, \sigma^2)} = \frac{\mathbb{P}(R_{\ell_{L,1}}, \dots, R_{\ell_{L,n_{\ell,L}}} \mid \sigma^2) \mathbb{P}(R_{\ell_{R,1}}, \dots, R_{\ell_{R,n_{\ell,R}}} \mid \sigma^2)}{\mathbb{P}(R_{\ell_1}, \dots, R_{\ell_{n_\ell}} \mid \sigma^2)} \quad (\text{A.4})$$



Plugging Equation A.23 into Equation A.24 three times yields the ratio for the GROW step:

$$\sqrt{\frac{\sigma^2 (\sigma^2 + n_\ell \sigma_\mu^2)}{(\sigma^2 + n_{\ell_L} \sigma_\mu^2) (\sigma^2 + n_{\ell_R} \sigma_\mu^2)}} \exp \left( \frac{\sigma_\mu^2}{2\sigma^2} \left( \frac{(\sum_{i=1}^{n_{\ell_L}} R_{\ell_L,i})^2}{\sigma^2 + n_{\ell_L} \sigma_\mu^2} + \frac{(\sum_{i=1}^{n_{\ell_R}} R_{\ell_R,i})^2}{\sigma^2 + n_{\ell_R} \sigma_\mu^2} - \frac{(\sum_{i=1}^{n_\ell} R_{\ell,i})^2}{\sigma^2 + n_\ell \sigma_\mu^2} \right) \right)$$

where  $n_{\ell_L}$  and  $n_{\ell_R}$  denote the number of data points in the newly grown left and right daughter nodes.

### Tree Structure Ratio

In Section 7.2.1 we discussed the prior on the tree structure (where the splits occur) as well as the tree rules. For the entire tree,

$$\mathbb{P} \left( \frac{\mathfrak{F}}{\mathfrak{T}} \right) = \prod_{\eta \in H_{\text{terminals}}} (1 - \mathbb{P}_{\text{SPLIT}}(\eta)) \prod_{\eta \in H_{\text{internals}}} \mathbb{P}_{\text{SPLIT}}(\eta) \prod_{\eta \in H_{\text{internals}}} \mathbb{P}_{\text{RULE}}(\eta)$$

where  $H_{\text{terminals}}$  denotes the set of terminal nodes and  $H_{\text{internals}}$  denotes the internal nodes.

Recall that the probability of splitting on a given node  $\eta$  is  $\mathbb{P}_{\text{SPLIT}}(\eta) = \alpha / (1 + d_\eta)^\beta$ . The probability is controlled by two hyperparameters,  $\alpha$  and  $\beta$ , and  $d_\eta$  is the depth (number of parent generations) of node  $\eta$ . When assigning a rule, recall that **BART** picks from all available attributes and then from all available unique split points. Using the notation from the transition ratio section,  $\mathbb{P}_{\text{RULE}}(\eta) = 1/p_{\text{adj}}(\eta) \times 1/n_{j\text{-adj}}(\eta)$ .

Once again, the original tree features a node  $\eta$  that was selected to be grown. The proposal tree differs with two daughter nodes denoted  $\eta_L$  and  $\eta_R$ . We can now form the ratio:

$$\begin{aligned}
\frac{\mathbb{P}\left(\mathfrak{T}_*\right)}{\mathbb{P}\left(\mathfrak{T}\right)} &= \frac{(1 - \mathbb{P}_{\text{SPLIT}}(\eta_L))(1 - \mathbb{P}_{\text{SPLIT}}(\eta_R)) \mathbb{P}_{\text{SPLIT}}(\eta) \mathbb{P}_{\text{RULE}}(\eta)}{(1 - \mathbb{P}_{\text{SPLIT}}(\eta))} \\
&= \frac{\left(1 - \frac{\alpha}{(1 + d_{\eta_L})^\beta}\right) \left(1 - \frac{\alpha}{(1 + d_{\eta_R})^\beta}\right) \frac{\alpha}{(1 + d_\eta)^\beta} \frac{1}{p_{\text{adj}}(\eta)} \frac{1}{n_{j \cdot \text{adj}}(\eta)}}{1 - \frac{\alpha}{(1 + d_\eta)^\beta}} \\
&= \alpha \frac{\left(1 - \frac{\alpha}{(1 + d_\eta)^\beta}\right)^2}{\left((1 + d_\eta)^\beta - \alpha\right) p_{\text{adj}}(\eta) n_{j \cdot \text{adj}}(\eta)}
\end{aligned}$$

The last line follows from algebra and using the fact that the depth of the grown nodes is the depth of the parent node incremented by one ( $d_{\eta_L} = d_{\eta_R} = d_\eta + 1$ ).

### A.2.3 Prune Proposal

A prune proposal is the “opposite” of a grow proposal. Prune selects a node with two daughters and removes them. Thus, each ratio will be approximately the inverse of the ratios found in the previous section concerning the grow proposal. Note also that prune steps are not considered in trees that consist of a single root node.

#### Transition Ratio

We begin with transitioning from the original tree to the proposal tree:

$$\mathbb{P}\left(\mathfrak{T} \rightarrow \mathfrak{T}_*\right) = \mathbb{P}(\text{PRUNE}) \mathbb{P}(\text{selecting } \eta \text{ to prune from}) = \mathbb{P}(\text{PRUNE}) \frac{1}{w_2}$$

where  $w_2$  denotes the number of parent nodes that have two daughters but no grand-daughters. To transition in the opposite direction, we are obligated to grow from node  $\eta$ . This is similar to Equation A.2 except the proposed tree has one less terminal node

due to the pruning of the original tree, resulting in a  $1/(b-1)$  term:

$$\mathbb{P}\left(\mathfrak{T}_* \rightarrow \mathfrak{T}\right) = \mathbb{P}(\text{GROW}) \frac{1}{b-1} \frac{1}{p_{\text{adj}}(\eta^*)} \frac{1}{n_{j^* \cdot \text{adj}}(\eta^*)}.$$

Thus, the transition ratio is:

$$\frac{\mathbb{P}\left(\mathfrak{T}_* \rightarrow \mathfrak{T}\right)}{\mathbb{P}\left(\mathfrak{T} \rightarrow \mathfrak{T}_*\right)} = \frac{\mathbb{P}(\text{GROW})}{\mathbb{P}(\text{PRUNE})} \frac{w_2}{(b-1)p_{\text{adj}}(\eta^*)n_{j^* \cdot \text{adj}}(\eta^*)}.$$

### Likelihood Ratio

This is simply the inverse of the likelihood ratio for the grow proposal:

$$\begin{aligned} \frac{\mathbb{P}\left(\mathbf{R} \mid \mathfrak{T}_*, \sigma^2\right)}{\mathbb{P}\left(\mathbf{R} \mid \mathfrak{T}, \sigma^2\right)} &= \sqrt{\frac{(\sigma^2 + n_{\ell_L} \sigma_\mu^2)(\sigma^2 + n_{\ell_R} \sigma_\mu^2)}{\sigma^2(\sigma^2 + n_\ell \sigma_\mu^2)}} \times \\ &\exp\left(\frac{\sigma_\mu^2}{2\sigma^2} \left( \frac{(\sum_{i=1}^{n_\ell} R_{\ell,i})^2}{\sigma^2 + n_\ell \sigma_\mu^2} - \frac{(\sum_{i=1}^{n_{\ell_L}} R_{\ell_L,i})^2}{\sigma^2 + n_{\ell_L} \sigma_\mu^2} - \frac{(\sum_{i=1}^{n_{\ell_R}} R_{\ell_R,i})^2}{\sigma^2 + n_{\ell_R} \sigma_\mu^2} \right)\right). \end{aligned}$$

### Tree Structure Ratio

This is also simply the inverse of the tree structure ratio for the grow proposal:

$$\frac{\mathbb{P}\left(\mathfrak{T}_*\right)}{\mathbb{P}\left(\mathfrak{T}\right)} = \frac{\left((1 + d_\eta)^\beta - \alpha\right) p_{\text{adj}}(\eta^*) n_{j^* \cdot \text{adj}}(\eta^*)}{\alpha \left(1 - \frac{\alpha}{(2+d_\eta)^\beta}\right)^2}.$$

### A.2.4 Change

A change proposal involves picking an internal node and changing its rule by picking both a new available predictor to split on and a new valid split value among values of the selected predictor. Although this could be implemented for use in any internal node in the tree, for simplicity we limit our implementation to *singly* internal nodes: those that have two terminal daughter nodes and thus, no grand-daughters.

#### Transition Ratio

The transition to a proposal tree is below:

$$\begin{aligned} \mathbb{P}\left(\mathfrak{P} \rightarrow \mathfrak{P}_*\right) &= \mathbb{P}(\text{CHANGE}) \mathbb{P}(\text{selecting node } \eta \text{ to change}) \times \\ &\quad \mathbb{P}(\text{selecting the new attribute to split on}) \times \\ &\quad \mathbb{P}(\text{selecting the new value to split on}) \end{aligned}$$

When calculating the ratio, the first three terms are shared in both numerator and denominator. The probability of selecting the new value to split on will differ as different split features have different numbers of unique values available. Thus we are left with

$$\frac{\mathbb{P}\left(\mathfrak{P}_* \rightarrow \mathfrak{P}\right)}{\mathbb{P}\left(\mathfrak{P} \rightarrow \mathfrak{P}_*\right)} = \frac{n_{j^*.\text{adj}}(\eta^*)}{n_{j.\text{adj}}(\eta)}$$

where  $n_{j^*.\text{adj}}(\eta^*)$  is the number of split values available under the proposal tree's splitting rule and  $n_{j.\text{adj}}(\eta)$  is the number of split values available under the original tree's splitting rule.

## Likelihood Ratio

The proposal tree differs from the original tree only in the two daughter nodes of the selected change node. These two terminal nodes have the unexplained responses apportioned differently. Denote  $R_{1\cdot}$  as the residuals of the first daughter node and  $R_{2\cdot}$  as the unexplained responses in the second daughter node. Thus we begin with:

$$\frac{\mathbb{P}(\mathbf{R} \mid \mathfrak{F}_*, \sigma^2)}{\mathbb{P}(\mathbf{R} \mid \mathfrak{F}, \sigma^2)} = \frac{\mathbb{P}(R_{1^*,1}, \dots, R_{1^*,n_{1^*}} \mid \sigma^2) \mathbb{P}(R_{2^*,1}, \dots, R_{2^*,n_{2^*}} \mid \sigma^2)}{\mathbb{P}(R_{1,1}, \dots, R_{1,n_1} \mid \sigma^2) \mathbb{P}(R_{2,1}, \dots, R_{2,n_2} \mid \sigma^2)}$$

where the responses denoted with an asterisk are the responses in the proposal tree's daughter nodes.

Substituting Equation A.23 four times and using algebra, the following expression is obtained for the ratio:

$$\sqrt{\frac{\left(\frac{\sigma^2}{\sigma_\mu^2} + n_1\right) \left(\frac{\sigma^2}{\sigma_\mu^2} + n_2\right)}{\left(\frac{\sigma^2}{\sigma_\mu^2} + n_1^*\right) \left(\frac{\sigma^2}{\sigma_\mu^2} + n_2^*\right)}} \times \exp\left(\frac{1}{2\sigma^2} \left( \frac{(\sum_{i=1}^{n_{1^*}} R_{1^*,i})^2}{n_{1^*} + \frac{\sigma^2}{\sigma_\mu^2}} + \frac{(\sum_{i=1}^{n_{2^*}} R_{2^*,i})^2}{n_{2^*} + \frac{\sigma^2}{\sigma_\mu^2}} - \frac{(\sum_{i=1}^{n_1} R_{1,i})^2}{n_1 + \frac{\sigma^2}{\sigma_\mu^2}} - \frac{(\sum_{i=1}^{n_2} R_{2,i})^2}{n_2 + \frac{\sigma^2}{\sigma_\mu^2}} \right)\right)$$

which simplifies to

$$\exp\left(\frac{1}{2\sigma^2} \left( \frac{(\sum_{i=1}^{n_{1^*}} R_{1^*,i})^2 - (\sum_{i=1}^{n_{1^*}} R_{1,i})^2}{n_1 + \frac{\sigma^2}{\sigma_\mu^2}} + \frac{(\sum_{i=1}^{n_{2^*}} R_{2^*,i})^2 - (\sum_{i=1}^{n_{2^*}} R_{2,i})^2}{n_2 + \frac{\sigma^2}{\sigma_\mu^2}} \right)\right)$$

if the number of responses in the children do not change in the proposal ( $n_1 = n_1^*$  and  $n_2 = n_2^*$ ).

## Tree Structure Ratio

The proposal tree has the same structure as the original tree. Thus we only need to take into account the changed node's daughters:

$$\frac{\mathbb{P}\left(\frac{\mathfrak{P}}{\mathfrak{T}}_*\right)}{\mathbb{P}\left(\frac{\mathfrak{P}}{\mathfrak{T}}\right)} = \frac{(1 - \mathbb{P}_{\text{SPLIT}}(\eta_{1*}))(1 - \mathbb{P}_{\text{SPLIT}}(\eta_{2*})) \mathbb{P}_{\text{SPLIT}}(\eta_*) \mathbb{P}_{\text{RULE}}(\eta_*)}{(1 - \mathbb{P}_{\text{SPLIT}}(\eta_1)(1 - \mathbb{P}_{\text{SPLIT}}(\eta_2))) \mathbb{P}_{\text{SPLIT}}(\eta) \mathbb{P}_{\text{RULE}}(\eta)}.$$

The probability of splits remain the same because the daughter nodes are at the same depths. Thus we only need to consider the ratio of the probability of the rules. Once again, the probability of selecting the new value to split on will differ as different split features have different numbers of unique values available. We are left with

$$\frac{\mathbb{P}\left(\frac{\mathfrak{P}}{\mathfrak{T}}_*\right)}{\mathbb{P}\left(\frac{\mathfrak{P}}{\mathfrak{T}}\right)} = \frac{n_{j \cdot \text{adj}}(\eta)}{n_{j^* \cdot \text{adj}}(\eta^*)}.$$

Note that this is the inverse of the transition ratio. Hence, for the change step, only the likelihood ratio needs to be computed to determine the Metropolis-Hastings ratio  $r$ .

### A.2.5 Bakeoff

We baked off nine regression data sets and assessed out-of-sample RMSE using 10-fold cross-validation. The results are displayed in Table A.1.

We conclude that the implementation outlined in this paper performs approximately the same as the previous implementation with regards to predictive accuracy.

Dataset Name	<code>bartMachine</code>	<code>BayesTree</code>	<code>randomForest</code>
boston	4.66	4.59	4.58
triazine	0.04*	0.05	0.04
ozone	48.02	47.75	48.77
baseball	0.02	0.02*	0.02
wine.red	0.73	0.69*	0.76
ankara	2.62	2.61	3.46
wine.white	0.49	0.49	0.54
pole	4.00	3.65	11.56
compactiv	46.17*	47.92	40.74

Table A.1: Average of 20 replicates of out-of-sample RMSE values on 9 datasets using three machine learning algorithms. Asterisks indicate a significant difference between `bartMachine` and `BayesTree` at a significance level of 5% with a Bonferroni correction. Comparisons with `randomForest`'s performance were not conducted.

## A.3 Supplement for Chapter 9

### A.3.1 Pseudo-code for Variable Selection Procedures

---

**Algorithm 4** Local Threshold Procedure

---

```
Compute  $p_1, \dots, p_K$  ▷ Inclusion proportions from original data  
for  $i \leftarrow \{1, \dots, P\}$  do ▷  $P$  is the number of null permutations  
     $\mathbf{y}^* \leftarrow \text{Permute}(\mathbf{y})$   
    Run BART using  $\mathbf{y}^*$  as response  
    Compute  $p_{i1}^*, \dots, p_{iK}^*$  ▷ Inclusion proportions from permuted data  
end for  
for  $j \leftarrow \{1, \dots, K\}$  do  
     $q_j^* \leftarrow \text{Quantile}(p_{1j}^*, \dots, p_{Pj}^*, 1 - \alpha)$  ▷  $1 - \alpha$  quantile of  $\mathbf{x}_j$  permutation  
    distribution  
    if  $p_j > q_j^*$  then Include  $p_j$  in Vars end if  
end for  
return Vars
```

---



---

**Algorithm 5** Global Maximum Threshold Procedure

---

Compute  $p_1, \dots, p_K$  ▷ Inclusion proportions from original data

**for**  $i \leftarrow \{1, \dots, P\}$  **do** ▷  $P$  is the number of null permutations

$\mathbf{y}^* \leftarrow \text{Permute}(\mathbf{y})$

Run BART using  $\mathbf{y}^*$  as response

Compute  $p_{i1}^*, \dots, p_{iK}^*$  ▷ Inclusion proportions from permuted data

$g_i \leftarrow \text{Max}(p_{i1}^*, \dots, p_{iK}^*)$  ▷ Maximum of proportions from permuted data

**end for**

$g^* \leftarrow \text{Quantile}(g_1, \dots, g_P, 1 - \alpha)$  ▷  $1 - \alpha$  Quantile of maxima

**for**  $j \leftarrow \{1, \dots, K\}$  **do**

**if**  $p_j > g^*$  **then** Include  $p_j$  in Vars **end if**

**end for**

**return** Vars

---

---

**Algorithm 6** Global Standard Error Threshold Procedure

---

Compute  $p_1, \dots, p_K$  ▷ Inclusion proportions from original data

**for**  $i \leftarrow \{1, \dots, P\}$  **do** ▷  $P$  is the number of null permutations

$\mathbf{y}^* \leftarrow \text{Permute}(\mathbf{y})$

Run BART using  $\mathbf{y}^*$  as response

Compute  $p_{i1}^*, \dots, p_{iK}^*$  ▷ Inclusion proportions from permuted data

**end for**

**for**  $j \leftarrow \{1, \dots, K\}$  **do**

$m_j \leftarrow \text{Avg}(p_{1j}^*, \dots, p_{Pj}^*)$  ▷ Sample average of  $\mathbf{x}_j$  permutation distribution

$s_j \leftarrow \text{SD}(p_{1j}^*, \dots, p_{Pj}^*)$  ▷ Sample sd of  $\mathbf{x}_j$  permutation distribution

**end for**

$C^* \leftarrow \inf_{C \in \mathbb{R}^+} \left\{ \forall j, \quad \frac{1}{P} \sum_{i=1}^P \mathbb{1}_{p_{ij}^* \leq m_j + C \cdot s_j} > 1 - \alpha \right\}$  ▷ Simultaneous coverage

**for**  $j \leftarrow \{1, \dots, K\}$  **do**

**if**  $p_j > m_j + C^* \cdot s_j$  **then** Include  $p_j$  in Vars **end if**

**end for**

**return** Vars

---

---

**Algorithm 7** Cross-Validated Comparison of Threshold Procedures

---

Divide the data into  $K$  training-test splits

**for**  $k \leftarrow \{1, \dots, K\}$  **do**

**for** method  $\leftarrow \{\text{Local, Global Maximum, Global SE}\}$  **do**

$\text{Var}_{\text{method}} \leftarrow$  Selected variables using method on BART

$\text{BART}_{\text{method}} \leftarrow$  BART built from  $k^{\text{th}}$  training set using only  $\text{Var}_{\text{method}}$

$L2_{k,\text{method}} \leftarrow$   $L2$  error from  $\text{BART}_{\text{method}}$  on  $k^{\text{th}}$  test set

**end for**

**end for**

**for** method  $\leftarrow \{\text{Local, Global Maximum, Global SE}\}$  **do**

$L2_{\text{method}} \leftarrow \sum_{k=1}^K L2_{k,\text{method}}$        $\triangleright$  Aggregate  $L2$  error over entire training set

**end for**

    method\*  $\leftarrow \arg \min_{\text{method}} \{L2_{\text{method}}\}$        $\triangleright$  Choose the best method from the three

**return** Selected variables using method\* on BART on full training set

---

## A.4 Supplement for Chapter 10

### A.4.0.1 Gibbs Sampling Details for HBART

In the following sections, we provide implementation details for each step of the HBART Gibbs sampler (Equation A.14).

#### Drawing $\sigma^2$ (step $2m + 1$ )

Drawing  $\sigma^2$  in HBART requires a slight modification to the scheme used in the original homoskedastic BART. With the errors distributed multivariate normal (Equation 10.2) and the prior on  $\sigma^2$  being distributed inverse-gamma, conjugacy yields the posterior as an inverse-gamma distribution just as in BART. The quadratic form of the errors and their covariance matrix carries into the scale parameter. Hence, we sample  $\sigma^2$  as

$$\begin{aligned}
\sigma^2 &\sim \text{InvGamma} \left( \frac{\nu + n}{2}, \frac{\nu\lambda + \boldsymbol{\mathcal{E}}^T \boldsymbol{D}^{-1} \boldsymbol{\mathcal{E}}}{2} \right) \\
&= \text{InvGamma} \left( \frac{\nu + n}{2}, \frac{\nu\lambda}{2} + \frac{1}{2} \left( \sum_{i=1}^n \exp(-\mathbf{z}_i^\top \boldsymbol{\gamma}) \mathcal{E}_i^2 \right) \right).
\end{aligned}$$

The default values of hyperparameters  $\nu$  and  $q$  are set to be the same as the defaults of the original homoskedastic **BART** algorithm.

### Drawing $\boldsymbol{\gamma}$ (step $2m + 2$ )

The posterior of  $\boldsymbol{\gamma} \mid \mathcal{E}_1, \dots, \mathcal{E}_n, \sigma^2$  is proportional to:

$$\begin{aligned}
\mathbb{P}(\boldsymbol{\gamma} \mid \mathcal{E}_1, \dots, \mathcal{E}_n, \sigma^2) &\propto \mathbb{P}(\mathcal{E}_1, \dots, \mathcal{E}_n \mid \boldsymbol{\gamma}, \sigma^2) \mathbb{P}(\boldsymbol{\gamma}) \\
&= \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2 \exp(\mathbf{z}_i^\top \boldsymbol{\gamma})}} \exp \left( -\frac{1}{2\sigma^2 \exp(\mathbf{z}_i^\top \boldsymbol{\gamma})} \mathcal{E}_i^2 \right) \times \\
&\quad \prod_{j=1}^p \frac{1}{\sqrt{2\pi\Sigma_{jj}}} \exp \left( -\frac{1}{2\Sigma_{jj}} (\gamma_j - \gamma_{0j})^2 \right) \\
&\propto \frac{1}{\sqrt{\sigma^2 \exp(\sum_{i=1}^n \mathbf{z}_i^\top \boldsymbol{\gamma})}} \times \\
&\quad \exp \left( -\frac{1}{2} \left( \sum_{i=1}^n \frac{\mathcal{E}_i^2}{\sigma^2 \exp(\mathbf{z}_i^\top \boldsymbol{\gamma})} + \sum_{j=1}^p \frac{(\gamma_j - \gamma_{0j})^2}{\Sigma_{jj}} \right) \right)
\end{aligned}$$

Since it is not tractable to draw directly from the above distribution, a proposal distribution is required. A suitable proposal distribution can be obtained using Gamerman (1997), which is based on a single step of an iteratively reweighted least squares algorithm. Letting  $\boldsymbol{\gamma}$  represent the current value of the parameter vector, the recommended proposal density is  $\mathbb{P}(\boldsymbol{\gamma}^* \mid \boldsymbol{\gamma}) = \mathcal{N}_p(\mathbf{a}(\boldsymbol{\gamma}), \mathbf{B})$  where

$$\mathbf{a} = \mathbf{B} \left( \boldsymbol{\Sigma}^{-1} \boldsymbol{\gamma}_0 + \frac{1}{2} \mathbf{Z}^T \mathbf{w} \right), \quad \mathbf{B} = \left( \boldsymbol{\Sigma}^{-1} + \frac{1}{2} \mathbf{Z}^T \mathbf{Z} \right)^{-1} \text{ and}$$

$$\mathbf{w} = \left[ \mathbf{z}_1^\top \gamma + \frac{\mathcal{E}_1^2}{\sigma^2 \exp(\mathbf{z}_1^\top \gamma)} - 1 \dots \mathbf{z}_n^\top \gamma + \frac{\mathcal{E}_n^2}{\sigma^2 \exp(\mathbf{z}_n^\top \gamma)} - 1 \right]^\top.$$

Then,  $\gamma^*$  is accepted if a draw from a standard uniform distribution is less than the Metropolis-Hastings ratio (Gelman et al., 2004, p.291),

$$r = \underbrace{\frac{\mathbb{P}(\gamma^* | \gamma)}{\mathbb{P}(\gamma | \gamma^*)}}_{\text{jump ratio}} \underbrace{\frac{\mathbb{P}(\gamma^* | \sigma^2, \mathcal{E}_1, \dots, \mathcal{E}_n)}{\mathbb{P}(\gamma | \sigma^2, \mathcal{E}_1, \dots, \mathcal{E}_n)}}_{\text{likelihood ratio}}. \quad (\text{A.5})$$

The *jump ratio* in Equation A.5 becomes

$$\begin{aligned} \frac{\mathbb{P}(\gamma^* | \gamma)}{\mathbb{P}(\gamma | \gamma^*)} &= \frac{(2\pi)^{-\frac{p}{2}} |\mathbf{B}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} (\gamma^* - a(\gamma))^\top \mathbf{B}^{-1} (\gamma^* - a(\gamma))\right)}{(2\pi)^{-\frac{p}{2}} |\mathbf{B}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} (\gamma - a(\gamma^*))^\top \mathbf{B}^{-1} (\gamma - a(\gamma^*))\right)} \\ &= \exp\left(\frac{1}{2} (\gamma - \gamma^* + a(\gamma) - a(\gamma^*))^\top \mathbf{B}^{-1} (\gamma - \gamma^* + a(\gamma) - a(\gamma^*))\right), \end{aligned}$$

the *likelihood ratio* in Equation A.5 is calculated to be

$$\begin{aligned} \frac{\mathbb{P}(\gamma^* | \sigma^2, \mathcal{E}_1, \dots, \mathcal{E}_n)}{\mathbb{P}(\gamma | \sigma^2, \mathcal{E}_1, \dots, \mathcal{E}_n)} &= \frac{\prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2 \exp(\mathbf{z}_i^\top \gamma^*)}} \exp\left(-\frac{1}{2\sigma^2 \exp(\mathbf{z}_i^\top \gamma^*)} \mathcal{E}_i^2\right)}{\prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2 \exp(\mathbf{z}_i^\top \gamma)}} \exp\left(-\frac{1}{2\sigma^2 \exp(\mathbf{z}_i^\top \gamma)} \mathcal{E}_i^2\right)} \\ &\quad \times \frac{\prod_{j=1}^p \frac{1}{\sqrt{2\pi\Sigma_{jj}}} \exp\left(-\frac{1}{2\Sigma_{jj}} (\gamma_j^* - \gamma_{0j})^2\right)}{\prod_{j=1}^p \frac{1}{\sqrt{2\pi\Sigma_{jj}}} \exp\left(-\frac{1}{2\Sigma_{jj}} (\gamma_j - \gamma_{0j})^2\right)} \\ &= \exp\left(\frac{1}{2} \left( \sum_{i=1}^n \mathbf{z}_i^\top (\gamma - \gamma^*) \right. \right. \\ &\quad \left. \left. + \frac{1}{\sigma^2} \left( \sum_{i=1}^n \mathcal{E}_i^2 \left( \exp(-\mathbf{z}_i^\top \gamma) - \exp(-\mathbf{z}_i^\top \gamma^*) \right) \right) \right) \right) \end{aligned}$$

$$+ \sum_{j=1}^p \frac{1}{\Sigma_{jj}} \left( \gamma_j^2 - \gamma_j^{*2} + 2\gamma_{0j}(\gamma_j^* - \gamma_j) \right) \Bigg)$$

which results in the Metropolis-Hastings ratio of Equation A.5 of

$$\begin{aligned} r = & \exp \left( \frac{1}{2} \left( (\gamma - \gamma^* + a(\gamma) - a(\gamma^*))^\top \mathbf{B}^{-1} (\gamma + \gamma^* - (a(\gamma) + a(\gamma^*))) \right. \right. \\ & + \sum_{i=1}^n \mathbf{z}_i^\top (\gamma - \gamma^*) + \frac{1}{\sigma^2} \sum_{i=1}^n \mathcal{E}_i^2 \left( \exp(-\mathbf{z}_i^\top \gamma) - \exp(-\mathbf{z}_i^\top \gamma^*) \right) \\ & \left. \left. + \sum_{j=1}^p \frac{1}{\Sigma_{jj}} (\gamma_j^2 - \gamma_j^{*2} + 2\gamma_{0j}(\gamma_j^* - \gamma_j)) \right) \right). \end{aligned}$$

We now choose default values for the hyperparameters  $\gamma_{0,1}, \dots, \gamma_{0,p}$  and  $\Sigma_{11}, \dots, \Sigma_{pp}$ . All  $\gamma_{0,j}$  are set to 0. This choice centers the prior distribution of the linear model coefficients at zero. For the variance hyperparameters, we choose the  $\Sigma_{jj}$ 's to be 1000 for all  $j$  which is sufficiently large so our model will not impose shrinkage of the coefficients towards 0. Investigating this parameter's role in our algorithm we view as fruitful future work.

### Drawing the $\mathcal{D}$ 's (steps 2, 4, $\dots$ , $2m$ )

Sampling the leaf parameters must be adjusted to reflect the heteroskedasticity in the model. Observations considered highly variable will now *be downweighted* when constructing an estimate of a terminal node's prediction. Recall that  $\mathcal{D}_t \mid \mathfrak{F}_1, \mathbf{R}_1, \sigma_1^2, \dots, \sigma_n^2$  is the sampling for all leaves where each leaf is considered independent,

$$\begin{aligned} \mu_{t1} & \mid \mathfrak{F}_t, \mathbf{R}_{t_1}, \sigma_1^2, \dots, \sigma_n^2 \\ \mu_{t2} & \mid \mathfrak{F}_t, \mathbf{R}_{t_2}, \sigma_1^2, \dots, \sigma_n^2 \\ & \vdots \\ \mu_{tb_t} & \mid \mathfrak{F}_t, \mathbf{R}_{t_{b_t}}, \sigma_1^2, \dots, \sigma_n^2 \end{aligned}$$

where  $b_t$  denotes the number of terminal nodes in the  $t$ th tree and the subscripts on the  $R_{t_i}$  terms denote only the data that is apportioned to the specific terminal node. Recall that the prior on the leaf parameters,  $\mu$ 's, are normal and the likelihood of the responses,  $R_t$ 's, are assumed normal as well.

Given the normal-normal conjugacy, we derive the normal posterior distribution for a given leaf's prediction which is necessarily a function of the heterogeneous variances. We drop the subscripts on the  $R$  term for convenience and denote  $k$  as the number of data records that fell into the given leaf. Note that we drop double subscripting on the variances  $\{\sigma_1^2, \dots, \sigma_k^2\}$  which are a subset of  $\{\sigma_1^2, \dots, \sigma_n^2\}$  for the data records in this leaf.

$$\begin{aligned}
\mathbb{P}(\mu \mid \mathbf{R}, \sigma_1^2, \dots, \sigma_k^2, \gamma) &\propto \mathbb{P}(R \mid \mu, \sigma_1^2, \dots, \sigma_k^2, \gamma) \mathbb{P}(\mu \mid \sigma_1^2, \dots, \sigma_k^2) \\
&= \mathcal{N}_k(\mu \mathbf{1}, \sigma^2 \mathbf{D}) \mathcal{N}(0, \sigma_\mu^2) = \left( \prod_{i=1}^k \mathcal{N}(\mu, \sigma_i^2) \right) \mathcal{N}(0, \sigma_\mu^2) \\
&= \mathcal{N} \left( \frac{\sum_{i=1}^k \frac{R_i}{\sigma_i^2}}{\frac{1}{\sigma_\mu^2} + \sum_{i=1}^k \frac{1}{\sigma_i^2}}, \frac{1}{\frac{1}{\sigma_\mu^2} + \sum_{i=1}^k \frac{1}{\sigma_i^2}} \right)
\end{aligned}$$

### Drawing the $\mathfrak{F}$ 's (steps 1, 3, $\dots$ , $2m - 1$ )

As described in Kapelner and Bleich (2013, Appendix A), the draw of a new tree structure relies on a Metropolis-Hastings step where trees can be altered via growing new daughter nodes from an existing terminal node (**GROW**), pruning two terminal nodes such that their parent becomes terminal (**PRUNE**), or changing the splitting rule in a node (**CHANGE**).

Below is the Metropolis-Hastings ratio where the parameter sampled is the tree and the data is the responses unexplained by other trees denoted by  $\mathbf{R}$ . We denote

the new, proposal tree with an asterisk and the original tree without the asterisk.

$$r = \frac{\mathbb{P}\left(\mathfrak{T}_* \rightarrow \mathfrak{T}\right) \mathbb{P}\left(\mathfrak{T}_* \mid \mathbf{R}, \sigma^2, \gamma\right)}{\mathbb{P}\left(\mathfrak{T} \rightarrow \mathfrak{T}_*\right) \mathbb{P}\left(\mathfrak{T} \mid \mathbf{R}, \sigma^2, \gamma\right)} \quad (\text{A.6})$$

This can be reformulated using repeated applications of Bayes Rule to be a product of three ratios.

$$r = \underbrace{\frac{\mathbb{P}\left(\mathfrak{T}_* \rightarrow \mathfrak{T}\right)}{\mathbb{P}\left(\mathfrak{T} \rightarrow \mathfrak{T}_*\right)}}_{\text{transition ratio}} \times \underbrace{\frac{\mathbb{P}\left(\mathbf{R} \mid \mathfrak{T}_*, \sigma^2, \gamma\right)}{\mathbb{P}\left(\mathbf{R} \mid \mathfrak{T}, \sigma^2, \gamma\right)}}_{\text{likelihood ratio}} \times \underbrace{\frac{\mathbb{P}\left(\mathfrak{T}_*\right)}{\mathbb{P}\left(\mathfrak{T}\right)}}_{\text{tree structure ratio}}$$

Note that the probability of the tree structure is independent of  $\sigma^2$  and  $\gamma$ .

The transition ratio and the tree structure ratio remain the same as in the original BART as they do not depend on the variance parameters. The likelihood ratio now must take into account the heterogeneity in variances. The PRUNE likelihood ratio is the inverse of the GROW likelihood ratio. Thus, we only need to compute likelihood ratios for GROW and CHANGE.

### The Likelihood Ratio for the GROW proposal

Since the likelihoods are solely determined by the terminal nodes, the grown proposal tree differs from the original tree by only the selected node to be grown. We denote the node to be grown by  $\ell$ , the left child by  $\ell_L$  and the right child by  $\ell_R$ .

$$\frac{\mathbb{P}\left(\mathbf{R} \mid \mathfrak{T}_*, \sigma^2, \gamma\right)}{\mathbb{P}\left(\mathbf{R} \mid \mathfrak{T}, \sigma^2, \gamma\right)} = \frac{\int_{\mathbb{R}} \mathbb{P}\left(\mathbf{R}_{\ell_L} \mid \mu, \sigma^2, \gamma\right) \mathbb{P}(\mu) \, \mathrm{d}\mu \int_{\mathbb{R}} \mathbb{P}\left(\mathbf{R}_{\ell_R} \mid \mu, \sigma^2, \gamma\right) \mathbb{P}(\mu) \, \mathrm{d}\mu}{\int_{\mathbb{R}} \mathbb{P}\left(\mathbf{R}_{\ell} \mid \mu, \sigma^2, \gamma\right) \mathbb{P}(\mu) \, \mathrm{d}\mu} \quad (\text{A.7})$$

Each of these three integrals are the same with regards to marginalizing the  $\mu$



term:

$$\begin{aligned}
& \int_{\mathbb{R}} \mathbb{P}(\mathbf{R} \mid \mu, \sigma^2, \gamma) \mathbb{P}(\mu) \, d\mu \\
&= \int_{\mathbb{R}} \frac{1}{(2\pi)^{\frac{n}{2}} |\mathbf{D}|^{-\frac{1}{2}}} \exp\left(-\frac{1}{2} (\mathbf{R} - \mu \mathbf{1})^\top \mathbf{D}^{-1} (\mathbf{R} - \mu \mathbf{1})\right) \frac{1}{\sqrt{2\pi\sigma_\mu^2}} \exp\left(-\frac{1}{2\sigma_\mu^2} \mu^2\right) \, d\mu \\
&= \frac{1}{(2\pi)^{\frac{n}{2}+1} \sqrt{\sigma_\mu^2 \prod_{i=1}^n \sigma_i^2}} \exp\left(-\frac{1}{2} \sum_{i=1}^n \frac{R_i^2}{\sigma_i^2}\right) \\
&\quad \times \int_{\mathbb{R}} \exp\left(-\frac{1}{2} \left(\frac{\mu^2}{\sigma_\mu^2} - 2\mu \sum_{i=1}^n \frac{R_i}{\sigma_i^2} + \mu^2 \sum_{i=1}^n \frac{1}{\sigma_i^2}\right)\right) \, d\mu \\
&= \left( (2\pi)^n \left(1 + \sigma_\mu^2 \sum_{i=1}^n \frac{1}{\sigma_i^2}\right) \prod_{i=1}^n \sigma_i^2 \right)^{-\frac{1}{2}} \exp\left(\frac{1}{2} \left( \frac{\sigma_\mu^2 \left(\sum_{i=1}^n \frac{R_i}{\sigma_i^2}\right)^2}{1 + \sigma_\mu^2 \sum_{i=1}^n \frac{1}{\sigma_i^2}} - \sum_{i=1}^n \frac{R_i^2}{\sigma_i^2} \right)\right) \quad (\text{A.8})
\end{aligned}$$

The likelihood ratio can now be computed by substituting Equation A.8 into Equation A.7 three times to arrive at:

$$\begin{aligned}
\frac{\mathbb{P}(\mathbf{R} \mid \mathbf{\hat{\gamma}}_* \sigma^2, \gamma)}{\mathbb{P}(\mathbf{R} \mid \mathbf{\hat{\gamma}}, \sigma^2, \gamma)} &= \sqrt{\frac{1 + \sigma_\mu^2 \sum_{i=1}^{n_\ell} \frac{1}{\sigma_i^2}}{\left(1 + \sigma_\mu^2 \sum_{i=1}^{n_{\ell_L}} \frac{1}{\sigma_i^2}\right) \left(1 + \sigma_\mu^2 \sum_{i=1}^{n_{\ell_R}} \frac{1}{\sigma_i^2}\right)}} \\
&\quad \times \exp\left(\frac{\sigma_\mu^2}{2} \left( \frac{\left(\sum_{i=1}^{n_{\ell_L}} \frac{R_i}{\sigma_i^2}\right)^2}{1 + \sigma_\mu^2 \sum_{i=1}^{n_{\ell_L}} \frac{1}{\sigma_i^2}} + \frac{\left(\sum_{i=1}^{n_{\ell_R}} \frac{R_i}{\sigma_i^2}\right)^2}{1 + \sigma_\mu^2 \sum_{i=1}^{n_{\ell_R}} \frac{1}{\sigma_i^2}} - \frac{\left(\sum_{i=1}^{n_\ell} \frac{R_i}{\sigma_i^2}\right)^2}{1 + \sigma_\mu^2 \sum_{i=1}^{n_\ell} \frac{1}{\sigma_i^2}} \right)\right).
\end{aligned}$$

### The Likelihood Ratio for the CHANGE proposal

The homoskedastic BART implementation of Kapelner and Bleich (2013) considered change proposals for singly internal nodes only (i.e., both daughter nodes must be

terminal nodes). The likelihood ratio in this case simplifies to the likelihood of these two leaves before and after the change:

$$\frac{\mathbb{P}(R_{\ell^*,1}, \dots, R_{\ell^*,n_{\ell^*}} \mid \sigma^2, \gamma) \mathbb{P}(R_{r^*,1}, \dots, R_{r^*,n_{r^*}} \mid \sigma^2, \gamma)}{\mathbb{P}(R_{\ell,1}, \dots, R_{\ell,n_{\ell}} \mid \sigma^2, \gamma) \mathbb{P}(R_{r,1}, \dots, R_{r,n_r} \mid \sigma^2, \gamma)}. \quad (\text{A.9})$$

The  $\ell$  refers to the left terminal node and  $r$  refers to the terminal right node. The  $\ell^*$  and the  $r^*$  denote these same two nodes in the proposal tree, i.e after the parent's split rule was changed.

The likelihood with  $\mu$  margined out has been calculated in Equation A.8 and we express it here with a convenient factorization:

$$\underbrace{\left( (2\pi)^{n_{\ell}} \left( 1 + \sigma_{\mu}^2 \sum_{i=1}^{n_{\ell}} \frac{1}{\sigma_i^2} \right) \prod_{i=1}^{n_{\ell}} \sigma_i^2 \right)^{-\frac{1}{2}}}_{A} \underbrace{\exp \left( -\frac{1}{2} \sum_{i=1}^{n_{\ell}} \frac{R_i^2}{\sigma_i^2} \right)}_B \underbrace{\exp \left( \frac{\sigma_{\mu}^2}{2} \left( \frac{\left( \sum_{i=1}^{n_{\ell}} \frac{R_i}{\sigma_i^2} \right)^2}{1 + \sigma_{\mu}^2 \sum_{i=1}^{n_{\ell}} \frac{1}{\sigma_i^2}} \right) \right)}_C.$$

To find the ratio, we must substitute this expression into equation A.9 four times.

We begin by substituting only the term marked  $A$  above to arrive at

$$\frac{(2\pi)^{\frac{n_{\ell}}{2}} \sqrt{\left( \prod_{i=1}^{n_{\ell}} \sigma_i^2 \right) \left( 1 + \sigma_{\mu}^2 \sum_{i=1}^{n_{\ell}} \frac{1}{\sigma_i^2} \right)} (2\pi)^{\frac{n_r}{2}} \sqrt{\left( \prod_{i=1}^{n_r} \sigma_i^2 \right) \left( 1 + \sigma_{\mu}^2 \sum_{i=1}^{n_r} \frac{1}{\sigma_i^2} \right)}}{(2\pi)^{\frac{n_{\ell^*}}{2}} \sqrt{\left( \prod_{i=1}^{n_{\ell^*}} \sigma_i^2 \right) \left( 1 + \sigma_{\mu}^2 \sum_{i=1}^{n_{\ell^*}} \frac{1}{\sigma_i^2} \right)} (2\pi)^{\frac{n_{r^*}}{2}} \sqrt{\left( \prod_{i=1}^{n_{r^*}} \sigma_i^2 \right) \left( 1 + \sigma_{\mu}^2 \sum_{i=1}^{n_{r^*}} \frac{1}{\sigma_i^2} \right)}}.$$

Of course  $n_{\ell^*} + n_{r^*} = n_{\ell} + n_r = n$  and the  $\sigma^2$ 's are the same since they aren't drawn until later on in the Gibbs sampling scheme. Thus, the above reduces to

$$\sqrt{\frac{\left(1 + \sigma_\mu^2 \sum_{i=1}^{n_\ell} \frac{1}{\sigma_i^2}\right) \left(1 + \sigma_\mu^2 \sum_{i=1}^{n_r} \frac{1}{\sigma_i^2}\right)}{\left(1 + \sigma_\mu^2 \sum_{i=1}^{n_{\ell^*}} \frac{1}{\sigma_i^2}\right) \left(1 + \sigma_\mu^2 \sum_{i=1}^{n_{r^*}} \frac{1}{\sigma_i^2}\right)}}. \quad (\text{A.10})$$

In term  $B$ , upon making all four substitutions, it is clear all the parents' observations must be summed in the numerator as well as the denominator. Thus, this term cancels.

Now we examine term  $C$ . Due to the exponentiation, multiplication becomes addition and division becomes subtraction and all four substitutions yield

$$\exp \left( \frac{\sigma_\mu^2}{2} \left( \frac{\left( \sum_{i=1}^{n_{\ell^*}} \frac{R_i}{\sigma_i^2} \right)^2}{1 + \sigma_\mu^2 \sum_{i=1}^{n_{\ell^*}} \frac{1}{\sigma_i^2}} + \frac{\left( \sum_{i=1}^{n_{r^*}} \frac{R_i}{\sigma_i^2} \right)^2}{1 + \sigma_\mu^2 \sum_{i=1}^{n_{r^*}} \frac{1}{\sigma_i^2}} - \frac{\left( \sum_{i=1}^{n_\ell} \frac{R_i}{\sigma_i^2} \right)^2}{1 + \sigma_\mu^2 \sum_{i=1}^{n_\ell} \frac{1}{\sigma_i^2}} - \frac{\left( \sum_{i=1}^{n_r} \frac{R_i}{\sigma_i^2} \right)^2}{1 + \sigma_\mu^2 \sum_{i=1}^{n_r} \frac{1}{\sigma_i^2}} \right) \right). \quad (\text{A.11})$$

Multiplying terms A.10 and A.11 yields the likelihood ratio for the **CHANGE** proposal.

## A.5 Supplement for Chapter 11

---

**Algorithm 8** ICE algorithm: Given  $\mathbf{X}$ , the  $N \times p$  feature matrix,  $\hat{f}$ , the fitted model,  $S \subset \{1, \dots, p\}$ , the subset of predictors for which to compute partial dependence, return  $\hat{f}_S^{(1)}, \dots, \hat{f}_S^{(N)}$ , the estimated partial dependence curves for constant values of  $\mathbf{x}_C$ .

---

```

1: function ICE( $\mathbf{X}$ ,  $\hat{f}$ ,  $S$ )
2:   for  $i \leftarrow 1 \dots N$  do
3:      $\hat{f}_S^{(i)} \leftarrow \mathbf{0}_{N \times 1}$ 
4:      $\mathbf{x}_C \leftarrow \mathbf{X}[i, C]$  ▷ fix  $\mathbf{x}_C$  at the  $i$ th observation's  $C$  columns
5:     for  $\ell \leftarrow 1 \dots N$  do
6:        $\mathbf{x}_S \leftarrow \mathbf{X}[\ell, S]$  ▷ vary  $\mathbf{x}_S$ 
7:        $\hat{f}_{S\ell}^{(i)} \leftarrow \hat{f}([\mathbf{x}_S, \mathbf{x}_C])$  ▷ the  $i$ th curve's  $\ell$ th coordinate
8:     end for
9:   end for
10:  return  $[\hat{f}_S^{(1)}, \dots, \hat{f}_S^{(N)}]$ 
11: end function

```

---

---

**Algorithm 9** d-ICE algorithm: Given  $\mathbf{X}$ , the  $N \times p$  feature matrix;  $\hat{f}_S^{(1)}, \dots, \hat{f}_S^{(N)}$ , the estimated partial dependence functions for subset  $S$  in the ICE plot;  $D$ , a function that computes the numerical derivative; returns  $d\hat{f}_S^{(1)}, \dots, d\hat{f}_S^{(N)}$ , the derivatives of the estimated partial dependence. In our implementation  $D$  first smooths the ICE plot using the “super-smoother” and subsequently estimates the derivative from the smoothed ICE plot.

---

```

1: function D-ICE( $\mathbf{X}, \hat{f}_S^{(1)}, \dots, \hat{f}_S^{(N)}, D$ )
2:   for  $i \leftarrow 1 \dots N$  do
3:      $df_S^{(i)} \leftarrow \mathbf{0}_{N \times 1}$ 
4:      $\mathbf{x}_C \leftarrow \mathbf{X}[i, C]$   $\triangleright$  row of the  $i$ th observation, columns corresponding to  $C$ 
5:     for  $\ell \leftarrow 1 \dots N$  do
6:        $\mathbf{x}_S \leftarrow \mathbf{X}[\ell, S]$ 
7:        $df_{S\ell}^{(i)} \leftarrow D \left[ \hat{f}^{(i)}(\mathbf{x}_S, \mathbf{x}_C) \right]$   $\triangleright$  numerical partial derivative at  $\hat{f}^{(i)}(\mathbf{x}_S, \mathbf{x}_C)$  w.r.t.
          $\mathbf{x}_S$ 
8:     end for
9:   end for
10:  return  $[df_S^{(1)}, \dots, df_S^{(N)}]$ 
11: end function

```

---

## A.6 Supplement for Chapter 12

### A.6.1 Demographics Questions

Questions and answer choices will be displayed in random order to reduce possible satisficing bias. The fade-in *a la* Kaplaner and Chandler (2010) will not be used here to reduce burden to the subject. Questions marked “NFC” are a subset of the “need for cognition” survey, a scale which measures to what degree subjects desire to think abstractly and expend mental effort (see Cacioppo et al., 1984). Questions marked “BIG5” measure attributes on the well-accepted NEO five-factor model of

personality traits (see McCrae and Costa, 1987); we employ a highly modified survey that includes only one question per factor.

- 1) Age (natural number  $\geq 18$ )
- 2) Gender (binary: 1 for male)
- 3) Urban or Rural location (binary: 1 for urban)
- 4) Highest Education Attainment (categorical: high school, some college, bachelors, masters, doctorate)
- 5) Marital Status (categorical: married, divorced, single, widowed, in a relationship)
- 6) Employment Status (categorical: full-time, part-time, self-employed, unemployed)
- 7) Number of children (natural number)
- 8) Annual Personal Income (approximated to the nearest dollar)
- 9) Religion by Birth (categorical: Protestant Christian, Roman Catholic, Evangelical Christian, Jewish, Muslim, Hindu, Buddhist, Other)
- 10) Belief in a divine being (binary: 1 for belief)
- 11) Race (categorical: White, Hispanic, Black, Asian, Native American, Other)
- 12) Would you want to participate in an academic study? (binary: 1 for participate)
- 13) NFC: I prefer my life to be filled with puzzles I must solve (binary: 1 for prefer)
- 14) NFC: It's enough for me that something gets the job done; I don't care how or why it works (binary: prefer)
- 15) BIG5: I am more (i) inventive and curious (ii) consistent and cautious (binary: 1 for i)

- 16) BIG5: I am more (i) efficient and organized (ii) easy-going and more careless  
(binary: 1 for i)
- 17) BIG5: I am more (i) outgoing and energetic (ii) solitary and reserved (binary: 1  
for i)
- 18) BIG5: I am more (i) friendly and compassionate (ii) cold and unkind (binary: 1  
for i)
- 19) BIG5: I am more (i) sensitive and nervous (ii) secure and confident (binary: 1 for  
i)

### **A.6.2 Experimental Covariates**

In addition to the demographic questions, we will also collected as much information as we could during the experiment about subject behavior:

- 1) Time (in seconds) it takes the subject to accept the HIT and begin the task.
- 2) Time (in seconds) it takes for the subject to fill in the demographic questions.
- 3) Time (in seconds) it takes the subject to do the experimental question.
- 4) Number of page reloads on the experimental question.
- 5) Number of characters in the free-response comments box after the experiment is completed.

### **A.6.3 Full Text of the Studies**

Below is the full text of each of the studies. Text displayed for subjects in the *treatment* group is colored red and text displayed for subjects in the *control* group is colored green.

(a) Framing Study

You are lying on the beach on a hot day. All you have to drink is ice water. For the last hour you have been thinking about how much you would enjoy a nice cold bottle of your favorite brand of beer.

A companion gets up to go make a phone call and offers to bring back a beer from the only nearby place where beer is sold, **a fancy resort hotel / a small, run-down grocery store**

He says that the beer might be expensive and so asks how much you are willing to pay for the beer. He says that he will buy the beer if it costs as much or less than the price you state. But if it costs more than the price you state he will not buy it.

You trust your friend, and there is no possibility of bargaining with **the bartender. / store owner**. What price do you tell him?

(b) Priming Study

**As Jesus started on his way, a man ran up to him and fell on his knees before him. “Good teacher,” he asked, “what must I do to inherit eternal life?”**

**“Why do you call me good?” Jesus answered. “No one is good-except God alone. You know the commandments: You shall not murder, you shall not commit adultery, you shall not steal, you shall not give false testimony, you shall not defraud, honor your father and mother.”**

**“Teacher,” he declared, “all these I have kept since I was a boy.”**

**Jesus looked at him and loved him. “One thing you lack,” he said.**

**“Go, sell everything you have and give to the poor, and you will have treasure in heaven. Then come, follow me.”**

**At this the man’s face fell. He went away sad, because he had great**



wealth.

Jesus looked around and said to his disciples, “How hard it is for the rich to enter the kingdom of God!”

Some species of fish are viviparous. In such species the mother retains the eggs and nourishes the embryos. Typically, viviparous fish have a structure analogous to the placenta seen in mammals connecting the mother’s blood supply with that of the embryo.

Examples of viviparous fish include the surf-perches, spltfins, and lemon shark. Some viviparous fish exhibit oophagy, in which the developing embryos eat other eggs produced by the mother. This has been observed primarily among sharks, such as the shortfin mako and porbeagle, but is known for a few bony fish as well, such as the halfbeak *Nomorhamphus ebrardtii*.

Intrauterine cannibalism is an even more unusual mode of vivipary, in which the largest embryos eat weaker and smaller siblings. This behavior is also most commonly found among sharks, such as the grey nurse shark, but has also been reported for *Nomorhamphus ebrardtii*. Aquarists commonly refer to ovoviviparous and viviparous fish as live-bearers although they are known by other colloquial names as well.

You have just been randomly paired with another Turker. He or she is seeing the same thing you are right now and you can choose to cooperate with each other or not to cooperate. If both of you choose to cooperate, you will both be awarded a bonus of 10 cents.

However, if you choose to cooperate and the other person doesn’t, you make a bonus of 20 cents and the other person gets nothing.

If you both choose to not cooperate with each other at the same time, you both will get 4 cents as a bonus.

Would you like to cooperate with this other person?”

(c) Sunk Cost

**You paid handsomely for tickets** / **Your friend gave you tickets for free** for a theater production that received rave reviews. You’ve been wanting to see this show for quite some time and it only plays once in your city.

However, during the night of the show, the weather turned for the worst. There’s heavy rain and some hail. Your city’s mayor is not recommending anyone drive on the roads. Despite the inclement weather, the show is still going to be performed.

How likely are you to go? 0 means “you’re not going” and 9 means “you’re definitely going” while 5 would mean that “you’re unsure about going.”

## A.7 Supplement for Chapter 13

An additional strategy is to match after the experiment has completed. We choose to dynamically allocate using the stratification procedure found in Section 3. Once all  $n$  subjects are allocated, we then calculate all  $\binom{n}{2}$  Mahalanobis-squared distances according to Equation 2. Note that we have the luxury here of using all  $n$  subjects to calculate  $\mathbf{S}^{-1}$ . Then, using the  $F_{p,n-p}$  approximation, we convert the squared distances to nominal probabilities. We then use  $\lambda$  as a cutoff: if the probability distance between two subjects is greater than  $\lambda$ , we set this distance to  $\infty$ . We then use Hansen and Klopfer (2006)’s `optmatch` package in R to do an optimal matching based on this enhanced distance matrix. The optimal matches found become the

*matched* data and the data that was not optimally matched becomes the *unmatched* data. The unmatched data is the analogue of the reservoir during the *on-the-fly* matching as explained in Section 2.2. Using these two subsets of the data, we can now generate estimates according to Equations 4 and 6. Over many repeated samples, we can estimate power, just like among the other competitors.

Figures A.10 to A.17 show the power results of Section 3’s simulations for many different values of  $\lambda$ . We include the post-matching results in these figures under “SPM.”

Table A.2 displays results for bias including post-matching results.

Sample Average Absolute Bias							
$n$	Allocation	Scenario NL		Scenario LI		Scenario ZE	
	Method	Classic	Linear	Classic	Linear	Classic	Linear
50	CR	0.704	0.635	0.752	0.404	0.402	0.395
	Efron's BCD	0.685	0.654	0.730	0.395	0.391	0.412
	Stratification	0.612	0.556	0.509	0.395	0.380	0.385
	Minimization	0.618	0.601	0.519	0.392	0.387	0.382
	Seq. Matching	0.466	0.466	0.478	0.434	0.418	0.455
	Seq. Post Matching	0.552	0.552	0.504	0.419	0.400	0.427
100	CR	0.515	0.454	0.530	0.287	0.280	0.283
	Efron's BCD	0.502	0.447	0.534	0.279	0.285	0.282
	Stratification	0.427	0.409	0.362	0.270	0.276	0.277
	Minimization	0.441	0.424	0.340	0.275	0.278	0.277
	Seq. Matching	0.318	0.322	0.324	0.298	0.286	0.295
	Seq. Post Matching	0.388	0.420	0.350	0.295	0.272	0.291
200	CR	0.369	0.321	0.376	0.198	0.196	0.193
	Efron's BCD	0.340	0.321	0.379	0.201	0.197	0.195
	Stratification	0.304	0.284	0.255	0.197	0.196	0.198
	Minimization	0.315	0.299	0.240	0.194	0.193	0.196
	Seq. Matching	0.210	0.221	0.210	0.199	0.195	0.207
	Seq. Post Matching	0.290	0.313	0.242	0.200	0.198	0.194

Table A.2: Average absolute bias of sequential matching and post matching using stratification to allocate versus competitors by scenario and testing procedure. Both matching methods used  $\lambda = 0.10$  (other values yielded similar results) and the  $Z$  approximation of Equations 4 and 6. Exact tests are not shown because they do not admit an estimate, only a  $p$  value.

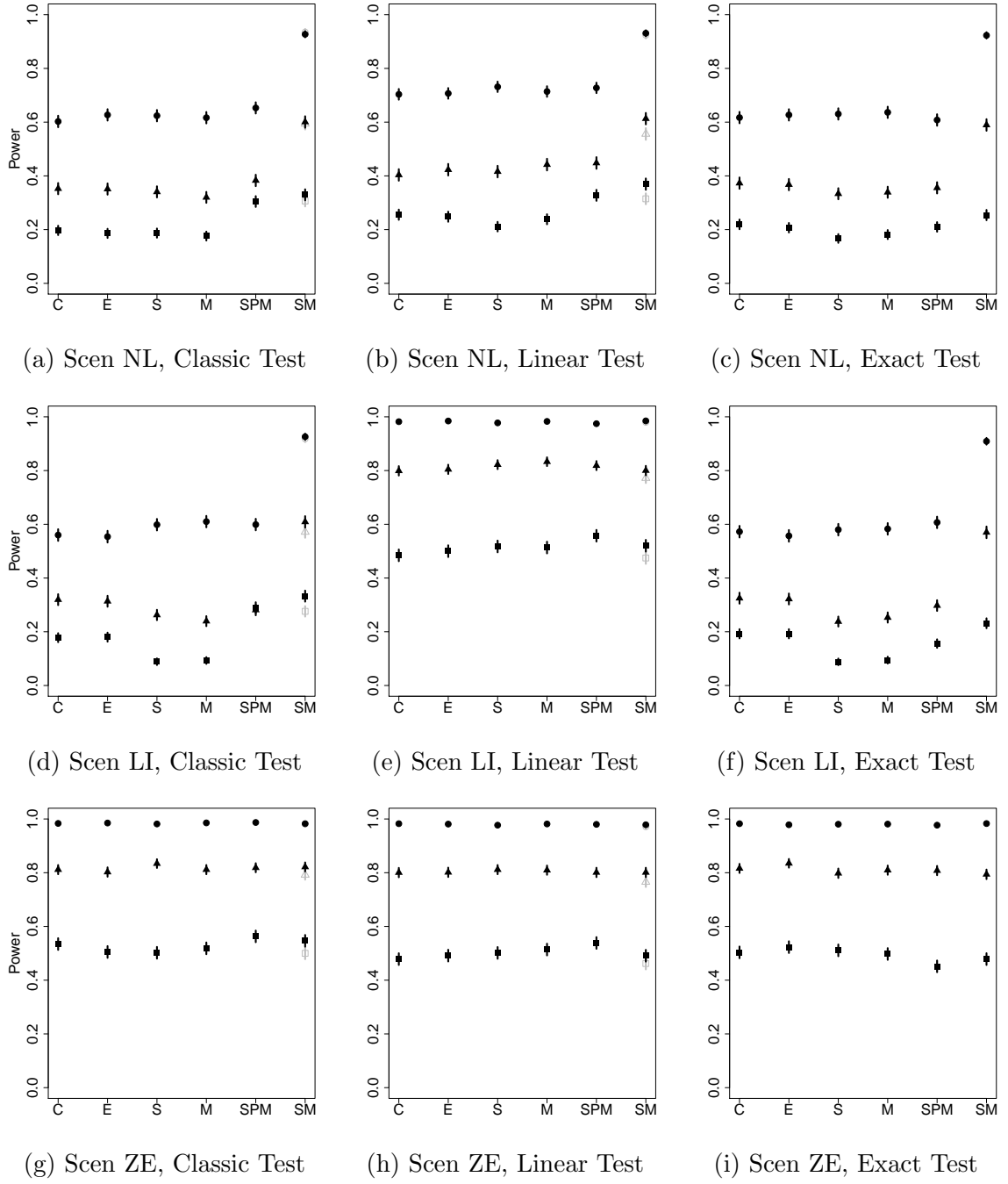


Figure A.10: Power illustrated for matching parameter  $\lambda = 1\%$ . We also include the post matching procedure of Appendix A.7 as “SPM.” See paper, Figure 1 for full caption information.

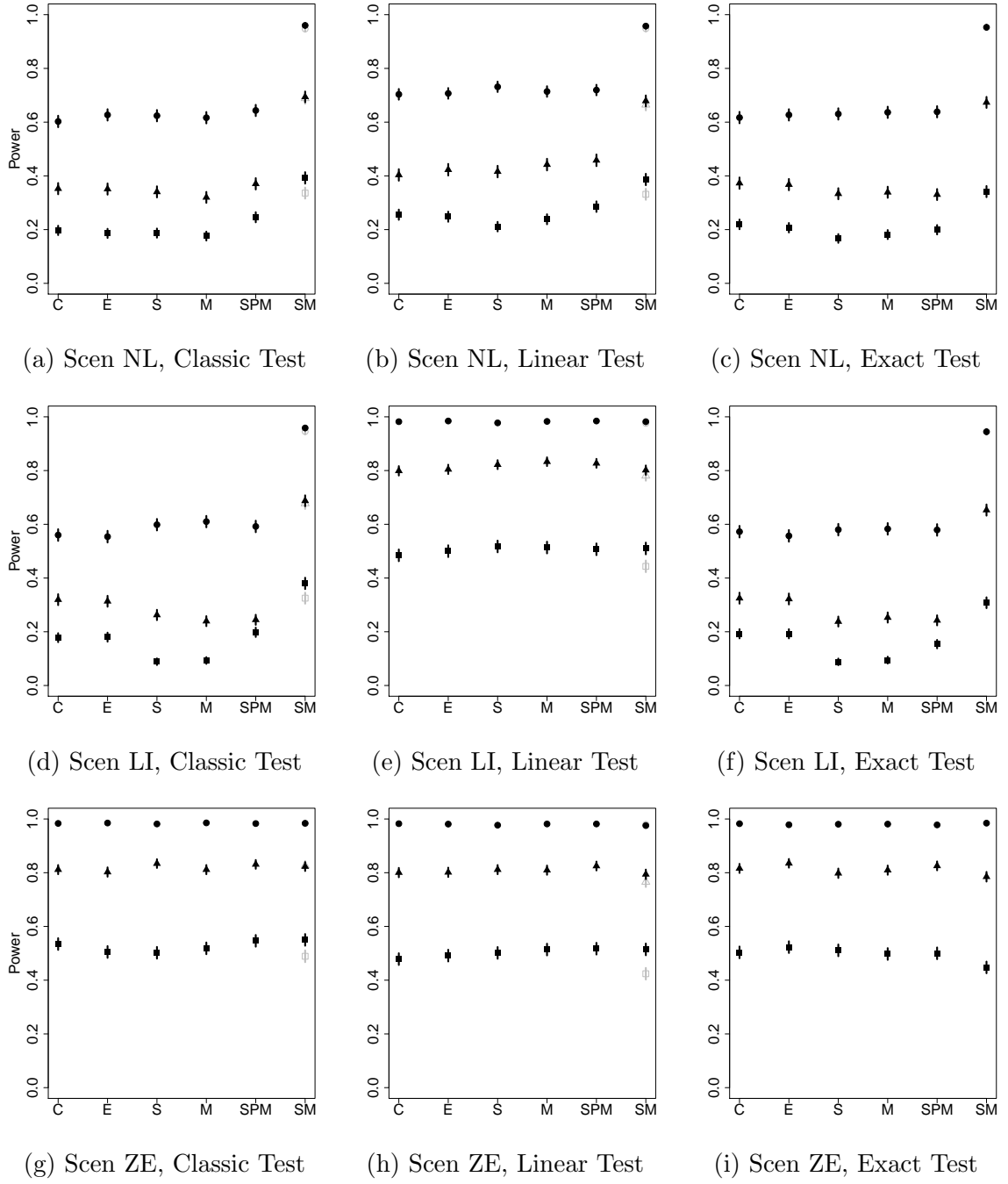


Figure A.11: Power illustrated for matching parameter  $\lambda = 2.5\%$ . We also include the post matching procedure of Appendix A.7 as “SPM.” See paper, Figure 1 for full caption information.

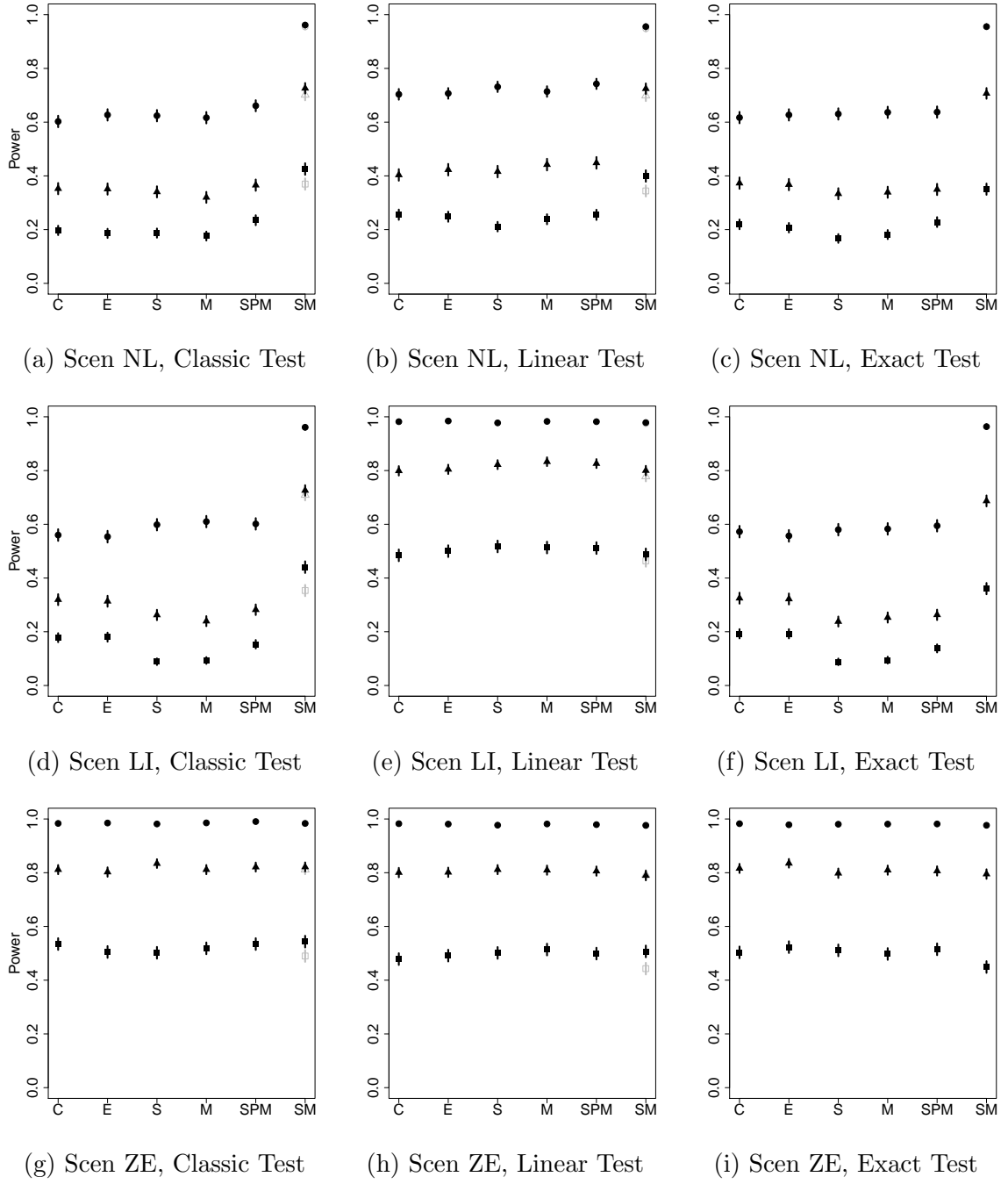


Figure A.12: Power illustrated for matching parameter  $\lambda = 5\%$ . We also include the post matching procedure of Appendix A.7 as “SPM.” See paper, Figure 1 for full caption information.

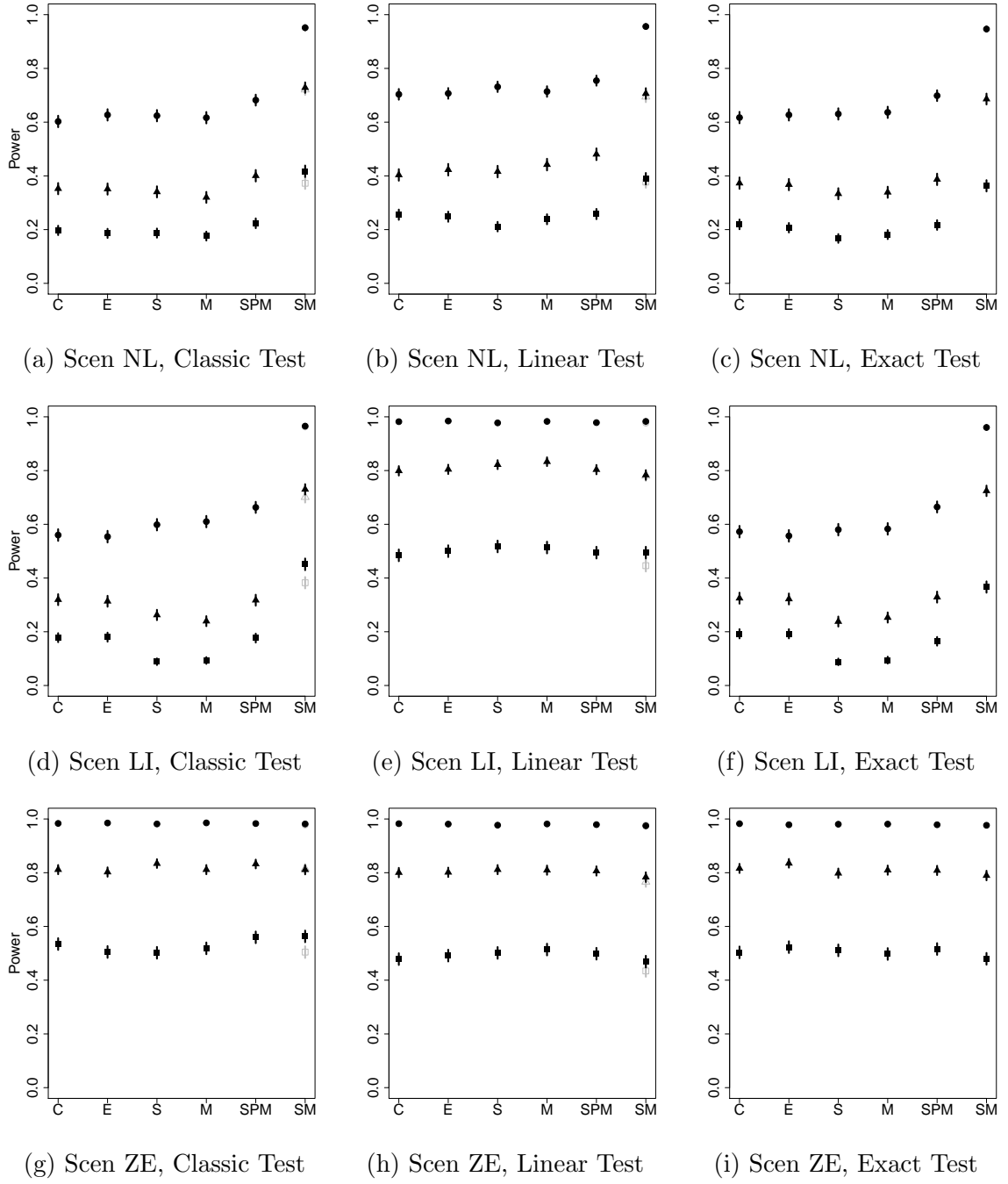


Figure A.13: Power illustrated for matching parameter  $\lambda = 7.5\%$ . We also include the post matching procedure of Appendix A.7 as “SPM.” See paper, Figure 1 for full caption information.



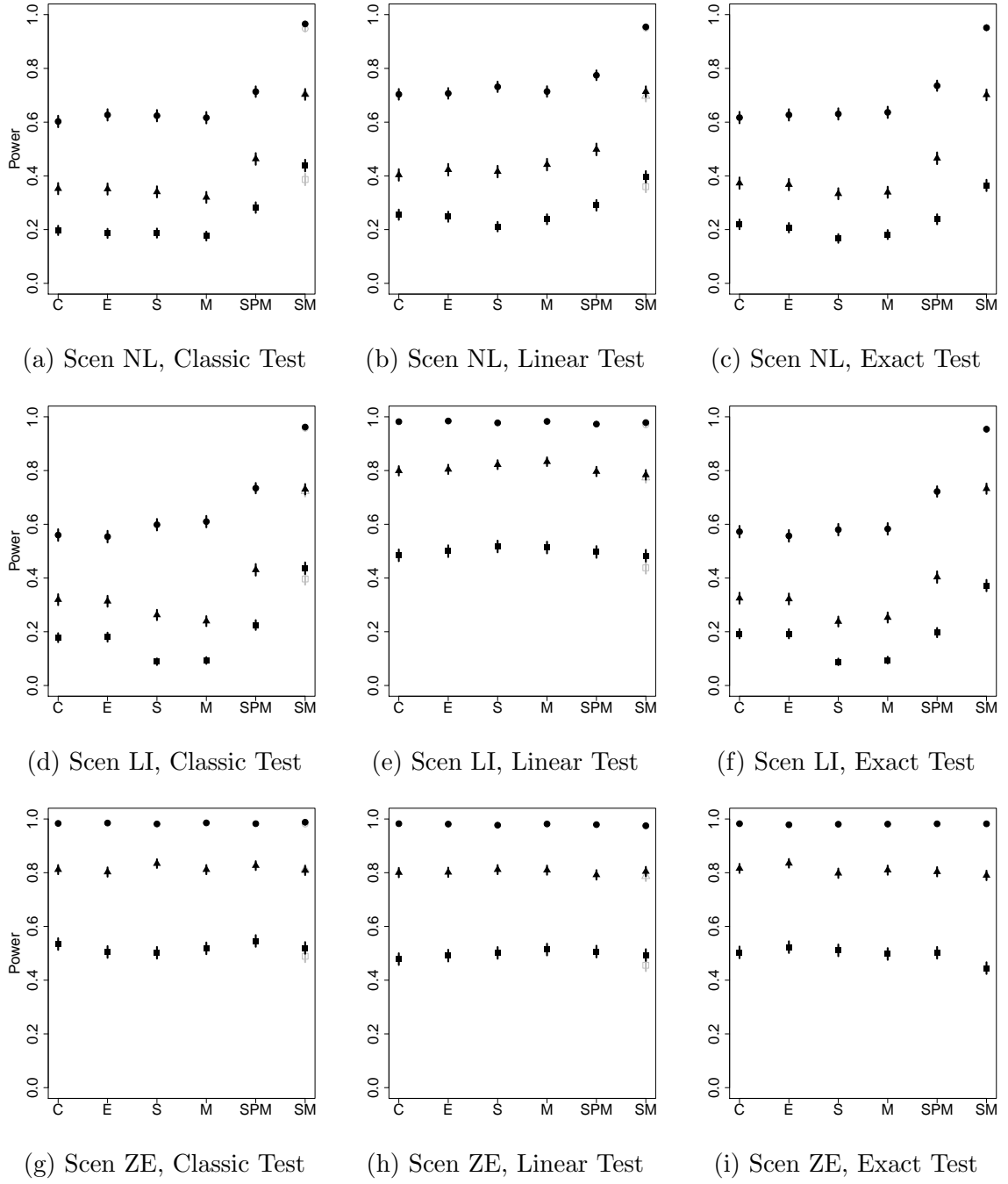


Figure A.14: Power illustrated for matching parameter  $\lambda = 10\%$ . We also include the post matching procedure of Appendix A.7 as “SPM.” See paper, Figure 1 for full caption information.

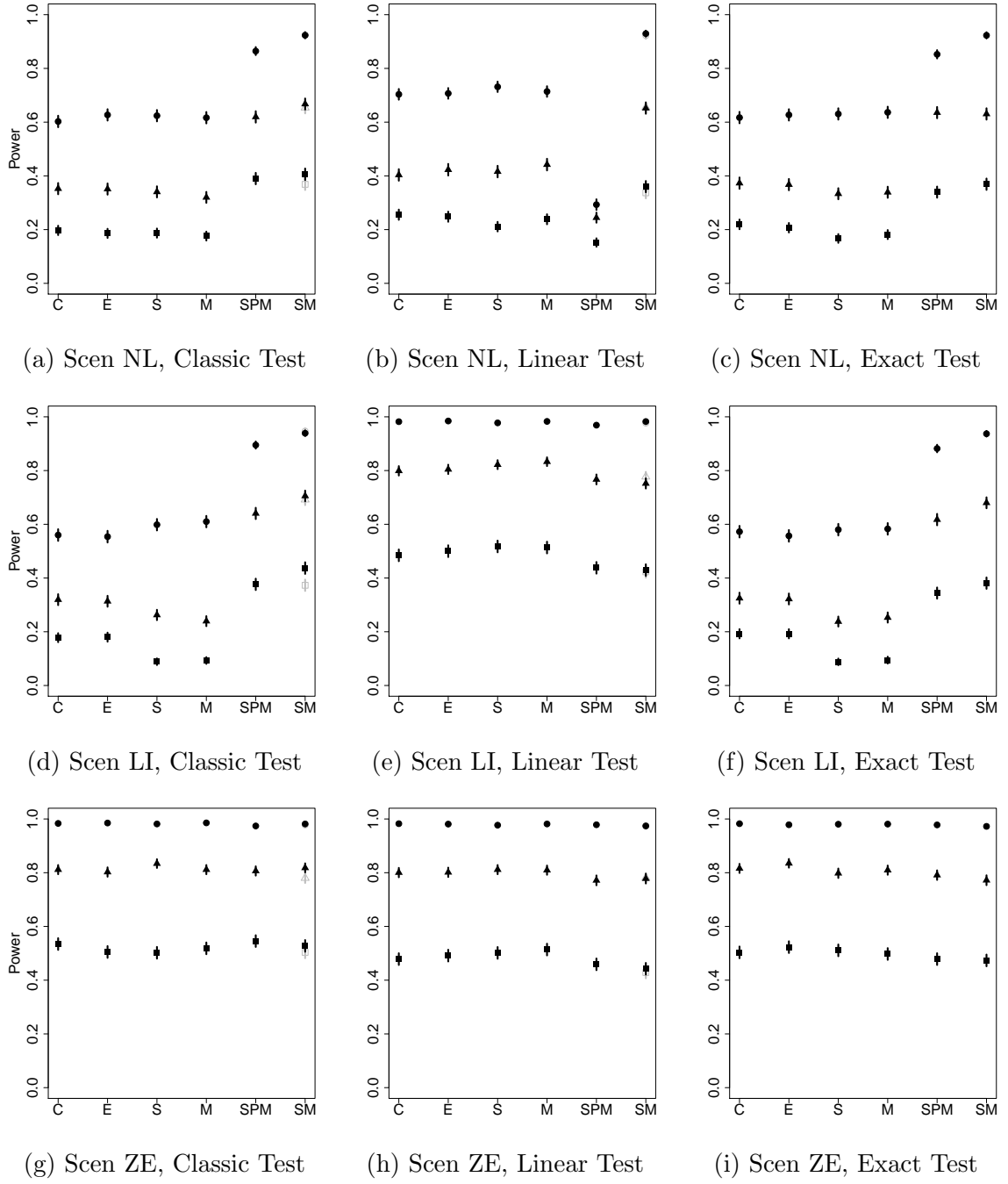


Figure A.15: Power illustrated for matching parameter  $\lambda = 20\%$ . We also include the post matching procedure of Appendix A.7 as “SPM.” See paper, Figure 1 for full caption information.

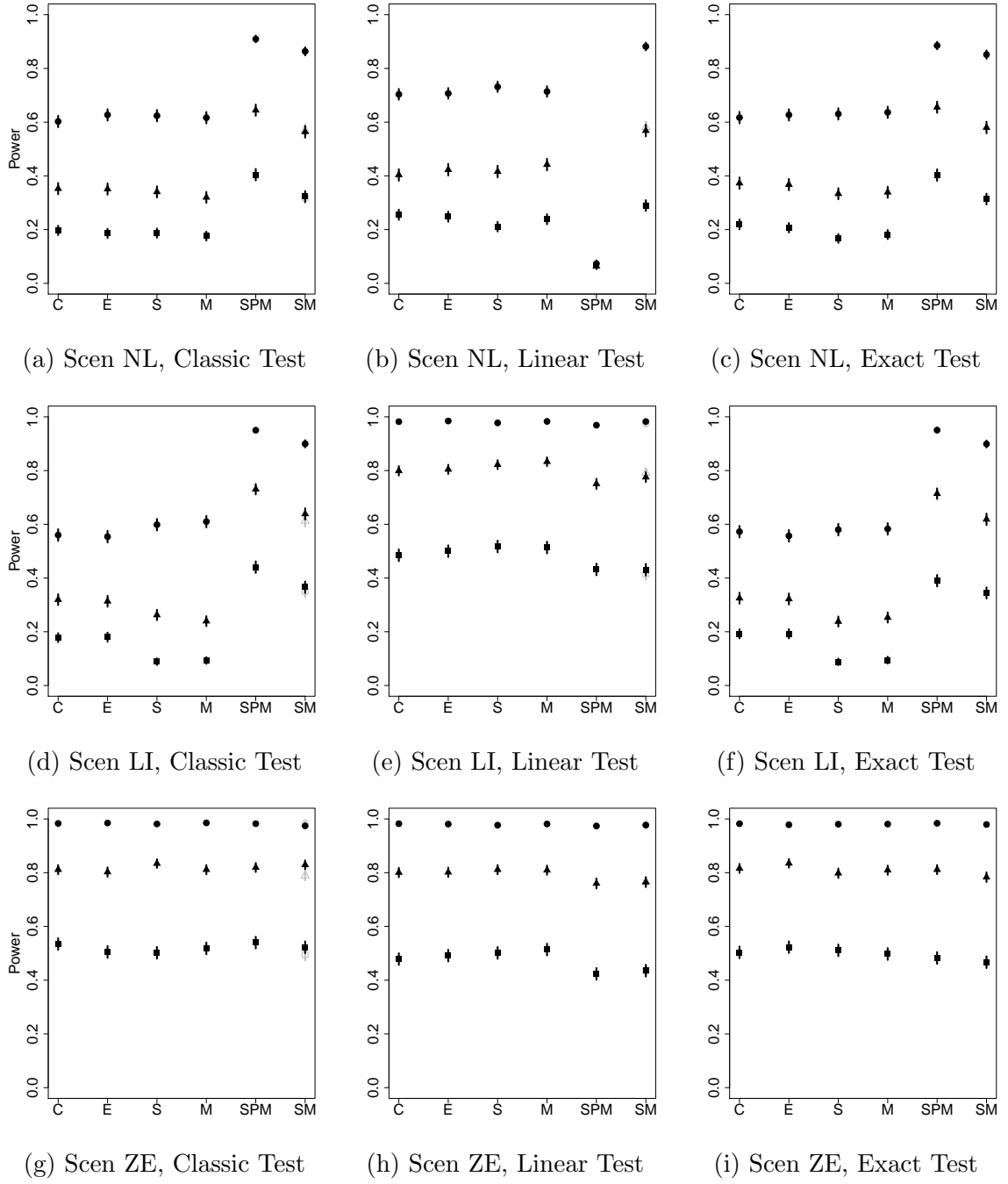


Figure A.16: Power illustrated for matching parameter  $\lambda = 35\%$ . We also include the post matching procedure of Appendix A.7 as “SPM.” See paper, Figure 1 for full caption information.

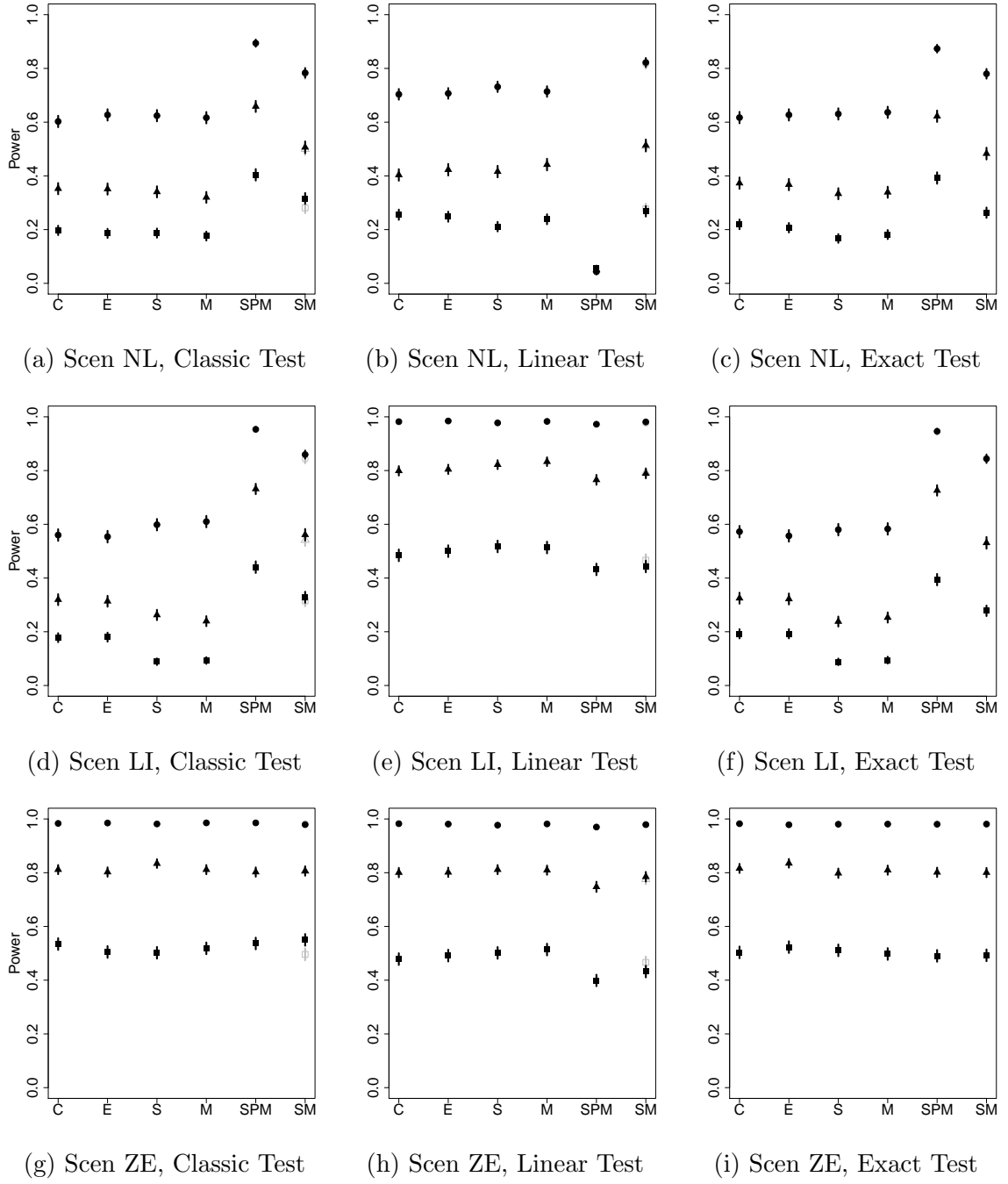


Figure A.17: Power illustrated for matching parameter  $\lambda = 50\%$ . We also include the post matching procedure of Appendix A.7 as “SPM.” See paper, Figure 1 for full caption information.

## A.8 BART for Panel Data

Sometimes data is collected in a panel, meaning the  $n$  observations are collected from  $w$  people or “workers” (the usual notation is  $m$  but this is being used for number of trees in the BART setup). Many times all observations for a given worker have things in common, *i.e.* they are correlated. The higher the correlation, the closer the effective sample size is to  $w$  and not  $n$ . The lower the correlation, the closer the sample size is to  $n$  and not  $w$ . The goal of this project is to create BART models that can account for these correlations.

### A.8.1 The Updated Model

Denote  $n_1, n_2, \dots, n_w$  as the number of observations collected from all  $w$  workers ( $n = \sum_{q=1}^w n_q$ ). We further assume that each worker’s submissions are exchangeable meaning there is no time dependency. And further, we assume each worker *shares* the same correlation. The BART model now becomes:

$$\mathbf{Y} = f(\mathbf{X}) + \boldsymbol{\varepsilon} \approx \mathfrak{F}_1^{\text{leaf}}(\mathbf{X}) + \mathfrak{F}_2^{\text{leaf}}(\mathbf{X}) + \dots + \mathfrak{F}_m^{\text{leaf}}(\mathbf{X}) + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}_n(\mathbf{0}, \sigma^2 \mathbf{B}) \quad (\text{A.12})$$

where  $\mathbf{X}$  is the  $n \times p$  design matrix *sorted* by from the first worker’s contributions to the last worker’s contribution and  $\mathbf{B}$  is a block diagonal matrix with the blocks being intraclass correlation matrices of dimensions  $n_1, n_2, \dots, n_w$ :

$$\mathbf{B} := \begin{bmatrix} \mathbf{D}_{n_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_{n_2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{D}_{n_w} \end{bmatrix} \quad \text{where}$$

$$\mathbf{D}_{n_q} := \begin{bmatrix} 1 & \rho & \cdots & \rho \\ \rho & 1 & \cdots & \rho \\ \vdots & \vdots & \ddots & \vdots \\ \rho & \rho & \cdots & 1 \end{bmatrix}_{n_q} = \rho(\mathbf{J}_{n_q} - \mathbf{I}_{n_q}) + \mathbf{I}_{n_q} \quad (\text{A.13})$$

Thus, we need to just update the Gibbs sampler to sample for  $\rho$  and update the likelihood calculations.

### A.8.2 The Update Metropolis-within-Gibbs Sampler

The new Gibbs sampler is below and we spend the rest of the section going over each piece that has changed.

$$\begin{aligned} 1 : & \quad \mathfrak{F}_1 \mid \mathbf{R}_{-1}, \sigma^2, \rho \\ 2 : & \quad \mathcal{L}_1 \mid \mathfrak{F}_1, \mathbf{R}_{-1}, \sigma^2, \rho \\ 3 : & \quad \mathfrak{F}_2 \mid \mathbf{R}_{-2}, \sigma^2, \rho \\ 4 : & \quad \mathcal{L}_2 \mid \mathfrak{F}_2, \mathbf{R}_{-2}, \sigma^2, \rho \\ & \quad \vdots \\ 2m - 1 : & \quad \mathfrak{F}_m \mid \mathbf{R}_{-m}, \sigma^2, \rho \\ 2m : & \quad \mathcal{L}_m \mid \mathfrak{F}_m, \mathbf{R}_{-m}, \sigma^2, \rho \\ 2m + 1 : & \quad \sigma^2 \mid \mathfrak{F}_1, \mathcal{L}_1, \dots, \mathfrak{F}_m, \mathcal{L}_m, \mathbf{E}, \rho \\ 2m + 2 : & \quad \rho \mid \mathfrak{F}_1, \mathcal{L}_1, \dots, \mathfrak{F}_m, \mathcal{L}_m, \mathbf{E}, \sigma^2 \end{aligned} \quad (\text{A.14})$$

We now tackle each posterior sampling in its own subsection.

### A.8.2.1 Posterior Sampling of $\vartheta_t$

The posterior  $\vartheta_t \mid \mathfrak{F}_t, \mathbf{R}_{-t}, \sigma^2, \rho$  is actually a compound step sampling  $\mu_\ell$  for all  $\ell = 1, \dots, b$  terminal nodes in the tree,  $\mu_\ell \mid \mathbf{R}_{-t,\ell}, \sigma^2, \rho$ . In the standard Bayesian setup, we have:

$$\mathbb{P}(\mu_\ell \mid \mathbf{R}_{-t,\ell}, \sigma^2, \rho) \propto \underbrace{\mathbb{P}(\mathbf{R}_{-t,\ell} \mid \mu_\ell, \sigma^2, \rho)}_{\text{likelihood}} \underbrace{\mathbb{P}(\mu_\ell; \sigma_\mu^2)}_{\text{prior}}$$

The likelihood is multivariate normal sharing the same mean and the prior is univariate normal. Let  $\boldsymbol{\mu}_\ell := \mu_\ell \mathbf{1}_{n_\ell}$

$$\begin{aligned} \mathbb{P}(\mu_\ell \mid \mathbf{R}_{-t,\ell}, \sigma^2, \rho) &\propto \mathcal{N}_{n_\ell}(\boldsymbol{\mu}_\ell, \sigma^2 \mathbf{B}_\ell) \mathcal{N}(0, \sigma_\mu^2) \\ &= \frac{1}{(2\pi)^{n_\ell/2} |\sigma^2 \mathbf{B}_\ell|^{1/2}} \exp\left(-\frac{1}{2} (\mathbf{R}_{-t,\ell} - \boldsymbol{\mu}_\ell)^\top (\sigma^2 \mathbf{B}_\ell)^{-1} (\mathbf{R}_{-t,\ell} - \boldsymbol{\mu}_\ell)\right) \times \\ &\quad \frac{1}{\sqrt{2\pi\sigma_\mu^2}} \exp\left(-\frac{1}{2\sigma_\mu^2} \mu_\ell^2\right) \\ &\propto \exp\left(-\frac{1}{2} \left( \underbrace{\frac{1}{\sigma^2} (\mathbf{R}_{-t,\ell} - \boldsymbol{\mu}_\ell)^\top \mathbf{B}_\ell^{-1} (\mathbf{R}_{-t,\ell} - \boldsymbol{\mu}_\ell)}_{\text{quadratic form}} + \frac{1}{\sigma_\mu^2} \mu_\ell^2 \right)\right) \end{aligned}$$

The underbraced above is a quadratic form which can be written as:

$$(\mathbf{R}_{-t,\ell} - \boldsymbol{\mu}_\ell)^\top \mathbf{B}_\ell^{-1} (\mathbf{R}_{-t,\ell} - \boldsymbol{\mu}_\ell) = \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{\ell,ij}^{-1} (r_i - \mu_\ell)(r_j - \mu_\ell)$$

where  $b_{\ell,ij}^{-1}$  is the  $i, j$ th entry of  $\mathbf{B}_\ell^{-1}$  and  $r_i$  is the  $i$ th entry of  $\mathbf{R}_{-t,\ell}$ . Rewriting everything inside the overbraced terms, we find:

$$\frac{1}{\sigma^2} \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{\ell,ij}^{-1} r_i r_j - \frac{1}{\sigma^2} \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{\ell,ij}^{-1} r_i \mu_\ell - \frac{1}{\sigma^2} \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{\ell,ij}^{-1} r_j \mu_\ell + \frac{1}{\sigma^2} \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{\ell,ij}^{-1} \mu_\ell^2 + \frac{1}{\sigma_\mu^2} \mu_\ell^2$$

We now collect terms to arrive at the coefficients for a quadratic equation:

$$\begin{aligned}
& \underbrace{\left( \frac{1}{\sigma_\mu^2} + \frac{1}{\sigma^2} \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{\ell,ij}^{-1} \right)}_a \mu_\ell^2 + \underbrace{\left( -\frac{1}{\sigma^2} \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{\ell,ij}^{-1} (r_i + r_j) \right)}_b \mu_\ell \\
& + \underbrace{\frac{1}{\sigma^2} \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{\ell,ij}^{-1} r_i r_j}_c
\end{aligned} \tag{A.15}$$

We can safely ignore  $c$  since within the proportionality of the posterior, this term does not depend on our parameter of interest  $\mu_\ell$ . In order to complete the square, we need the above form to look like  $d(\mu_\ell - e)^2$  which means  $d = a$  and  $e = -\frac{b}{2a}$  (the  $d^2$  term can be safely ignored since it is not a function of  $\mu_\ell$  and disappears as a proportionality constant). Thus, we arrive at:

$$\begin{aligned}
\mathbb{P}(\mu_\ell \mid \mathbf{R}_{-t,\ell}, \sigma^2, \rho) & \propto \exp \left( -\frac{1}{2} a \left( \mu_\ell - \left( -\frac{b}{2a} \right) \right)^2 \right) \propto \mathcal{N} \left( -\frac{b}{2a}, \frac{1}{a} \right) \\
& = \mathcal{N} \left( \frac{\frac{1}{\sigma^2} \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{\ell,ij}^{-1} (r_i + r_j)}{2 \left( \frac{1}{\sigma_\mu^2} + \frac{1}{\sigma^2} \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{\ell,ij}^{-1} \right)}, \frac{1}{\frac{1}{\sigma_\mu^2} + \frac{1}{\sigma^2} \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{\ell,ij}^{-1}} \right)
\end{aligned}$$

We can further simplify this expression. If we organize the elements of  $\mathbf{R}_{-t,\ell}$  by worker, then since all workers are independent, we can add their contributions together because the elements of  $\mathbf{B}^{-1}$  are zeroes outside of a common worker's rows and columns.



$$\mu_\ell \mid \mathbf{R}_{-t,\ell}, \sigma^2, \rho \sim \mathcal{N} \left( \frac{\frac{1}{\sigma^2} \sum_{q=1}^w \sum_{i=1}^{n_{\ell_q}} \sum_{j=1}^{n_{\ell_q}} d_{\ell_q,ij}^{-1} (r_i + r_j)}{2 \left( \frac{1}{\sigma_\mu^2} + \frac{1}{\sigma^2} \sum_{q=1}^w \sum_{i=1}^{n_{\ell_q}} \sum_{j=1}^{n_{\ell_q}} d_{\ell_q,ij}^{-1} \right)}, \frac{1}{\frac{1}{\sigma_\mu^2} + \frac{1}{\sigma^2} \sum_{q=1}^w \sum_{i=1}^{n_{\ell_q}} \sum_{j=1}^{n_{\ell_q}} d_{\ell_q,ij}^{-1}} \right) \quad (\text{A.16})$$

The notation  $n_{\ell_q}$  refers to the number of the  $q$ th worker in the  $\ell$ th node. The notation  $d_{\ell_q,ij}^{-1}$  is the  $i, j$ th term of the  $q$ th diagonal matrix in  $\mathbf{B}$  (see Equation A.13). Each of the matrices  $\mathbf{D}$  is an “equicorrelation matrix” whose inverse is:

$$\begin{aligned} \mathbf{D}_{n_{\ell_q}}^{-1} &= \frac{1}{1-\rho} \mathbf{I}_{n_{\ell_q}} + -\frac{\rho}{(1-\rho)(1+(n_{\ell_q}-1)\rho)} \mathbf{J}_{n_{\ell_q}} \\ \Rightarrow d_{\ell_q,ij}^{-1} &= \begin{cases} -\frac{\rho}{(1-\rho)(1+(n_{\ell_q}-1)\rho)} + \frac{1}{1-\rho} & \text{when } i = j \\ -\frac{\rho}{(1-\rho)(1+(n_{\ell_q}-1)\rho)} & \text{when } i \neq j \end{cases} \end{aligned}$$

Now we can simplify some of the terms in the normal notation:

$$\begin{aligned} \sum_{i=1}^{n_{\ell_q}} \sum_{j=1}^{n_{\ell_q}} d_{\ell_q,ij}^{-1} &= n_{\ell_q} \left( \frac{1}{1-\rho} \right) + n_{\ell_q}^2 \left( -\frac{\rho}{(1-\rho)(1+(n_{\ell_q}-1)\rho)} \right) \\ &= \frac{n_{\ell_q}}{1-\rho} \left( 1 - \frac{n_{\ell_q}\rho}{1+(n_{\ell_q}-1)\rho} \right) \\ &= \frac{n_{\ell_q}}{1-\rho} \left( \frac{1+(n_{\ell_q}-1)\rho}{1+(n_{\ell_q}-1)\rho} - \frac{n_{\ell_q}\rho}{1+(n_{\ell_q}-1)\rho} \right) \\ &= \frac{n_{\ell_q}}{1-\rho} \frac{1-\rho}{1+(n_{\ell_q}-1)\rho} \\ &= \frac{n_{\ell_q}}{1+(n_{\ell_q}-1)\rho} \end{aligned} \quad (\text{A.17})$$

Now we consider the term  $\sum_{i=1}^{n_{\ell_q}} \sum_{j=1}^{n_{\ell_q}} d_{\ell_q,ij}^{-1} (r_i + r_j)$ . On the diagonal we have:

$$\begin{aligned}
& \sum_{i=1}^n \left( \frac{1}{1-\rho} - \frac{\rho}{(1-\rho)(1+(n_{\ell_q}-1)\rho)} \right) (r_i + r_i) \\
&= 2n_{\ell_q} \left( \frac{1}{1-\rho} - \frac{\rho}{(1-\rho)(1+(n_{\ell_q}-1)\rho)} \right) \bar{r}_q
\end{aligned}$$

On the off-diagonal we have:

$$\sum_{i \neq j} -\frac{\rho}{(1-\rho)(1+(n_{\ell_q}-1)\rho)} (r_i + r_j) = -2 \left( \frac{\rho}{(1-\rho)(1+(n_{\ell_q}-1)\rho)} \right) (n_{\ell_q}-1)n_{\ell_q}\bar{r}_q$$

Adding these two together we have:

$$\begin{aligned}
& n_{\ell_q} \left( \frac{1}{1-\rho} - \frac{\rho}{(1-\rho)(1+(n_{\ell_q}-1)\rho)} \right) \bar{r}_q - \\
& 2 \left( \frac{\rho}{(1-\rho)(1+(n_{\ell_q}-1)\rho)} \right) (n_{\ell_q}-1)n_{\ell_q}\bar{r}_q \\
&= 2n_{\ell_q}\bar{r}_q \left( \frac{1}{1-\rho} - \frac{\rho}{(1-\rho)(1+(n_{\ell_q}-1)\rho)} - (n_{\ell_q}-1)\frac{\rho}{(1-\rho)(1+(n_{\ell_q}-1)\rho)} \right) \\
&= 2n_{\ell_q}\bar{r}_q \left( \frac{1}{1-\rho} - n_{\ell_q}\frac{\rho}{(1-\rho)(1+(n_{\ell_q}-1)\rho)} \right) \\
&= 2n_{\ell_q}\bar{r}_q \frac{1}{1-\rho} \left( 1 - n_{\ell_q}\frac{\rho}{1+(n_{\ell_q}-1)\rho} \right) \\
&= 2n_{\ell_q}\bar{r}_q \frac{1}{1-\rho} \left( \frac{1-\rho}{1+(n_{\ell_q}-1)\rho} \right) \\
&= \frac{2n_{\ell_q}\bar{r}_q}{1+(n_{\ell_q}-1)\rho} = \frac{2\sum_{i=1}^{n_{\ell_q}} r_i}{1+(n_{\ell_q}-1)\rho} \tag{A.18}
\end{aligned}$$

Thus, our posterior sampling becomes:

$$\mathbb{P}(\mu_\ell \mid \mathbf{R}_{-t,\ell}, \sigma^2, \rho) = \mathcal{N} \left( \frac{\frac{1}{\sigma^2} \sum_{q=1}^w \frac{\sum_{i=1}^{n_{\ell q}} r_i}{1 + (n_{\ell q} - 1)\rho}}{\frac{1}{\sigma_\mu^2} + \frac{1}{\sigma^2} \sum_{q=1}^w \frac{n_{\ell q}}{1 + (n_{\ell q} - 1)\rho}}, \frac{1}{\frac{1}{\sigma_\mu^2} + \sum_{q=1}^w \frac{n_{\ell q}}{1 + (n_{\ell q} - 1)\rho}} \right)$$

### A.8.2.2 Posterior Sampling of $\sigma^2$

The standard Bayes Rule gives us:

$$\begin{aligned} \mathbb{P}(\sigma^2 \mid \mathbf{E}, \rho) &\propto \underbrace{\mathbb{P}(\mathbf{E} \mid \sigma^2, \rho)}_{\text{likelihood}} \underbrace{\mathbb{P}(\sigma^2; \nu, \lambda)}_{\text{prior}} \\ &= \mathcal{N}_n(\mathbf{0}, \sigma^2 \mathbf{B}) \text{InvGamma}\left(\frac{\nu}{2}, \frac{\nu\lambda}{2}\right) \\ &= \frac{1}{(2\pi)^{n/2} |\sigma^2 \mathbf{B}|^{1/2}} \exp\left(-\frac{1}{2} \mathbf{E}^\top (\sigma^2 \mathbf{B})^{-1} \mathbf{E}\right) \times \\ &\quad \frac{\left(\frac{\nu\lambda}{2}\right)^{\frac{\nu}{2}}}{\Gamma\left(\frac{\nu}{2}\right)} (\sigma^2)^{-\left(\frac{\nu}{2}+1\right)} \exp\left(-\frac{\frac{\nu\lambda}{2}}{\sigma^2}\right) \\ &\propto \frac{1}{(\sigma^2)^{n/2}} (\sigma^2)^{-\left(\frac{\nu}{2}+1\right)} \exp\left(-\frac{1}{2\sigma^2} (\mathbf{E}^\top \mathbf{B}^{-1} \mathbf{E} + \nu\lambda)\right) \\ &= (\sigma^2)^{-\left(\frac{\nu+n}{2}+1\right)} \exp\left(-\frac{\frac{1}{2} (\mathbf{E}^\top \mathbf{B}^{-1} \mathbf{E} + \nu\lambda)}{\sigma^2}\right) \\ &= \text{InvGamma}\left(\frac{\nu+n}{2}, \frac{\mathbf{E}^\top \mathbf{B}^{-1} \mathbf{E} + \nu\lambda}{2}\right) \end{aligned}$$

Now, we can simplify the quadratic form above. Ordering  $\mathbf{E}$  by worker and using the fact that  $\mathbf{B}$  is diagonal,

$$\mathbf{E}^\top \mathbf{B}^{-1} \mathbf{E} = \sum_{q=1}^w \mathbf{E}_q^\top \mathbf{D}_q^{-1} \mathbf{E}_q = \sum_{q=1}^w \underbrace{\sum_{i=1}^{n_{\ell q}} \sum_{j=1}^{n_{\ell q}} d_{\ell q, ij}^{-1} e_{q,i} e_{q,j}}_{\text{quadratic form}}$$

where  $\mathbf{E}_q$  is the error vector for worker  $q$ . The underbraced above becomes:

$$\begin{aligned}
& \left( \frac{1}{1-\rho} - \frac{\rho}{(1-\rho)(1+(n_{\ell_q}-1)\rho)} \right) \sum_{i=1}^{n_{\ell_q}} e_i^2 - \\
& \frac{\rho}{(1-\rho)(1+(n_{\ell_q}-1)\rho)} \sum_{i=1}^{n_{\ell_q}} \sum_{j=1}^{n_{\ell_q}} e_{q,i} e_{q,j} \mathbb{1}_{i \neq j} \\
&= \frac{1}{1-\rho} \left( 1 - \frac{\rho}{1+(n_{\ell_q}-1)\rho} \sum_{i=1}^{n_{\ell_q}} e_i^2 - \frac{\rho}{1+(n_{\ell_q}-1)\rho} \sum_{i=1}^{n_{\ell_q}} \sum_{j=1}^{n_{\ell_q}} e_{q,i} e_{q,j} \mathbb{1}_{i \neq j} \right) \\
&= \frac{1}{1-\rho} \left( \frac{1+n_{\ell_q}\rho-2\rho}{1+(n_{\ell_q}-1)\rho} \sum_{i=1}^{n_{\ell_q}} e_i^2 - \frac{\rho}{1+(n_{\ell_q}-1)\rho} \sum_{i=1}^{n_{\ell_q}} \sum_{j=1}^{n_{\ell_q}} e_{q,i} e_{q,j} \mathbb{1}_{i \neq j} \right) \\
&= \frac{1}{(1-\rho)(1+(n_{\ell_q}-1)\rho)} \left( (1+n_{\ell_q}\rho-2\rho) \underbrace{\sum_{i=1}^{n_{\ell_q}} e_i^2}_{SSE_q} - \rho \underbrace{\sum_{i=1}^{n_{\ell_q}} \sum_{j=1}^{n_{\ell_q}} e_{q,i} e_{q,j} \mathbb{1}_{i \neq j}}_{CPE_q} \right) \\
&= \frac{1}{(1-\rho)(1+(n_{\ell_q}-1)\rho)} ((1+n_{\ell_q}\rho-2\rho)SSE_q - \rho CPE_q) \tag{A.19}
\end{aligned}$$

where  $SSE_q$  is the sum of squared error in worker  $q$  and  $CPE_q$  is the sum of the cross products in worker  $q$  save the diagonal elements.

Thus, the sampling for  $\sigma^2$  becomes:

$$\mathbb{P}(\sigma^2 \mid \mathbf{E}, \rho) = \text{InvGamma} \left( \frac{\nu+n}{2}, \frac{\nu\lambda}{2} + \left( \frac{1}{2(1-\rho)} \sum_{q=1}^w \frac{(1+n_{\ell_q}\rho-2\rho)SSE_q - \rho CPE_q}{1+(n_{\ell_q}-1)\rho} \right) \right)$$

### A.8.2.3 Posterior Sampling of $\rho$

The standard Bayes Rule gives us:

$$\mathbb{P}(\rho \mid \mathbf{E}, \sigma^2) \propto \underbrace{\mathbb{P}(\mathbf{E} \mid \sigma^2, \rho)}_{\text{likelihood}} \underbrace{\mathbb{P}(\rho)}_{\text{prior}}$$

We now assume the prior is:

$$\mathbb{P}(\rho) = \mathcal{U}(-1, 1)$$

This leaves us with:

$$\begin{aligned} \mathbb{P}(\rho \mid \mathbf{E}, \sigma^2) &\propto \mathcal{N}_n(\mathbf{0}, \sigma^2 \mathbf{B}) \mathcal{U}(-1, 1) \\ &\propto \frac{1}{(2\pi)^{n/2} |\sigma^2 \mathbf{B}|^{1/2}} \exp\left(-\frac{1}{2} \mathbf{E}^\top (\sigma^2 \mathbf{B})^{-1} \mathbf{E}\right) (1) \\ &\propto \frac{1}{|\mathbf{B}|^{1/2}} \exp\left(-\frac{1}{2\sigma^2} \mathbf{E}^\top \mathbf{B}^{-1} \mathbf{E}\right) \end{aligned}$$

The determinant term above becomes:

$$\begin{aligned} |\mathbf{B}_\ell|^{1/2} &= (|\mathbf{D}_1| \cdot \dots \cdot |\mathbf{D}_{w_\ell}|)^{1/2} \\ &= (((n_1 - 1)\rho + 1)(1 - \rho)^{n_1 - 1} \cdot \dots \cdot ((n_{w_\ell} - 1)\rho + 1)(1 - \rho)^{n_{w_\ell} - 1})^{1/2} \\ &= \left( \prod_{q=1}^{w_\ell} ((n_q - 1)\rho + 1)(1 - \rho)^{n_q - 1} \right)^{\frac{1}{2}} \\ &= \left( (1 - \rho)^{n_\ell - w_\ell} \prod_{q=1}^{w_\ell} ((n_q - 1)\rho + 1) \right)^{\frac{1}{2}} \end{aligned} \tag{A.20}$$

Using the formula we found before, we have:

$$\begin{aligned} \mathbb{P}(\rho \mid \mathbf{E}, \sigma^2) &\propto (1 - \rho)^{-\frac{n-w}{2}} \left( \prod_{q=1}^w ((n_q - 1)\rho + 1) \right)^{-\frac{1}{2}} \times \\ &\exp\left(-\frac{1}{2\sigma^2(1 - \rho)(1 + (n_{\ell q} - 1)\rho)} \sum_{q=1}^w ((1 + n_{\ell q}\rho - 2\rho)SSE_q - \rho CPE_q)\right) \end{aligned}$$

This is clearly not a distribution we can directly sample from. Thus, we now

use a Metropolis-Hastings step which can be accepted or rejected. Denote  $\rho_*$  as the proposed value of  $\rho$ .

$$r = \frac{\mathbb{P}(\rho_* \rightarrow \rho) \mathbb{P}(\rho_* | \mathbf{E}, \sigma^2)}{\mathbb{P}(\rho \rightarrow \rho_*) \mathbb{P}(\rho | \mathbf{E}, \sigma^2)} \quad (\text{A.21})$$

We will propose a very simple jump distribution as  $\mathbb{P}(\rho \rightarrow \rho_*) = \text{U}(-1, 1)$ . The risk is a sampler that converges slowly. We will revisit this jump distribution later if the proportion of acceptances is too low. Since the jump ratio cancels, we are left with:

$$\begin{aligned} r &= \frac{\mathbb{P}(\rho_* | \mathbf{E}, \sigma^2)}{\mathbb{P}(\rho | \mathbf{E}, \sigma^2)} \\ &= \sqrt{\left( \frac{1-\rho}{1-\rho_*} \right)^{n-w} \prod_{q=1}^w \frac{(n_q-1)\rho+1}{(n_q-1)\rho_*+1} \exp \left( -\frac{1}{2\sigma^2} \left( \left( \frac{1}{(1-\rho_*)(1+(n_{\ell_q}-1)\rho_*)} \right) \sum_{q=1}^w ((1+n_{\ell_q}\rho_*-2\rho_*)SSE_q - \rho_*CPE_q) + \right. \right. \\ &\quad \left. \left. \left( \frac{1}{(1-\rho)(1+(n_{\ell_q}-1)\rho)} \right) \sum_{q=1}^w ((1+n_{\ell_q}\rho-2\rho)SSE_q - \rho CPE_q) \right) \right)} \end{aligned}$$

This in log form becomes

$$\begin{aligned} \ln(r) &= \frac{1}{2} \left( (n-w) \ln \left( \frac{1-\rho}{1-\rho_*} \right) + \sum_{q=1}^w \ln \left( \frac{(n_q-1)\rho+1}{(n_q-1)\rho_*+1} \right) \right) - \frac{1}{2\sigma^2} \left( \left( \frac{1}{(1-\rho_*)(1+(n_{\ell_q}-1)\rho_*)} \right) \sum_{q=1}^w ((1+n_{\ell_q}\rho_*-2\rho_*)SSE_q - \rho_*CPE_q) + \right. \\ &\quad \left. \left( \frac{1}{(1-\rho)(1+(n_{\ell_q}-1)\rho)} \right) \sum_{q=1}^w ((1+n_{\ell_q}\rho-2\rho)SSE_q - \rho CPE_q) \right) \end{aligned}$$

### A.8.3 Posterior Sampling of $\mathfrak{T}_t$

We now handle the Metropolis steps  $1, 3, \dots, 2m - 1$  for all three tree proposal steps.

Below is the Metropolis ratio where the parameter sampled is the tree and the data is the responses unexplained by other trees denoted by  $\mathbf{R}$ . We denote the new, proposal tree with an asterisk and the original tree without the asterisk.

$$r = \frac{\mathbb{P}(\mathfrak{T}_* \rightarrow \mathfrak{T}) \mathbb{P}(\mathfrak{T}_* \mid \mathbf{R}, \sigma^2, \rho)}{\mathbb{P}(\mathfrak{T} \rightarrow \mathfrak{T}_*) \mathbb{P}(\mathfrak{T} \mid \mathbf{R}, \sigma^2, \rho)} \quad (\text{A.22})$$

We accept a draw from the posterior distribution of trees if a draw from a standard uniform distribution is less than the value of  $r$ . Immediately we note that it is difficult (if not impossible) to calculate the posterior probabilities for the trees themselves. Instead, we employ Bayes' Rule,

$$\mathbb{P}(\mathfrak{T} \mid \mathbf{R}, \sigma^2) = \frac{\mathbb{P}(\mathbf{R} \mid \mathfrak{T}, \sigma^2, \rho) \mathbb{P}(\mathfrak{T} \mid \sigma^2, \rho)}{\mathbb{P}(\mathbf{R} \mid \sigma^2, \rho)},$$

and plug the result into Equation A.22 to obtain:

$$r = \underbrace{\frac{\mathbb{P}(\mathfrak{T}_* \rightarrow \mathfrak{T})}{\mathbb{P}(\mathfrak{T} \rightarrow \mathfrak{T}_*)}}_{\text{transition ratio}} \times \underbrace{\frac{\mathbb{P}(\mathbf{R} \mid \mathfrak{T}_*, \sigma^2, \rho)}{\mathbb{P}(\mathbf{R} \mid \mathfrak{T}, \sigma^2, \rho)}}_{\text{likelihood ratio}} \times \underbrace{\frac{\mathbb{P}(\mathfrak{T}_*)}{\mathbb{P}(\mathfrak{T})}}_{\text{tree structure ratio}}.$$

Note that the probability of the tree structure is independent of  $\sigma^2$  and  $\rho$ .

The goal of this section is to explicitly calculate  $r$  for all possible tree proposals — GROW, PRUNE and CHANGE. Note that the transition ratio and tree structure ratios are the same between iidBART and panel data BART and these expressions are explained in the arxiv document. Thus, we only talk derive the likelihood ratios

here.

Note that our actual implementation uses the following expressions in log form for numerical accuracy.

#### A.8.3.1 Grow Proposal

To calculate the likelihood, the tree structure determines which responses fall into which of the  $b$  terminal nodes. Thus,

$$\mathbb{P}(R_1, \dots, R_n \mid \mathbb{F}, \sigma^2, \rho) = \prod_{\ell=1}^b \mathbb{P}(R_{\ell_1}, \dots, R_{\ell_{n_\ell}} \mid \sigma^2, \rho)$$

where each term on the right hand side is the probability of responses in one of the  $b$  terminal nodes, which are independent by assumption. The  $R_\ell$ 's denote the data in the  $\ell$ th terminal node and where  $n_\ell$  denotes how many observations are in each terminal node and  $n = \sum_{\ell=1}^b n_\ell$ .

We now find an analytic expression for the node likelihood term. Remember, if the mean in each terminal node, which we denote  $\mu_\ell$ , was known, then we would have  $R_{\ell_1}, \dots, R_{\ell_{n_\ell}} \mid \mu_\ell, \sigma^2 \stackrel{iid}{\sim} \mathcal{N}(\mu_\ell, \sigma^2)$ . BART requires  $\mu_\ell$  to be margined out, allowing the Gibbs sampler in Equation A.14 to avoid dealing with jumping between continuous spaces of varying dimensions (Chipman et al., 2010, page 275). Recall that one of the BART model assumptions is a prior on the average value of  $\mu \sim \mathcal{N}(0, \sigma_\mu^2)$  and thus,

$$\mathbb{P}(R_{\ell_1}, \dots, R_{\ell_{n_\ell}} \mid \sigma^2, \rho) = \int_{\mathbb{R}} \mathbb{P}(R_{\ell_1}, \dots, R_{\ell_{n_\ell}} \mid \mu_\ell, \sigma^2, \rho) \mathbb{P}(\mu_\ell; \sigma_\mu^2) d\mu_\ell$$

Substituting, we find



$$\frac{1}{((2\pi)^{n_\ell+1} (\sigma^2)^{n_\ell} |\mathbf{B}| \sigma_\mu^2)^{\frac{1}{2}}} \times \int_{\mathbb{R}} \exp \left( -\frac{1}{2} \left( \frac{1}{\sigma^2} (\mathbf{R}_{-t,\ell} - \boldsymbol{\mu}_\ell)^\top \mathbf{B}_\ell^{-1} (\mathbf{R}_{-t,\ell} - \boldsymbol{\mu}_\ell) + \frac{1}{\sigma_\mu^2} \mu_\ell^2 \right) \right) d\mu_\ell$$

we need to complete the square and account for the constant additive term:

$$a\mu^2 + b\mu + c = d(\mu - e)^2 + f \quad \Rightarrow \quad d = a \quad \text{and} \quad e = -\frac{b}{2a} \quad \text{and} \quad f = c - \frac{b^2}{4a}$$

Thus the integral above using using Equation A.15 should become

$$\exp(f) \int_{\mathbb{R}} \exp(d(\mu - e)^2) d\mu_\ell$$

Substituting, we find:

$$\exp \left( -\frac{1}{2\sigma^2} \left( \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{\ell,ij}^{-1} r_i r_j - \frac{\left( \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{\ell,ij}^{-1} (r_i + r_j) \right)^2}{4\sigma^2 \left( \frac{1}{\sigma_\mu^2} + \frac{1}{\sigma^2} \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{\ell,ij}^{-1} \right)} \right) \right) \times \int_{\mathbb{R}} \exp \left( -\frac{1}{2} \left( \frac{1}{\sigma_\mu^2} + \frac{1}{\sigma^2} \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{\ell,ij}^{-1} \right) \left( \mu_\ell - \frac{\frac{1}{\sigma^2} \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{\ell,ij}^{-1} (r_i + r_j)}{2 \left( \frac{1}{\sigma_\mu^2} + \frac{1}{\sigma^2} \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{\ell,ij}^{-1} \right)} \right)^2 \right) d\mu_\ell$$

The above integral is a Gaussian integral which resolves to  $\sqrt{2\pi\text{var}}$  where “var” is the variance parameter. Thus the margined likelihood becomes:

$$\begin{aligned}
& \mathbb{P}(R_{\ell_1}, \dots, R_{\ell_{n_\ell}} \mid \sigma^2, \rho) \\
&= \frac{1}{((2\pi)^{n_\ell+1} (\sigma^2)^{n_\ell} |\mathbf{B}_\ell| \sigma_\mu^2)^{\frac{1}{2}}} \sqrt{\frac{2\pi}{\frac{1}{\sigma_\mu^2} + \frac{1}{\sigma^2} \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{\ell,ij}^{-1}}} \times \\
& \exp \left( -\frac{1}{2\sigma^2} \left( \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{\ell,ij}^{-1} r_i r_j - \frac{\left( \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{\ell,ij}^{-1} (r_i + r_j) \right)^2}{4\sigma^2 \left( \frac{1}{\sigma_\mu^2} + \frac{1}{\sigma^2} \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{\ell,ij}^{-1} \right)} \right) \right) \\
&= \left( (2\pi)^{n_\ell} (\sigma^2)^{n_\ell} |\mathbf{B}_\ell| \sigma_\mu^2 \left( \frac{1}{\sigma_\mu^2} + \frac{1}{\sigma^2} \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{\ell,ij}^{-1} \right) \right)^{-\frac{1}{2}} \times \\
& \exp \left( -\frac{1}{2\sigma^2} \left( \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{\ell,ij}^{-1} r_i r_j - \frac{\left( \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{\ell,ij}^{-1} (r_i + r_j) \right)^2}{4\sigma^2 \left( \frac{1}{\sigma_\mu^2} + \frac{1}{\sigma^2} \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{\ell,ij}^{-1} \right)} \right) \right) \quad (\text{A.23})
\end{aligned}$$

Since the likelihoods are solely determined by the terminal nodes, the proposal tree differs from the original tree by only the selected node to be grown, denoted by  $\ell$ , which becomes two daughters after the GROW step denoted by  $\ell_L$  and  $\ell_R$ . Hence, the likelihood ratio becomes:

$$\frac{\mathbb{P}(\mathbf{R} \mid \mathfrak{F}_*, \sigma^2)}{\mathbb{P}(\mathbf{R} \mid \mathfrak{F}, \sigma^2)} = \frac{\mathbb{P}(R_{\ell_{L,1}}, \dots, R_{\ell_{L,n_{\ell,L}}} \mid \sigma^2) \mathbb{P}(R_{\ell_{R,1}}, \dots, R_{\ell_{R,n_{\ell,R}}} \mid \sigma^2)}{\mathbb{P}(R_{\ell_1}, \dots, R_{\ell_{n_\ell}} \mid \sigma^2)} \quad (\text{A.24})$$

Plugging Equation A.23 into Equation A.24 three times yields the ratio for the GROW step. We begin with the non-exponential term. We can reduce it to

$$\sqrt{\frac{|B_\ell|}{\underbrace{|B_{\ell_L}| |B_{\ell_R}|}} \frac{\left(1 + \frac{\sigma_\mu^2}{\sigma^2} \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{\ell,ij}^{-1}\right)}{\left(1 + \frac{\sigma_\mu^2}{\sigma^2} \sum_{i=1}^{n_{\ell_L}} \sum_{j=1}^{n_{\ell_L}} b_{\ell_L,ij}^{-1}\right) \left(1 + \frac{\sigma_\mu^2}{\sigma^2} \sum_{i=1}^{n_{\ell_R}} \sum_{j=1}^{n_{\ell_R}} b_{\ell_R,ij}^{-1}\right)}} \quad (\text{A.25})$$

where  $n_{\ell_L}$  and  $n_{\ell_R}$  denote the number of data points in the newly grown left and right daughter nodes.

We can now use the determinant formula of Equation A.20 to simplify the underbraced term above:

$$(1 - \rho)^{w_{\ell_L} + w_{\ell_R} - w_\ell} \prod_{q=1}^w \frac{((n_{\ell_q} - 1)\rho + 1)}{\left((n_{\ell_{Lq}} - 1)\rho + 1\right) \left((n_{\ell_{Rq}} - 1)\rho + 1\right)} \quad (\text{A.26})$$

Since the  $B_\ell$ 's are diagonal, the second term can be rewritten as sums over the workers from Equation A.16:

$$\frac{\left(1 + \frac{\sigma_\mu^2}{\sigma^2} \sum_{q=1}^w \sum_{i=1}^{n_{\ell_q}} \sum_{j=1}^{n_{\ell_q}} d_{\ell_q,ij}^{-1}\right)}{\left(1 + \frac{\sigma_\mu^2}{\sigma^2} \sum_{q=1}^w \sum_{i=1}^{n_{\ell_{Lq}}} \sum_{j=1}^{n_{\ell_{Lq}}} d_{\ell_{Lq},ij}^{-1}\right) \left(1 + \frac{\sigma_\mu^2}{\sigma^2} \sum_{q=1}^w \sum_{i=1}^{n_{\ell_{Rq}}} \sum_{j=1}^{n_{\ell_{Rq}}} d_{\ell_{Rq},ij}^{-1}\right)}$$

Now we have expressions for the  $d_{\ell_q,ij}^{-1}$  terms from Equation A.17 which we substitute in to become:

$$\frac{\left(1 + \frac{\sigma_\mu^2}{\sigma^2} \sum_{q=1}^{w_\ell} \frac{n_{\ell_q}}{1 + (n_{\ell_q} - 1)\rho}\right)}{\left(1 + \frac{\sigma_\mu^2}{\sigma^2} \sum_{q=1}^{w_{\ell_L}} \frac{n_{\ell_{Lq}}}{1 + (n_{\ell_{Lq}} - 1)\rho}\right) \left(1 + \frac{\sigma_\mu^2}{\sigma^2} \sum_{q=1}^{w_{\ell_R}} \frac{n_{\ell_{Rq}}}{1 + (n_{\ell_{Rq}} - 1)\rho}\right)}$$

$$= \frac{\sigma^2 \left( \sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_\ell} \frac{n_{\ell q}}{1 + (n_{\ell q} - 1)\rho} \right)}{\left( \sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_{\ell_L}} \frac{n_{\ell_L q}}{1 + (n_{\ell_L q} - 1)\rho} \right) \left( \sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_{\ell_R}} \frac{n_{\ell_R q}}{1 + (n_{\ell_R q} - 1)\rho} \right)} \quad (\text{A.27})$$

Thus, the whole term becomes:

$$\sqrt{\sigma^2 (1 - \rho)^{w_{\ell_L} + w_{\ell_R} - w_\ell} \left( \prod_{q=1}^w \frac{(n_{\ell q} - 1) \rho + 1}{((n_{\ell_L q} - 1) \rho + 1) ((n_{\ell_R q} - 1) \rho + 1)} \right)} \times$$

$$\sqrt{\frac{\sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_\ell} \frac{n_{\ell q}}{1 + (n_{\ell q} - 1)\rho}}{\left( \sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_{\ell_L}} \frac{n_{\ell_L q}}{1 + (n_{\ell_L q} - 1)\rho} \right) \left( \sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_{\ell_R}} \frac{n_{\ell_R q}}{1 + (n_{\ell_R q} - 1)\rho} \right)}}$$

Now we examine the portion of the exponential term and we find:

$$\exp \left( -\frac{1}{2\sigma^2} \left( \sum_{i=1}^{n_{\ell_L}} \sum_{j=1}^{n_{\ell_L}} b_{\ell_L, ij}^{-1} r_i r_j - \frac{\left( \sum_{i=1}^{n_{\ell_L}} \sum_{j=1}^{n_{\ell_L}} b_{\ell_L, ij}^{-1} (r_i + r_j) \right)^2}{4\sigma^2 \left( \frac{1}{\sigma_\mu^2} + \frac{1}{\sigma^2} \sum_{i=1}^{n_{\ell_L}} \sum_{j=1}^{n_{\ell_L}} b_{\ell_L, ij}^{-1} \right)} \right) \right) \times$$

$$\exp \left( -\frac{1}{2\sigma^2} \left( \sum_{i=1}^{n_{\ell_R}} \sum_{j=1}^{n_{\ell_R}} b_{\ell_R, ij}^{-1} r_i r_j - \frac{\left( \sum_{i=1}^{n_{\ell_R}} \sum_{j=1}^{n_{\ell_R}} b_{\ell_R, ij}^{-1} (r_i + r_j) \right)^2}{4\sigma^2 \left( \frac{1}{\sigma_\mu^2} + \frac{1}{\sigma^2} \sum_{i=1}^{n_{\ell_R}} \sum_{j=1}^{n_{\ell_R}} b_{\ell_R, ij}^{-1} \right)} \right) \right) \times$$

$$\exp \left( \frac{1}{2\sigma^2} \left( \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{\ell, ij}^{-1} r_i r_j - \frac{\left( \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{\ell, ij}^{-1} (r_i + r_j) \right)^2}{4\sigma^2 \left( \frac{1}{\sigma_\mu^2} + \frac{1}{\sigma^2} \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{\ell, ij}^{-1} \right)} \right) \right)$$

Collecting terms inside the exponential and dropping the exponential notation

and the  $-\frac{1}{2\sigma^2}$  term, we arrive at:

$$\begin{aligned} & \sum_{i=1}^{n_{\ell_L}} \sum_{j=1}^{n_{\ell_L}} b_{\ell_L,ij}^{-1} r_i r_j + \sum_{i=1}^{n_{\ell_R}} \sum_{j=1}^{n_{\ell_R}} b_{\ell_R,ij}^{-1} r_i r_j - \sum_{i=1}^{n_{\ell}} \sum_{j=1}^{n_{\ell}} b_{\ell,ij}^{-1} r_i r_j + \\ & \frac{1}{4\sigma^2} \left( \frac{\left( \sum_{i=1}^{n_{\ell_L}} \sum_{j=1}^{n_{\ell_L}} b_{\ell_L,ij}^{-1} (r_i + r_j) \right)^2}{\frac{1}{\sigma_{\mu}^2} + \frac{1}{\sigma^2} \sum_{i=1}^{n_{\ell_L}} \sum_{j=1}^{n_{\ell_L}} b_{\ell_L,ij}^{-1}} - \frac{\left( \sum_{i=1}^{n_{\ell_L}} \sum_{j=1}^{n_{\ell_L}} b_{\ell_L,ij}^{-1} (r_i + r_j) \right)^2}{\frac{1}{\sigma_{\mu}^2} + \frac{1}{\sigma^2} \sum_{i=1}^{n_{\ell_L}} \sum_{j=1}^{n_{\ell_L}} b_{\ell_L,ij}^{-1}} - \frac{\left( \sum_{i=1}^{n_{\ell_R}} \sum_{j=1}^{n_{\ell_R}} b_{\ell_R,ij}^{-1} (r_i + r_j) \right)^2}{\frac{1}{\sigma_{\mu}^2} + \frac{1}{\sigma^2} \sum_{i=1}^{n_{\ell_R}} \sum_{j=1}^{n_{\ell_R}} b_{\ell_R,ij}^{-1}} \right) \end{aligned}$$

Since the  $\mathbf{B}_{\ell}$ 's are diagonal, all terms can be rewritten as sums over the workers from Equation A.16:

$$\begin{aligned} & \sum_{q=1}^{w_{\ell_L}} \sum_{i=1}^{n_{\ell_L}} \sum_{j=1}^{n_{\ell_L}} d_{\ell_{Lq},ij}^{-1} r_i r_j + \sum_{q=1}^{w_{\ell_R}} \sum_{i=1}^{n_{\ell_R}} \sum_{j=1}^{n_{\ell_R}} d_{\ell_{Rq},ij}^{-1} r_i r_j - \sum_{q=1}^{w_{\ell}} \sum_{i=1}^{n_{\ell}} \sum_{j=1}^{n_{\ell}} d_{\ell_q,ij}^{-1} r_i r_j + \quad (\text{A.28}) \\ & \frac{1}{4\sigma^2} \left( \frac{\left( \sum_{q=1}^{w_{\ell_L}} \sum_{i=1}^{n_{\ell_L}} \sum_{j=1}^{n_{\ell_L}} d_{\ell_{Lq},ij}^{-1} (r_i + r_j) \right)^2}{\frac{1}{\sigma_{\mu}^2} + \frac{1}{\sigma^2} \sum_{q=1}^{w_{\ell_L}} \sum_{i=1}^{n_{\ell_L}} \sum_{j=1}^{n_{\ell_L}} d_{\ell_{Lq},ij}^{-1}} - \frac{\left( \sum_{q=1}^{w_{\ell_L}} \sum_{i=1}^{n_{\ell_L}} \sum_{j=1}^{n_{\ell_L}} d_{\ell_{Lq},ij}^{-1} (r_i + r_j) \right)^2}{\frac{1}{\sigma_{\mu}^2} + \frac{1}{\sigma^2} \sum_{q=1}^{w_{\ell_L}} \sum_{i=1}^{n_{\ell_L}} \sum_{j=1}^{n_{\ell_L}} d_{\ell_{Lq},ij}^{-1}} - \frac{\left( \sum_{q=1}^{w_{\ell_R}} \sum_{i=1}^{n_{\ell_R}} \sum_{j=1}^{n_{\ell_R}} d_{\ell_{Rq},ij}^{-1} (r_i + r_j) \right)^2}{\frac{1}{\sigma_{\mu}^2} + \frac{1}{\sigma^2} \sum_{q=1}^{w_{\ell_R}} \sum_{i=1}^{n_{\ell_R}} \sum_{j=1}^{n_{\ell_R}} d_{\ell_{Rq},ij}^{-1}} \right) \end{aligned}$$

Now we can use the simplification found in Equation A.19 to reduce the first line in the above expression to:

$$\begin{aligned} & \frac{1}{1-\rho} \left( \sum_{q=1}^{w_{\ell_L}} \frac{1}{1+(n_{\ell_{Lq}}-1)\rho} \left( (1+n_{\ell_{Lq}}\rho-2\rho)SSR_{Lq} - \rho CPR_{Lq} \right) \right. \\ & + \sum_{q=1}^{w_{\ell_R}} \frac{1}{1+(n_{\ell_{Rq}}-1)\rho} \left( (1+n_{\ell_{Rq}}\rho-2\rho)SSR_{Rq} - \rho CPR_{Rq} \right) \\ & \left. - \sum_{q=1}^{w_{\ell}} \frac{1}{1+(n_{\ell_q}-1)\rho} \left( (1+n_{\ell_q}\rho-2\rho)SSR_q - \rho CPR_q \right) \right) \quad (\text{A.29}) \end{aligned}$$

Now we can use simplifications found in Equations A.18 and A.17 to reduce the second line of the above expression to (and then we further simplify it):

$$\begin{aligned}
& \frac{1}{4\sigma^2} \left( \frac{\left( \sum_{q=1}^{w_\ell} \frac{2 \sum_{i=1}^{n_{\ell q}} r_i}{1 + (n_{\ell q} - 1)\rho} \right)^2}{\frac{1}{\sigma_\mu^2} + \frac{1}{\sigma^2} \sum_{q=1}^{w_\ell} \frac{n_{\ell q}}{1 + (n_{\ell q} - 1)\rho}} - \frac{\left( \sum_{q=1}^{w_{\ell L}} \frac{2 \sum_{i=1}^{n_{\ell L q}} r_i}{1 + (n_{\ell L q} - 1)\rho} \right)^2}{\frac{1}{\sigma_\mu^2} + \frac{1}{\sigma^2} \sum_{q=1}^{w_{\ell L}} \frac{n_{\ell L q}}{1 + (n_{\ell L q} - 1)\rho}} - \frac{\left( \sum_{q=1}^{w_{\ell R}} \frac{2 \sum_{i=1}^{n_{\ell R q}} r_i}{1 + (n_{\ell R q} - 1)\rho} \right)^2}{\frac{1}{\sigma_\mu^2} + \frac{1}{\sigma^2} \sum_{q=1}^{w_{\ell R}} \frac{n_{\ell R q}}{1 + (n_{\ell R q} - 1)\rho}} \right) \\
&= \frac{1}{4} \left( \frac{\left( \sum_{q=1}^{w_\ell} \frac{2 \sum_{i=1}^{n_{\ell q}} r_i}{1 + (n_{\ell q} - 1)\rho} \right)^2}{\frac{\sigma^2}{\sigma_\mu^2} + \sum_{q=1}^{w_\ell} \frac{n_{\ell q}}{1 + (n_{\ell q} - 1)\rho}} - \frac{\left( \sum_{q=1}^{w_{\ell L}} \frac{2 \sum_{i=1}^{n_{\ell L q}} r_i}{1 + (n_{\ell L q} - 1)\rho} \right)^2}{\frac{\sigma^2}{\sigma_\mu^2} + \sum_{q=1}^{w_{\ell L}} \frac{n_{\ell L q}}{1 + (n_{\ell L q} - 1)\rho}} - \frac{\left( \sum_{q=1}^{w_{\ell R}} \frac{2 \sum_{i=1}^{n_{\ell R q}} r_i}{1 + (n_{\ell R q} - 1)\rho} \right)^2}{\frac{\sigma^2}{\sigma_\mu^2} + \sum_{q=1}^{w_{\ell R}} \frac{n_{\ell R q}}{1 + (n_{\ell R q} - 1)\rho}} \right) \\
&= \sigma_\mu^2 \left( \frac{\left( \sum_{q=1}^{w_\ell} \frac{\sum_{i=1}^{n_{\ell q}} r_i}{1 + (n_{\ell q} - 1)\rho} \right)^2}{\sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_\ell} \frac{n_{\ell q}}{1 + (n_{\ell q} - 1)\rho}} - \frac{\left( \sum_{q=1}^{w_{\ell L}} \frac{\sum_{i=1}^{n_{\ell L q}} r_i}{1 + (n_{\ell L q} - 1)\rho} \right)^2}{\sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_{\ell L}} \frac{n_{\ell L q}}{1 + (n_{\ell L q} - 1)\rho}} - \frac{\left( \sum_{q=1}^{w_{\ell R}} \frac{\sum_{i=1}^{n_{\ell R q}} r_i}{1 + (n_{\ell R q} - 1)\rho} \right)^2}{\sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_{\ell R}} \frac{n_{\ell R q}}{1 + (n_{\ell R q} - 1)\rho}} \right) \quad (\text{A.30})
\end{aligned}$$

Putting it all together, the log ratio is:

$$\begin{aligned}
& \frac{1}{2} \left( \ln(\sigma^2) + (w_{\ell L} + w_{\ell R} - w_\ell) \ln(1 - \rho) + \sum_{q=1}^{w_\ell} \left( \ln \left( \frac{(n_{\ell q} - 1)\rho + 1}{((n_{\ell L q} - 1)\rho + 1)((n_{\ell R q} - 1)\rho + 1)} \right) \right) + \right. \\
& \ln \left( \sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_\ell} \frac{n_{\ell q}}{1 + (n_{\ell q} - 1)\rho} \right) - \ln \left( \sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_{\ell L}} \frac{n_{\ell L q}}{1 + (n_{\ell L q} - 1)\rho} \right) - \ln \left( \sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_{\ell R}} \frac{n_{\ell R q}}{1 + (n_{\ell R q} - 1)\rho} \right) \Big) \\
& + \frac{\sigma_\mu^2}{2\sigma^2} \left( \frac{\left( \sum_{q=1}^{w_{\ell L}} \frac{\sum_{i=1}^{n_{\ell L q}} r_i}{1 + (n_{\ell L q} - 1)\rho} \right)^2}{\sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_{\ell L}} \frac{n_{\ell L q}}{1 + (n_{\ell L q} - 1)\rho}} + \frac{\left( \sum_{q=1}^{w_{\ell R}} \frac{\sum_{i=1}^{n_{\ell R q}} r_i}{1 + (n_{\ell R q} - 1)\rho} \right)^2}{\sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_{\ell R}} \frac{n_{\ell R q}}{1 + (n_{\ell R q} - 1)\rho}} - \frac{\left( \sum_{q=1}^{w_\ell} \frac{\sum_{i=1}^{n_{\ell q}} r_i}{1 + (n_{\ell q} - 1)\rho} \right)^2}{\sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_\ell} \frac{n_{\ell q}}{1 + (n_{\ell q} - 1)\rho}} \right) \\
& - \frac{1}{2\sigma^2(1 - \rho)} \left( \sum_{q=1}^{w_{\ell L}} \frac{1}{1 + (n_{\ell L q} - 1)\rho} \left( (1 + n_{\ell L q}\rho - 2\rho)SSR_{Lq} - \rho CPR_{Lq} \right) + \right. \\
& \sum_{q=1}^{w_{\ell R}} \frac{1}{1 + (n_{\ell R q} - 1)\rho} \left( (1 + n_{\ell R q}\rho - 2\rho)SSR_{Rq} - \rho CPR_{Rq} \right) - \\
& \left. \sum_{q=1}^{w_\ell} \frac{1}{1 + (n_{\ell q} - 1)\rho} \left( (1 + n_{\ell q}\rho - 2\rho)SSR_q - \rho CPR_q \right) \right) \quad (\text{A.31})
\end{aligned}$$

A good way to check if this is correct is to set  $\rho = 0$  and it should simplify to the log form of the iidBART GROW ratio.

### A.8.3.2 Prune Proposal

This is simply the inverse of the likelihood ratio for the grow proposal. Thus the log-likelihood ratio is just the negative of Equation A.31.

### A.8.3.3 Change

The proposal tree differs from the original tree only in the two daughter nodes of the selected change node. These two terminal nodes have the unexplained responses apportioned differently. Denote  $R_{1.}$  as the residuals of the first daughter node and  $R_{2.}$  as the unexplained responses in the second daughter node. Thus we begin with:

$$\frac{\mathbb{P}(\mathbf{R} \mid \mathfrak{F}_*, \sigma^2)}{\mathbb{P}(\mathbf{R} \mid \mathfrak{F}, \sigma^2)} = \frac{\mathbb{P}(R_{L_*1}, \dots, R_{L_*n_{L_*}} \mid \sigma^2) \mathbb{P}(R_{R_*1}, \dots, R_{R_*n_{R_*}} \mid \sigma^2)}{\mathbb{P}(R_{L1}, \dots, R_{L_{N_L}} \mid \sigma^2) \mathbb{P}(R_{R1}, \dots, R_{R_{N_R}} \mid \sigma^2)} \quad (\text{A.32})$$

Substitution Equation A.23 four times into the above Equation A.32, we find a constant term and an exponential term. We begin with the constant term which is similar to Equation A.25:

$$\sqrt{\underbrace{\frac{|B_{L_*}| |B_{R_*}|}{|B_L| |B_R|}}_{\text{constant term}} \frac{\left(1 + \frac{\sigma_\mu^2}{\sigma^2} \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{L_*,ij}^{-1}\right) \left(1 + \frac{\sigma_\mu^2}{\sigma^2} \sum_{i=1}^{n_\ell} \sum_{j=1}^{n_\ell} b_{R_*,ij}^{-1}\right)}{\left(1 + \frac{\sigma_\mu^2}{\sigma^2} \sum_{i=1}^{n_{L_L}} \sum_{j=1}^{n_{L_L}} b_{L,ij}^{-1}\right) \left(1 + \frac{\sigma_\mu^2}{\sigma^2} \sum_{i=1}^{n_{R_R}} \sum_{j=1}^{n_{R_R}} b_{R,ij}^{-1}\right)}}$$

We can simplify the above underbraced term similar to Equation A.26 to find:

$$(1 - \rho)^{(w_{L_*} + w_{R_*}) - (w_L + w_R)} \prod_{q=1}^w \frac{((n_{L_q} - 1) \rho + 1) ((n_{R_q} - 1) \rho + 1)}{((n_{L_*q} - 1) \rho + 1) ((n_{R_*q} - 1) \rho + 1)}$$

We can then simplify the other term and it is similar to what we found in Equ-

tion A.27:

$$\frac{\left(\sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_{L*}} \frac{n_{L*q}}{1 + (n_{L*q} - 1)\rho}\right) \left(\sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_{R*}} \frac{n_{R*q}}{1 + (n_{R*q} - 1)\rho}\right)}{\left(\sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_L} \frac{n_{Lq}}{1 + (n_{Lq} - 1)\rho}\right) \left(\sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_R} \frac{n_{Rq}}{1 + (n_{Rq} - 1)\rho}\right)}$$

Now for the exponential component of Equation A.32. Collecting terms inside the exponential and dropping the exponential notation and the  $-\frac{1}{2\sigma^2}$  term, we arrive at an expression similar to Equation A.28:

$$\begin{aligned} & \sum_{q=1}^{w_{L*}} \sum_{i=1}^{n_{L*}} \sum_{j=1}^{n_{L*}} d_{L*q,ij}^{-1} r_i r_j + \sum_{q=1}^{w_{R*}} \sum_{i=1}^{n_{R*}} \sum_{j=1}^{n_{R*}} d_{R*q,ij}^{-1} r_i r_j - \\ & \sum_{q=1}^{w_L} \sum_{i=1}^{n_L} \sum_{j=1}^{n_L} d_{Lq,ij}^{-1} r_i r_j - \sum_{q=1}^{w_R} \sum_{i=1}^{n_R} \sum_{j=1}^{n_R} d_{Rq,ij}^{-1} r_i r_j \end{aligned}$$

plus  $\frac{1}{4\sigma^2}$  times the following

$$\begin{aligned} & \frac{\left(\sum_{q=1}^{w_L} \sum_{i=1}^{n_L} \sum_{j=1}^{n_L} d_{Lq,ij}^{-1} (r_i + r_j)\right)^2}{\frac{1}{\sigma_\mu^2} + \frac{1}{\sigma^2} \sum_{q=1}^{w_L} \sum_{i=1}^{n_L} \sum_{j=1}^{n_L} d_{Lq,ij}^{-1}} + \frac{\left(\sum_{q=1}^{w_R} \sum_{i=1}^{n_R} \sum_{j=1}^{n_R} d_{Rq,ij}^{-1} (r_i + r_j)\right)^2}{\frac{1}{\sigma_\mu^2} + \frac{1}{\sigma^2} \sum_{q=1}^{w_R} \sum_{i=1}^{n_R} \sum_{j=1}^{n_R} d_{Rq,ij}^{-1}} - \\ & \frac{\left(\sum_{q=1}^{w_{L*}} \sum_{i=1}^{n_{L*}} \sum_{j=1}^{n_{L*}} d_{L*q,ij}^{-1} (r_i + r_j)\right)^2}{\frac{1}{\sigma_\mu^2} + \frac{1}{\sigma^2} \sum_{q=1}^{w_{L*}} \sum_{i=1}^{n_{L*}} \sum_{j=1}^{n_{L*}} d_{L*q,ij}^{-1}} - \frac{\left(\sum_{q=1}^{w_{R*}} \sum_{i=1}^{n_{R*}} \sum_{j=1}^{n_{R*}} d_{R*q,ij}^{-1} (r_i + r_j)\right)^2}{\frac{1}{\sigma_\mu^2} + \frac{1}{\sigma^2} \sum_{q=1}^{w_{R*}} \sum_{i=1}^{n_{R*}} \sum_{j=1}^{n_{R*}} d_{R*q,ij}^{-1}} \end{aligned}$$

The first line of the tiny formula above can be reduced similarly to Equation A.29:



$$\begin{aligned}
& \frac{1}{1-\rho} \left( \sum_{q=1}^{w_{L*}} \frac{1}{1+(n_{L*q}-1)\rho} ((1+n_{L*q}\rho-2\rho)SSR_{L*q} - \rho CPR_{L*q}) \right. \\
& \quad + \sum_{q=1}^{w_{R*}} \frac{1}{1+(n_{R*q}-1)\rho} ((1+n_{R*q}\rho-2\rho)SSR_{R*q} - \rho CPR_{R*q}) \\
& \quad - \sum_{q=1}^{w_L} \frac{1}{1+(n_{Lq}-1)\rho} ((1+n_{Lq}\rho-2\rho)SSR_{Lq} - \rho CPR_{Lq}) \\
& \quad \left. - \sum_{q=1}^{w_R} \frac{1}{1+(n_{Rq}-1)\rho} ((1+n_{Rq}\rho-2\rho)SSR_{Rq} - \rho CPR_{Rq}) \right)
\end{aligned}$$

And the second line of the tiny formula can be reduced similarly to Equation A.30 as  $\sigma_\mu^2$  times:

$$\begin{aligned}
& \frac{\left( \sum_{q=1}^{w_L} \frac{\sum_{i=1}^{n_{Lq}} r_i}{1+(n_{Lq}-1)\rho} \right)^2}{\sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_L} \frac{n_{Lq}}{1+(n_{Lq}-1)\rho}} + \frac{\left( \sum_{q=1}^{w_R} \frac{\sum_{i=1}^{n_{Rq}} r_i}{1+(n_{Rq}-1)\rho} \right)^2}{\sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_R} \frac{n_{Rq}}{1+(n_{Rq}-1)\rho}} - \\
& \frac{\left( \sum_{q=1}^{w_{L*}} \frac{\sum_{i=1}^{n_{L*q}} r_i}{1+(n_{L*q}-1)\rho} \right)^2}{\sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_{L*}} \frac{n_{L*q}}{1+(n_{L*q}-1)\rho}} - \frac{\left( \sum_{q=1}^{w_{R*}} \frac{\sum_{i=1}^{n_{R*q}} r_i}{1+(n_{R*q}-1)\rho} \right)^2}{\sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_{R*}} \frac{n_{R*q}}{1+(n_{R*q}-1)\rho}}
\end{aligned}$$

Thus, the log ratio is below and similar to Equation A.31:

$$\begin{aligned}
& \frac{1}{2} \left( ((w_{L*} + w_{R*}) - (w_L + w_R)) \ln(1-\rho) \right. \\
& \quad + \sum_{q=1}^{w_\ell} \left( \ln \left( \frac{((n_{Lq}-1)\rho+1)((n_{Rq}-1)\rho+1)}{((n_{L*q}-1)\rho+1)((n_{R*q}-1)\rho+1)} \right) \right) \\
& \quad \left. + \ln \left( \sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_{L*}} \frac{n_{L*q}}{1+(n_{L*q}-1)\rho} \right) + \ln \left( \sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_{R*}} \frac{n_{R*q}}{1+(n_{R*q}-1)\rho} \right) \right)
\end{aligned}$$

$$\begin{aligned}
& -\ln \left( \sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_L} \frac{n_{Lq}}{1 + (n_{Lq} - 1)\rho} \right) \\
& -\ln \left( \sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_R} \frac{n_{Rq}}{1 + (n_{Rq} - 1)\rho} \right) \Bigg) \tag{A.33}
\end{aligned}$$

plus  $\frac{\sigma_\mu^2}{2\sigma^2}$  times:

$$\begin{aligned}
& \frac{\left( \sum_{q=1}^{w_{L*}} \frac{\sum_{i=1}^{n_{L*q}} r_i}{1 + (n_{L*q} - 1)\rho} \right)^2}{\sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_{L*}} \frac{n_{L*q}}{1 + (n_{L*q} - 1)\rho}} + \frac{\left( \sum_{q=1}^{w_{R*}} \frac{\sum_{i=1}^{n_{R*q}} r_i}{1 + (n_{R*q} - 1)\rho} \right)^2}{\sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_{R*}} \frac{n_{R*q}}{1 + (n_{R*q} - 1)\rho}} - \\
& \frac{\left( \sum_{q=1}^{w_L} \frac{\sum_{i=1}^{n_{Lq}} r_i}{1 + (n_{Lq} - 1)\rho} \right)^2}{\sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_L} \frac{n_{Lq}}{1 + (n_{Lq} - 1)\rho}} - \frac{\left( \sum_{q=1}^{w_R} \frac{\sum_{i=1}^{n_{Rq}} r_i}{1 + (n_{Rq} - 1)\rho} \right)^2}{\sigma^2 + \sigma_\mu^2 \sum_{q=1}^{w_R} \frac{n_{Rq}}{1 + (n_{Rq} - 1)\rho}}
\end{aligned}$$

minus  $\frac{1}{2\sigma^2(1-\rho)}$  times:

$$\begin{aligned}
& \sum_{q=1}^{w_{L*}} \frac{1}{1 + (n_{L*q} - 1)\rho} \left( (1 + n_{L*q}\rho - 2\rho)SSR_{L*q} - \rho CPR_{L*q} \right) + \\
& \sum_{q=1}^{w_{R*}} \frac{1}{1 + (n_{R*q} - 1)\rho} \left( (1 + n_{R*q}\rho - 2\rho)SSR_{R*q} - \rho CPR_{R*q} \right) - \\
& \sum_{q=1}^{w_L} \frac{1}{1 + (n_{Lq} - 1)\rho} \left( (1 + n_{Lq}\rho - 2\rho)SSR_{Lq} - \rho CPR_{Lq} \right) - \\
& \sum_{q=1}^{w_R} \frac{1}{1 + (n_{Rq} - 1)\rho} \left( (1 + n_{Rq}\rho - 2\rho)SSR_{Rq} - \rho CPR_{Rq} \right) \tag{A.34}
\end{aligned}$$

## Bibliography

- Abramovitz, M., Scitovsky, T., and Inkeles, A. (1973). Economic growth and its discontents. *Bulletin of the American Academy of Arts and Sciences*, pages 11–27.
- Akkaya, C., Conrad, A., Wiebe, J., and Mihalcea, R. (2010). Amazon mechanical turk for subjectivity word sense disambiguation. In *NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazons Mechanical Turk*, pages 195–203.
- Albert, J. and Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679.
- Alm, C., Roth, D., and Sproat, R. (2005). Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of the conference on Empirical Methods in Natural Language Processing*, pages 579–586. Association for Computational Linguistics.
- Aman, S. and Szpakowicz, S. (2007). Identifying expressions of emotion in text. In *Text, Speech and Dialogue*, pages 196–205. Springer.
- Angrist, J. D. (2001). Estimation of Limited Dependent Variable Models With Dummy Endogenous Regressors. *Journal of Business & Economic Statistics*, 19(1):2–28.
- Appleton, J. J., Christenson, S. L., Kim, D., and Reschly, A. L. (2006). Measuring cognitive and psychological engagement: Validation of the student engagement instrument. *Journal of School Psychology*, 44(5):427–445.
- Appleton, J. J., Christenson, S. L., Kim, D., and Reschly, A. L. (2008). Student engagement with school: Critical conceptual and methodological issues of the construct. *Psychology in the Schools*, 45(5):369–386.
- Ariely, D., Kamenica, E., and Prelec, D. (2008). Man’s Search for Meaning: The Case of Legos. *Journal of Economic Behavior & Organization*, 67(3-4):671 – 677.
- Atkinson, A. (1982). Optimum biased coin designs for sequential clinical trials with prognostic factors. *Biometrika*, 69(1):61–67.
- Bache, K. and Lichman, M. (2013). UCI machine learning repository.
- Bachrach, Y., Kosinski, M., Graepel, T., Kohli, P., and Stillwell, D. (2012). Personality and patterns of facebook usage. *Web Science*.

- Barankay, I. (2010). Rankings and social tournaments: Evidence from a field experiment. *University of Pennsylvania mimeo*.
- Begg, C. and Iglewicz, B. (1980). A treatment allocation procedure for sequential clinical trials. *Biometrics*, 36(1):81–90.
- Berinsky, A. J., Huber, G. A., and Lenz, G. S. (2012). Evaluating online labor markets for experimental research: Amazon.com’s mechanical turk. *Political Analysis*, 20:351–368.
- Berk, R. and Bleich, J. (in press). Statistical procedures for forecasting criminal behavior: A comparative assessment. *Criminology and Public Policy*.
- Blattenberger, G. and Fowles, R. (2014). Avalanche forecasting: Using bayesian additive regression trees (BART). In *Demand for Communications Services—Insights and Perspectives*, pages 211–227. Springer.
- Bleich, J., Kapelner, A., Jensen, S., and George, E. (2013). Using bart for variable selection. *ArXiv e-prints*.
- Boscardin, W. and Gelman, A. (1994). Bayesian Computation for Parametric Models of Heteroscedasticity in the Linear Model. *University of California, Berkeley*.
- Bottolo, L. and Richardson, S. (2010). Evolutionary stochastic search for bayesian model exploration. *Bayesian Analysis*, 5:583–618.
- Breiman, L. (2001a). Random forests. *Machine learning*, 45(1):5–32.
- Breiman, L. (2001b). Statistical Modeling: The Two Cultures. *Statistical Science*, 16(3):199–231.
- Breiman, L. and Cutler, A. (2013). Online manual for random forests. *Website*; [www.stat.berkeley.edu/~breiman/RandomForests/cc\\_home.htm](http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm).
- Breiman, L. and Friedman, J. (1985). Estimating Optimal Transformations for Multiple Regression and Correlation. *Journal of the American Statistical Association*, 80(391):580–598.
- Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). Classification and regression trees. wadsworth & brooks. *Monterey, CA*.
- Brown, S. W., Rood, T., and Palmer, M. (2010). Number or nuance: Which factors restrict reliable word sense annotation? In *LREC: The International Conference on Language Resources and Evaluation*, pages 3237–3243.
- Buja, A., Cook, D., Hofmann, H., Lawrence, M., Lee, E., Swayne, D., and Wickham, H. (2009). Statistical inference for exploratory data analysis and model diagnostics. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 367(1906):4361–83.
- Bush, A. and Parasuraman, A. (1984). Assessing response quality. A self-disclosure approach to assessing response quality in mall intercept and telephone interviews. *Psychology and Marketing*, 1(3-4):57–71.
- Cacioppo, J. and Petty, R. (1982). The need for cognition. *Journal of Personality and Social Psychology*, 42(1):116–131.
- Cacioppo, J., Petty, R., and Kao, C. (1984). The efficient assessment of need for cognition. *Journal of Personality Assessment*, 48(3):306–307.

- Callison-Burch, C. (2010). Creating speech and language data with Amazon’s Mechanical Turk. *NAACL HLT Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk*, pages 1–12.
- Carroll, R. and Ruppert, D. (1988). *Transformation and Weighting in Regression*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis.
- Cartwright, N. (2007). Are rcts the gold standard? *BioSocieties*, 2:11–20.
- Cepeda, E. and Gamerman, D. (2001). Bayesian modeling of variance heterogeneity in normal regression models. *Brazilian Journal of Probability and Statistics*, pages 207–221.
- Chall, J. and Dale, E. (1995). *Readability revisited: The new Dale-Chall readability formula*. Brookline Books.
- Chan, D., Kohn, R., Nott, D., and Kirby, C. (2006). Locally adaptive semiparametric estimation of the mean and variance functions in regression models. *Journal of Computational and Graphical Statistics*, pages 0–26.
- Chandler, D. and Kapelner, A. (2013). Breaking monotony with meaning: Motivation in crowdsourcing markets. *Journal of Economic Behavior & Organization*, 90:123–133.
- Chen, G., Jensen, S. T., and Stoeckert, C. J. (2007). Clustering of genes into regulons using integrated modeling - COGRIM. *Genome Biology*, 8:R4.
- Chilton, L., Horton, J., Miller, R., and Azenkot, S. (2010). Task search in a human computation market. *Proceedings of the ACM SIGKDD workshop on human computation*, pages 1–9.
- Chipman, H., George, E., and McCulloch, R. (2010). BART: Bayesian Additive Regressive Trees. *The Annals of Applied Statistics*, 4(1):266–298.
- Chipman, H. A., George, E. I., and McCulloch, R. E. (1998). Bayesian CART Model Search. *Journal of the American Statistical Association*, 93(443):935.
- Chklovski, T. and Mihalcea, R. (2003). Exploiting agreement and disagreement of human annotators for word sense disambiguation. In *Proceedings of the Conference on Recent Advances on Natural Language Processing*.
- Chow, S. and Chang, M. (2008). Adaptive design methods in clinical trials - a review. *Orphanet journal of rare diseases*, 3:11.
- Cook, T. and Campbell, D. (1979). *Quasi-Experimentation Design & Analysis Issues for Field Settings*. Houghton Mifflin Company.
- Couper, M. P., Tourangeau, R., and Marvin, T. (2009). Taking the Audio Out of Audio-CASI. *Public Opinion Quarterly*, 73(2):281–303.
- Croson, R. and Gneezy, U. (2009). Gender differences in preferences. *Journal of Economic Literature*, 47(2):448–474.
- Csikszentmihalyi, M. (1997). *Creativity: Flow and the psychology of discovery and invention*. Harper Perennial.
- Damien, P., Dellaportas, P., Polson, N., and Stephens, D. (2013). *Bayesian Theory and Applications*, pages 455–464. Oxford University Press, first edition.

- Davidov, D., Tsur, O., and Rappoport, A. (2010). Enhanced sentiment learning using twitter hashtags and smileys. In *Coling 2010: Posters*, pages 241–249, Beijing, China. Coling 2010 Organizing Committee.
- Davies, M. (2008). *The Corpus of Contemporary American English: 425 million words, 1990-present*. Available online at <http://corpus.byu.edu/coca/>.
- DeMaio, T. (1984). Social desirability and survey measurement: A review. *Surveying Subjective Phenomena*, 2:257–282.
- Deng, H. and Runger, G. (2012). Feature selection via regularized trees. *The 2012 International Joint Conference on Neural Networks (IJCNN)*.
- DeRubeis, R., Cohen, Z., Forand, N., Fournier, J., Gelfand, L., and Lorenzo-Luaces, L. (2013). The personalized advantage index: translating research on prediction into individual treatment recommendations a demonstration. *UPenn mimeo*.
- Díaz-Uriarte, R. and Alvarez de Andrés, S. (2006). Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, 7:3.
- Diener, E., Emmons, R. A., Larsen, R. J., and Griffin, S. (1985). The satisfaction with life scale. *Journal of personality assessment*, 49(1):71–75.
- Diener, E., Inglehart, R., and Tay, L. (2012). Theory and validity of life satisfaction scales. *Social Indicators Research*, pages 1–31.
- Diener, E., Ng, W., Harter, J., and Arora, R. (2010). Wealth and happiness across the world: material prosperity predicts life evaluation, whereas psychosocial prosperity predicts positive feeling. *Journal of Personality and Social Psychology*, 99(1):52.
- Diener, E. and Seligman, M. E. (2002). Very happy people. *Psychological Science*, 13(1):81–84.
- Ding, Y. and Simonoff, J. (2010). An investigation of missing data methods for classification trees applied to binary response data. *The Journal of Machine Learning Research*, 11:131–170.
- Dodds, P. S., Harris, K. D., Kloumann, I. M., Bliss, C. A., and Danforth, C. M. (2011). Temporal patterns of happiness and information in a global social network: Hedonometrics and twitter. *Diversity*, page 26.
- Druckman, J. N. and Kam, C. D. (2011). *Students as experimental participants: A defense of the ‘narrow data base’ in Handbook of experimental political science*, pages 41–57. Cambridge University Press.
- Easton, M. (2006). Britain’s happiness in decline. *BBC News*, 2.
- Efron, B. (1971). Forcing a sequential experiment to be balanced. *Biometrika*, 58(3):403–417.
- Efron, B. and Thisted, R. (1976). Estimating the number of unseen species: How many words did shakespeare know? *Biometrika*, 63(3):435–447.
- Eliashberg, J. (2010). *Green-lighting Movie Scripts: Revenue Forecasting and Risk Management*. PhD thesis, University of Pennsylvania.
- Elith, J., Leathwick, J., and Hastie, T. (2008). A working guide to boosted regression trees. *The Journal of Animal Ecology*, 77(4):802–13.
- Ericsson, K. A. (2002). Attaining excellence through deliberate practice: Insights from the study of expert performance. *The pursuit of excellence through education*, pages 21–55.

- Eriksson, K. and Simpson, B. (2010). Emotional reactions to losing explain gender differences in entering a risky lottery. *Judgment and Decision Making*, 5(3):159–163.
- Fan, R., Chang, K., Hsieh, C., Wang, X., and Lin, C. (2008). LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.
- Fellbaum, C., Grabowski, J., Landes, S., and L, S. (1997). Analysis of a hand-tagging task. In *Proceedings of ANLP-97 Workshop on Tagging Text with Lexical Semantics*, pages 34–40.
- Forgeard, M. J., Jayawickreme, E., Kern, M. L., and Seligman, M. E. (2011). Doing the right thing: Measuring wellbeing for public policy. *International Journal of Wellbeing*, 1(1).
- Foster, H., Hanno, P., Nickel, J., Payne, C., Mayer, R., Burks, D., Yang, C., Chai, T., Kreder, K., Peters, K., Lukacz, E., FitzGerald, M., Cen, L., Landis, J., Propert, K., Yang, W., Kusek, J., and Nyberg, L. (2010). Effect of amitriptyline on symptoms in treatment naïve patients with interstitial cystitis/painful bladder syndrome. *The Journal of urology*, 183(5):1853–8.
- Frankl, V. E. (1997). *Man’s search for ultimate meaning*. Insight Books/Plenum Press.
- Fredricks, J., McColskey, W., Meli, J., Montrosse, B., Mordica, J., and Mooney, K. (2011). Measuring student engagement in upper elementary through high school: A description of 21 instruments. *Issues & Answers Report, REL*, 98(098).
- Freedman, D. (2008). On regression adjustments to experimental data. *Advances in Applied Mathematics*, 40(2):180–193.
- Friedman, J. (1984). A variable span smoother. Technical Report LCS-TR-5, Stanford University, Lab for Computational Statistics.
- Friedman, J. (1991). Multivariate adaptive regression splines (with discussion and a rejoinder by the author). *The annals of statistics*, 19:1–67.
- Friedman, J. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics*, 29(5):1189–1232.
- Friedman, J. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378.
- Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1).
- Gacs, P. and Lovász, L. (1981). *Khachiyan’s algorithm for linear programming*. Springer.
- Gamerman, D. (1997). Sampling from the posterior distribution in generalized linear mixed models. *Statistics and Computing*.
- Gelman, A., Carlin, J., Stern, H., and Rubin, D. (2004). *Bayesian Data Analysis*. Chapman & Hall / CRC, second edition.
- Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 6:721–741.
- George, E. and McCulloch, R. (1993). Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, 88:881–889.
- George, E. I. and Robert, C. P. (1992). Capture-Recapture Estimation Via Gibbs Sampling. *Biometrika*, 79(4):677.

- Gilks, W. R., Richardson, S., and Spiegelhalter, D. J. (1996). *Markov chain Monte Carlo in practice*, volume 2. CRC press.
- Gill, A., Nowson, S., and Oberlander, J. (2009). What are they blogging about? personality, topic and motivation in blogs. *Proc. of AAAI ICWSM*.
- Gneezy, U. and List, J. A. (2006). Putting Behavioral Economics to Work: Testing for Gift Exchange in Labor Markets Using Field Experiments. *Econometrica*, 74(5):1365–1384.
- Golder, S. and Macy, M. (2011). Diurnal and seasonal mood vary with work, sleep, and daylength across diverse cultures. *Science*, 333(6051):1878–1881.
- Gramacy, R. B. (2007). tgp: An R Package for Bayesian Nonstationary, Semiparametric Nonlinear Regression and Design by Treed Gaussian Process Models. *Journal Of Statistical Software*, 19(9).
- Gramacy, R. B., Taddy, M., and Wild, S. M. (2013). Variable selection and sensitivity analysis using dynamic trees, with an application to computer code performance tuning. *The Annals of Applied Statistics*, 7(1):51–80.
- Green, D., Kern, H., and King, G. (2010). Modeling heterogeneous treatment effects in large-scale experiments using Bayesian Additive Regression Trees. In *The annual summer meeting of the society of political methodology*.
- Greevy, R., Lu, B., Silber, J., and Rosenbaum, P. (2004). Optimal multivariate matching before randomization. *Biostatistics (Oxford, England)*, 5(2):263–75.
- Halko, N., Martinsson, P., and Tropp, J. (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288.
- Han, B., Enas, N., and McEntegart, D. (2009). Randomization by minimization for unbalanced treatment allocation. *Statistics in medicine*, 28:3329–3346.
- Hans, C. (2009). Bayesian lasso regression. *Biometrika*, 96:835–845.
- Hans, C., Dobra, A., and West, M. (2007). Shotgun stochastic search for “large p” regression. *Journal of the American Statistical Association*, 102:507–516.
- Hansen, B. and Klopfer, S. (2006). Optimal full matching and related designs via network flows. *Journal of Computational and Graphical Statistics*, 15(3):609–627.
- Harrison, G. W. and List, J. A. (2004). Field Experiments. *Journal of Economic Literature*, 42(4):1009 – 1055.
- Hastie, T. (2013). *GAM: Generalized Additive Models*. R package version 1.09.
- Hastie, T. and Tibshirani, R. (1986). Generalized additive models. *Statistical Science*, 1(3):297–310.
- Hastie, T. and Tibshirani, R. (2000). Bayesian Backfitting. *Statistical Science*, 15(3):196–213.
- Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*. Springer Science, second edition.
- Hastings, W. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.
- Henrich, J., Heine, S., and Norenzayan, A. (2010). The weirdest people in the world. *Behavioral and Brain Sciences*, 33(2-3):61–83.



- Hocking, R. (1976). The analysis and selection of variables in linear regression. *Biometrics*, 32:1–49.
- Hoerl, A. and Kennard, R. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.
- Hoffman, J. and Subramaniam, B. (1995). The role of visual attention in saccadic eye movements. *Perception and Psychophysics*.
- Holmes, S. and Kapelner, A. (2010). Quality assessment of feature counting using a distributed workforce: Crowd counting a crowd. working paper.
- Holmes, S., Kapelner, A., and Lee, P. (2009). An interactive java statistical image segmentation system: Gemident. *Journal of Statistical Software*.
- Horton, J. and Chilton, L. (2010). The labor economics of paid crowdsourcing. In *Proceedings of the 11th ACM conference on Electronic commerce*, pages 209–218. ACM.
- Horton, J. J., Rand, D. G., and Zeckhauser, R. J. (2011). The online laboratory: conducting experiments in a real labor market. *Experimental Economics*, 14(3):399–425.
- Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., and Weischedel, R. (2006). OntoNotes: the 90 percent solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers on XX*, pages 57–60. Association for Computational Linguistics.
- Hu, F. and Rosenberger, W. (2006). *The Theory of Response-Adaptive Randomization in Clinical Trials*. John Wiley & Sons, Inc.
- Ide, N. and Véronis, J. (1998). Introduction to the special issue on word sense disambiguation: the state of the art. *Computational Linguistics*, 24(1):2–40.
- Ipeirotis, P. (2010). Demographics of Mechanical Turk. CeDER working paper CeDER-10-01, New York University, Stern School of Business.
- Ishwaran, H. and Rao, J. (2010). Generalized ridge regression: geometry and computational solutions when  $p$  is larger than  $n$ . *Technical Report*.
- Ishwaran, H., Rao, J., and Kogalur, U. (2013). *spikeslab : Prediction and variable selection using spike and slab regression*. R package version 1.1.5.
- Ishwaran, H. and Rao, J. S. (2005). Spike and slab variable selection: Frequentist and Bayesian strategies. *The Annals of Statistics*, 33(2):730–773.
- Jakulin, A., Mozina, M., and Demsar, J. (2005). Nomograms for visualizing support vector machines. In *KDD*, pages 108–117.
- Jennison, C. and Turnbull, B. (2000). *Group Sequential Methods with Applications to Clinical Trials*. Chapman & Hall / CRC.
- Jensen, S. T., Chen, G., and Stoeckert, C. J. (2007). Bayesian variable selection and data integration for biological regulatory networks. *Annals of Applied Statistics*, 1:612–633.
- Jiang, T. and Owen, A. (2002). Quasi-regression for visualization and interpretation of black box functions. Technical report, Stanford University mimeo.
- Kalish, L. and Begg, C. (1987). The impact of treatment allocation procedures on nominal significance levels and bias. *Controlled clinical trials*, 8(2):121–35.

- Kapelner, A. and Bleich, J. (2013). Bartmachine: A powerful tool for machine learning. *ArXiv e-prints*.
- Kapelner, A. and Chandler, D. (2010). Preventing Satisficing in Online Surveys : A “Kapcha” to Ensure Higher Quality Data. In *CrowdConf ACM Proceedings*.
- Kapelner, A., Holmes, S., and Lee, P. (2007a). [www.GemIdent.com](http://www.GemIdent.com).
- Kapelner, A., Kaliannan, K., Schwartz, H., Ungar, L. H., and Foster, D. P. (2012). New insights from coarse word sense disambiguation in the crowd. In *CoLING*.
- Kapelner, A., Lee, P., and Holmes, S. (2007b). An interactive statistical image segmentation and visualization system. *Medical Information Visualisation- Medevis -IEEE*.
- Kapelner, A. and Ungar, L. (2013). Crowdsourcing for statisticians designing applications and analyzing data on mturk. [https://dl.dropboxusercontent.com/u/93146/jsm\\_tutorial.pdf](https://dl.dropboxusercontent.com/u/93146/jsm_tutorial.pdf). [JSM 2013 Tutorial Slides].
- Kibler, D., Aha, D., and Albert, M. (1989). Instance-based prediction of real-valued attributes. *Computational Intelligence*, 5:51.
- Kindo, B., Wang, H., and Pe, E. (2013). MBACT - Multiclass Bayesian Additive Classification Trees. *arXiv*, pages 1–29.
- Kosinski, M. and Stillwell, D. (2012). mypersonality project. In <http://www.mypersonality.org/wiki/>.
- Kramer, A. (2010). An unobtrusive behavioral model of gross national happiness. In *Proc of the 28th int conf on Human factors in comp sys*, pages 287–290. ACM.
- Krippendorff, K. (1970). Estimating the Reliability, Systematic Error and Random Error of Interval Data. *Educational and Psychological Measurement*, 30(1):61–70.
- Krosnick, J. A. (1991). Response strategies for coping with the cognitive demands of attitude measures in surveys. *Applied Cognitive Psychology*, 5(3):213–236.
- Krosnick, J. A. (1999). Survey research. *Annual Review of Psychology*, 50:537–67.
- Krosnick, J. A. (2000). The threat of satisficing in surveys: the shortcuts respondents take in answering questions. *Survey Methods Centre Newsletter*.
- Krosnick, J. A., Narayan, S., and Smith, W. R. (1996). Satisficing in surveys: Initial evidence. *New Directions for Evaluation*, 1996(70):29–44.
- Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, pages 159–174.
- Lawless, N. M. and Lucas, R. E. (2011). Predictors of regional well-being: a county level analysis. *Social Indicators Research*, 101(3):341–357.
- Layard, R. (2005). Rethinking public economics: The implications of rivalry and habit. *Economics and Happiness*, pages 147–170.
- Lee, T., Rinaldi, N., Robert, F., Odom, D., Bar-Joseph, Z., Gerber, G., Hannett, N., Harbison, C., Thompson, C., Simon, I., Zeitlinger, J., Jennings, E., Murray, H., Gordon, D., Ren, B., Wyrick, J., Tagne, J., Volkert, T., Fraenkel, E., Gifford, D., and Young, R. (2002). Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science*, 298:763–764.

- Leslie, D., Kohn, R., and Nott, D. (2007). A general approach to heteroscedastic linear regression. *Statistics and Computing*, pages 1–29.
- Levitt, S. and List, J. (2009). Field experiments in economics: the past, the present, and the future. *European Economic Review*, 53(1):1–18.
- Levitt, S. D. and List, J. A. (2007). What Do Laboratory Experiments Measuring Social Preferences Reveal About the Real World? *Journal of Economic Perspectives*, 21(2):153–174.
- Li, F. and Zhang, N. (2010). Bayesian Variable Selection in Structured High-dimensional Covariate Spaces with Applications in Genomics. *Journal of the American Statistical Association*, 105:1978–2002.
- Liaw, A. and Wiener, M. (2002). Classification and Regression by randomForest. *R news*, 2:18–22.
- Little, R. (1993). Pattern-mixture models for multivariate incomplete data. *Journal of the American Statistical Association*, 88(421):125–134.
- Little, R. and Rubin, D. (2002). *Statistical analysis with missing data*. Wiley, second edition.
- Mairesse, F., Walker, M., Mehl, M., and Moore, R. (2007). Using linguistic cues for the automatic recognition of personality in conversation and text. *Journal of Artificial Intelligence Research*, 30(1):457–500.
- Mason, W. and Suri, S. (2011). Conducting behavioral research on Amazon’s Mechanical Turk. *Behavior research methods*, *Forthcoming*.
- McCrae, R. and Costa, P. (1987). Validation of the five-factor model of personality across instruments and observers. *Journal of personality and social psychology*, 52(1):81–90.
- McCrae, R. R. and John, O. P. (1992). An introduction to the five-factor model and its applications. *Journal of Personality*, 60(2):175–215.
- McDonald, R., Hannan, K., Neylon, T., Wells, M., and Reynar, J. (2007). Structured models for fine-to-coarse sentiment analysis. In *Annual Meeting-Association For Computational Linguistics*, volume 45, page 432.
- McEntegart, D. (2003). The pursuit of balance using stratified and dynamic randomization techniques: an overview. *Drug Information Journal*, 37:293–308.
- Meyer, G., Finn, S., Eyde, L., Kay, G., Moreland, K., Dies, R., Eisman, E., Kubiszyn, T., and Read, G. (2001). Psychological testing and psychological assessment: A review of evidence and issues. *American psychologist*, 56(2):128.
- Mihalcea, R. and Liu, H. (2006). A corpus-based approach to finding happiness. In *Proceedings of the AAAI Spring Symposium on Computational Approaches to Weblogs*, page 19.
- Miller, A. J. (2002). *Subset Selection in Regression*. New York: Chapman and Hall, 2nd edition.
- Mitchell, T. and Beauchamp, J. (1988). Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032.
- Morgan, K. and Rubin, D. (2012). Rerandomization to improve covariate balance in experiments. *The Annals of Statistics*, 40(2):1263–1282.
- Morris, M. D. (2011). *Design of Experiments: An Introduction Based on Linear Models*. Chapman & Hall / CRC Press.

- Navigli, R. (2009). Word sense disambiguation: A Survey. *ACM Computing Surveys*, 41(2):1–69.
- Nowson, S. (2007). Identifying more bloggers: Towards large scale personality classification of personal weblogs. In *In Proceedings of the International Conference on Weblogs and Social*.
- Oppenheimer, D. M., Meyvis, T., and Davidenko, N. (2009). Instructional manipulation checks: Detecting satisficing to increase statistical power. *Journal of Experimental Social Psychology*, 45(4):867–872.
- Otis, D. L., Burnham, K. P., White, G. C., and Anderson, D. R. (1978). *Statistical inference from capture data on closed animal populations*, volume 62. wildlife society.
- Pang, B. and Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 271. Association for Computational Linguistics.
- Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–86.
- Paolacci, G., Chandler, J., and Ipeirotis, P. (2010a). Running experiments on amazon mechanical turk. *Judgment and Decision Making*, 5(5):411–419.
- Paolacci, G., Chandler, J., and Ipeirotis, P. G. (2010b). Running experiments on amazon mechanical turk. *Judgment and Decision making*, 5(5):411–419.
- Parent, G. (2010). Clustering dictionary definitions using Amazon Mechanical Turk. *NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk*, pages 21–29.
- Park, T. and Casella, G. (2008). The bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686.
- Passonneau, R. J., Bhardwaj, V., Salieb-Aouissi, A., and Ide, N. (2011). Multiplicity and word sense: Evaluating and learning from multiply labeled word sense annotations. *Language Resources and Evaluation*.
- Pavot, W. and Diener, E. (1993). Review of the satisfaction with life scale. *Psychological Assessment*, 5(2):164.
- Pearlin, L. I. and Schooler, C. (1978). The structure of coping. *Journal of health and social behavior*, pages 2–21.
- Pennebaker, J. W., Chung, C., Ireland, M., Gonzales, A., and Booth, R. (2007). The development and psychometric properties of liwc2007. *Austin, TX, LIWC. Net*.
- Pennebaker, J. W., Francis, M. E., and Booth, R. J. (2001). Linguistic inquiry and word count: Liwc 2001. *Word J. Of The Int Ling Assoc*.
- Peterson, C. and Park, N. (2003). Positive psychology as the evenhanded positive psychologist views it. *Psychological Inquiry*, 14(2):143–147.
- Pinter-Wollman, N., Wollman, R., Guetz, A., Holmes, S., and Gordon, D. M. (2011). The effect of individual variation on the structure and function of interaction networks in harvester ants. *Journal of The Royal Society Interface*, 8(64):1562–1573.

- Plate, T., Bert, J., Grace, J., and Band, P. (2000). Visualizing the function computed by a feedforward neural network. *Neural computation*, 12(6):1337–53.
- Pocock, S. and Simon, R. (1975). Sequential treatment assignment with balancing for prognostic factors in the controlled clinical trial. *Biometrics*, 31(1):103–115.
- Pradhan, S., Loper, E., and Dligach, D. (2007). Semeval-2007 task-17: English lexical sample, srl and all words. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, pages 87–92.
- Pratola, M., Chipman, H., Higdon, D., McCulloch, R., and Rust, W. (2013). Parallel bayesian additive regression trees. Technical report, University of Chicago.
- Preston, A. E. (1989). The Nonprofit Worker in a For-Profit World. *Journal of Labor Economics*, 7(4):438–463.
- R Development Core Team (2005). R: A language and environment for statistical computing. ISBN 3-900051-07-0.
- Raghavarao, D. (1980). Use of distance function in sequential treatment assignment for prognostic factors in the controlled clinical trial. *Calcutta Statist. Assoc. Bulletin*, 29:99–102.
- Rao, J. and Potts, W. (1997). Visualizing Bagged Decision Trees. In *KDD*, pages 243–246.
- Raudenbush, S., Martinez, A., and Spybrook, J. (2007). Strategies for Improving Precision in Group-Randomized Experiments. *Educational Evaluation and Policy Analysis*, 29(1):5–29.
- Read, J. (2005). Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL Student Research Workshop*, pages 43–48, Ann Arbor, Michigan. Association for Computational Linguistics.
- Ridgeway, G. (2013). *GBM: Generalized Boosted Regression Models*. R package version 2.0-8.
- Robert, C. (2007). *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer.
- Rockova, V. and George, E. (2013). EMVS: The EM approach to variable selection. *Journal of the American Statistical Association*, page (in press).
- Rosen, S. (1986). The theory of equalizing differences. *Handbook of labor economics*, 1:641–692.
- Rosenbaum, P. (2010). *Design of observational studies*. Springer, New York.
- Rosenberger, W. and Sverdlov, O. (2008). Handling Covariates in the Design of Clinical Trials. *Statistical Science*, 23(3):404–419.
- Rosso, B. D., Dekas, K. H., and Wrzesniewski, A. (2010). On the meaning of work: A theoretical integration and review. *Research in Organizational Behavior*, 30:91–127.
- Rothwell, D. (2005). External validity of randomised controlled trials: “to whom do the results of this trial apply?”. *Lancet*, 365:82–93.
- Rowe, J. W. and Kahn, R. L. (1987). Human aging: usual and successful. *Science*.
- Rubin, D. (1978). Multiple imputations in sample surveys—a phenomenological Bayesian approach to nonresponse. Technical report, Proceedings of the Survey Research Methods Section, American Statistical Association.

- Rubin, D. (1979). Using multivariate matched sampling and regression adjustment to control bias in observational studies. *Journal of the American Statistical Association*, 74:318–328.
- Ruppert, D., Wand, P., and Carroll, R. (2003). *Semiparametric Regression*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.
- Russell, B., Torralba, A., and Murphy, K. (2008). **LabelMe**: a database and web-based tool for image annotation. *International Journal of Computer Vision*, pages 157–173.
- Ryan, R. M. and Deci, E. L. (2000). Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American psychologist*, 55(1):68.
- Ryff, C. D. and Keyes, C. L. M. (1995). The structure of psychological well-being revisited. *Journal of personality and social psychology*, 69(4):719.
- Saleem, S., Prasad, R., Vitaladevuni, S., Pacula, M., Crystal, M., Marx, B., Sloan, D., Vasterling, J., and Speroff, T. (2012). Automatic detection of psychological distress indicators and severity assessment from online forum posts. In *Proceedings of COLING 2012*, pages 2375–2388, Mumbai, India. The COLING 2012 Organizing Committee.
- Schafer, J. (1997). *Analysis of Incomplete Multivariate Data*. Chapman & Hall / CRC.
- Schaufeli, W. B., Bakker, A. B., and Salanova, M. (2006). The measurement of work engagement with a short questionnaire a cross-national study. *Educational and psychological Measurement*, 66(4):701–716.
- Schmidt, G., Mattern, R., and Schüler, F. (1981). Biomechanical investigation to determine physical and traumatological differentiation criteria for the maximum load capacity of head and vertebral column with and without protective helmet under effect. Technical report, EFC Research Program on Biomechanics of Impacts. Final report Phase III, Project G5. Institute für Rechtsmedizin, Universität Heidelberg.
- Schueller, S. M. and Seligman, M. E. (2010). Pursuit of pleasure, engagement, and meaning: Relationships to subjective and objective measures of well-being. *The Journal of Positive Psychology*, 5(4):253–263.
- Scott, N., McPherson, G., Ramsay, C., and Campbell, M. (2002). The method of minimization for allocation to clinical trials. a review. *Controlled clinical trials*, 23(6):662–74.
- Seligman, M. E. P. (2011). *Flourish: A Visionary New Understand of Happiness and Well-being*. Free Press.
- Senn, S. (2000). Consensus and controversy in pharmaceutical statistics. *Journal of the Royal Statistical Society: Series D*, 49(2):135–176.
- Setiadi, A. F., Ray, N. C., Kohrt, H. E., Kapelner, A., Carcamo-Cavazos, V., Levic, E. B., Yadegarynia, S., van der Loos, C. M., Schwartz, E. J., and Holmes, S. (2010). Quantitative, architectural analysis of immune cell subsets in tumor-draining lymph nodes from breast cancer patients and healthy lymph nodes. *PloS one*, 5(8):e12420.
- Shrout, P. E. and Fleiss, J. L. (1979). Intraclass correlations: uses in assessing rater reliability. *Psychological bulletin*, 86(2):420.
- Simon, H. (1955). A Behavioral Model of Rational Choice. *The Quarterly Journal of Economics*.
- Simon, R. (1979). Restricted randomization designs in clinical trials. *Biometrics*, 35(2):503–512.

- Singh, P., Lin, T., Mueller, E., Lim, G., Perkins, T., and Li Zhu, W. (2002). Open Mind Common Sense: Knowledge acquisition from the general public. *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*, pages 1223–1237.
- Smith, J. and Everhart, J. (1988). Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings of the Symposium on Computer Applications and Medical Care*, pages 261–265. IEEE Computer Society Press.
- Snow, R., O’Connor, B., Jurafsky, D., and Ng, A. (2008). Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pages 254–263. Association for Computational Linguistics.
- Sorokin, A. and Forsyth, D. (2008). Utility Data Annotation with Amazon Mechanical Turk. *First IEEE Workshop on Internet Vision, CVPR 08*.
- Sprouse, J. (2011). A validation of amazon mechanical turk for the collection of acceptability judgments in linguistic theory. *Behavior Research Methods*, 43(1):155–167.
- Steger, M. F., Kashdan, T. B., Sullivan, B. A., and Lorentz, D. (2008). Understanding the search for meaning in life: Personality, cognitive style, and the dynamic between seeking and experiencing meaning. *Journal of Personality*, 76(2):199–228.
- Stekhoven, D. and Bühlmann, P. (2012). MissForest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28:112–8.
- Stern, S. (2004). Do scientists pay to be scientists? *Management Science*, 50(6):835–853.
- Stiglitz, J. E., Sen, A., and Fitoussi, J. (2009). Report by the commission on the measurement of economic performance and social progress. *The Commission on the Measurement of Economic Performance and Social Progress website*. Available online at: <http://www.stiglitz-sen-fitoussi>.
- Stingo, F. and Vannucci, M. (2011). Variable selection for discriminant analysis with markov random field priors for the analysis of microarray data. *Bioinformatics*, 27:495–501.
- Strapparava, C. and Mihalcea, R. (2008). Learning to identify emotions in text. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 1556–1560. ACM.
- Strumbelj, E. and Kononenko, I. (2011). A General Method for Visualizing and Explaining Black-Box Regression Models. In Dobnikar, A., Lotric, R., and Ster, B., editors, *Adaptive and Natural Computing Algorithms, Part II*, chapter 3, pages 21–30. Springer.
- Student (1931). The Lanarkshire milk experiment. *Biometrika*, 23(3):398–406.
- Sumner, C., Byers, A., Boochever, R., and Park, G. (2012). Predicting dark triad personality traits from twitter usage and a linguistic analysis of tweets. [www.onlineprivacyfoundation.org](http://www.onlineprivacyfoundation.org).
- Taddy, M., Gramacy, R., and Polson, N. (2011). Dynamic Trees for Learning and Design. *Journal of the American Statistical Association*, 106(493):109–123.
- Tang, Y. and Chen, H. (2011). Emotion modeling from writer/reader perspectives using a microblog dataset. *Sentiment Analysis where AI meets Psychology (SAIIP)*, page 11.
- Taves, D. (1974). Minimization: a new method of assigning patients to treatment and control groups. *Clinical pharmacology and therapeutics*, 15(5):443.
- Tay, L., Tan, K., Diener, E., and Gonzalez, E. (2012). Social relations, health behaviors, and health outcomes: A survey and synthesis. *Applied Psychology: Health and Well-Being*.

- Taylor, S. (1965). Eye movements in reading: Facts and fallacies. *American Educational Research Journal*, 2(4):187.
- Taylor, S. E. (2011). Social support: A review. *The Oxford Handbook of Health Psychology*, pages 189–214.
- Thaler, R. (1985). Mental Accounting and Consumer Choice. *Marketing Science*, 4(3):199 – 214.
- Therneau, T. and Atkinson, E. (1997). An introduction to recursive partitioning using the rpart routines. Technical report, Mayo Foundation.
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288.
- Tourangeau, R., Rips, L., and Rasinski, K. (2000). *The Psychology of Survey Response*. Cambridge University Press.
- Troyanskaya, O., Cantor, M., and Sherlock, G. (2001). Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525.
- Turney, P. (2002). Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 417–424, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.
- Twala, B. (2009). An empirical comparison of techniques for handling incomplete data using decision trees. *Applied Artificial Intelligence*, 23(5):1–35.
- Twala, B., Jones, M., and Hand, D. (2008). Good methods for coping with missing data in decision trees. *Pattern Recognition Letters*, 29(7):950–956.
- Tzeng, F. (2005). Opening the Black Box - Data Driven Visualization of Neural Networks. In *Visualization*, pages 383–390. IEEE.
- Urbanek, S. (2011). *rJava: Low-level R to Java interface*. R package version 0.9-3 available on CRAN.
- van Rijsbergen, C. (1979). *Information Retrieval*. Butterworth, 2nd edition edition.
- Venables, W. and Ripley, B. (2002). *Modern Applied Statistics with S*. Springer, New York, fourth edition.
- Von Ahn, L., Blum, M., Hopper, N., and Langford, J. (2003). CAPTCHA: Using hard AI problems for security. *Advances in Cryptology EUROCRYPT 2003*.
- White, H. (1980). A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica: Journal of the Econometric Society*, 48(4):817–838.
- Whitehead, J. (1997). *The Design and Analysis of Sequential Clinical Trials*. John Wiley & Sons, Inc., 2nd edition.
- Wickham, H., Cook, D., Hofmann, H., and Buja, A. (2010). Graphical inference for Infovis. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):973–9.
- Wrzesniewski, A., McCauley, C., Rozin, P., and Schwartz, B. (1997). Jobs, careers, and callings: People’s relations to their work. *Journal of research in personality*, 31(1):21–33.



- Yarkoni, T. (2010). Personality in 100,000 Words: A large-scale analysis of personality and word use among bloggers. *Journal of Research in Personality*, 44(3):363–373.
- Yau, P. and Kohn, R. (2003). Estimation and variable selection in nonparametric heteroscedastic regression. *Statistics and Computing*, pages 191–208.
- Yip, P., Xi, L., Arnold, R., and Hayakawa, Y. (2005). A beta-binomial model for estimating the size of a heterogeneous population. *Australian & New Zealand Journal of Statistics*, 47(3):299–308.
- Zhou, Q. and Liu, J. S. (2008). Extracting sequence features to predict protein–DNA interactions: a comparative study. *Nucleic acids research*, 36(12):4137–4148.
- Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.