Algorithmic Approaches to Detecting Interviewer Fabrication in Surveys

Benjamin Birnbaum

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2012

Reading Committee:

Gaetano Borriello, Chair

Anna R. Karlin, Chair

Abraham D. Flaxman

Program Authorized to Offer Degree:

Department of Computer Science & Engineering

University of Washington

Abstract

Algorithmic Approaches to Detecting
Interviewer Fabrication in Surveys

Benjamin Birnbaum

Co-Chairs of the Supervisory Committee:

Professor Gaetano Borriello

Computer Science and Engineering

Professor Anna R. Karlin

Computer Science and Engineering

Surveys are one of the principal means of gathering critical data from low-income regions. Bad data, however, may be no better—or worse—than no data at all. Interviewer data fabrication, one cause of bad data, is an ongoing concern of survey organizations and a constant threat to data quality. In my dissertation work, I build software that automatically identifies interviewer fabrication so that supervisors can act to reduce it. To do so, I draw on two tool sets from computer science, one algorithmic and the other technological. On the algorithmic side, I use two sets of techniques from machine learning, supervised classification and anomaly detection, to automatically identify interviewer fabrication. On the technological side, I modify data collection software running on mobile electronic devices to record user traces that can help to identify fabrication. I show, based on the results of two empirical studies, that the combination of these approaches makes it possible to accurately and robustly identify interviewer fabrication, even when interviewers are aware that the algorithms are being used, have some knowledge of how they work, and are incentivized to avoid detection.

Table of Contents

List of Figures

List of Tables

To my parents—Janet, Stan, and Sharon—in profound appreciation.

Chapter 1
INTRODUCTION

Effective health and economic development requires data. Leaders need it to allocate resources and respond to emergencies [84], and NGOs need it to make decisions, monitor operations, evaluate effectiveness, improve performance, and document results [30, 89, 117]. Surveys are one of the principal means of acquiring data from low- and middle-income countries [111], but bad data may be no better—or worse—than no data at all. Interviewer data fabrication, one cause of bad data, is an ongoing concern of survey organizations and a constant threat to data quality.

In this dissertation, *I develop and validate general algorithmic techniques to detect interviewer data fabrication so that supervisors can remedy it*. I start, in this chapter, by motivating my work and describing my thesis and contributions in more detail. In Section 1.1, I describe the importance of survey data to development; in Section 1.2, I give background on the problem of interviewer fabrication; in Section 1.3, I provide more detail on my approach to identify fabrication; in Section 1.4, I state my thesis and contributions; and in Section 1.5, I outline the organization of the dissertation.

1.1 The importance of survey data to development

We live an increasingly data-driven world, and nowhere is this more apparent than in health and economic development. On the one hand, governments and funding organizations have pushed for increased *monitoring and evaluation*—the gathering, reporting, and evaluation of results of development operations [117]. On the other, researchers in academia, aid organizations, and industry have found new ways in which data can be used to guide development: institutions such as the Poverty Action Lab [1] and the Institute for Health Metrics and Evaluation [62, 84] use rigorous experimental designs, advanced statistical methods, and comprehensive data collection to evaluate development interventions and to gather global health metrics; NGOs such as Ushahidi [8, 113] use crowdsourcing to

gather data on natural disasters and for election monitoring; and academic research groups mine data sets from cell phone companies and social network websites to identify important events, such as earthquakes [66], food price inflation [49], and disease outbreaks [118].

Yet—despite the speed of innovation in using data for development—many of the world's poorest people are illiterate and live in isolated rural areas, remaining outside of modern data infrastructure. In the least-developed countries, 72% of people live in rural areas [110] and 41% of adults are illiterate [112]. Often, the only way to gather critical health and economic data about these people is to use *household surveys*, that is, to send interviewers house-to-house to interview members of the population orally. One use of household surveys, for example, is the *verbal autopsy*, a technique in which families of recently deceased people are interviewed and the results are given to physicians or computer algorithms to deduce the cause of death [83, 105]. These surveys can provide the only available information on the prevalence of various causes of death in countries that do not have any official cause-specific mortality data, of which there are currently 75 [78]. There are many other examples of household surveys, including government-sponsored censuses, aid agency data-collection efforts, economic and epidemiological research studies, clinical trials, and vaccination drives. Like verbal autopsy, all of these surveys provide critical information needed by leaders to make decisions and to allocate resources.

1.2 The problem of interviewer fabrication

Data quality is a major concern for any survey organization. Survey methodologists classify systematic (or *non-sampling*) error in survey data into five categories: *specification error*, when what researchers think is being measured is not what the survey is asking; *frame error*, when the target population is not correctly enumerated for random sampling; *nonresponse error*, when too many potential respondents do not agree to be surveyed; *processing error*, when mistakes are made in the data analysis after it is collected; and *measurement error*, when incorrect data is collected by interviewers [13].

Surveys in developing countries often face problems of limited supervision, chaotic environments, language barriers, high turnover, and low literacy. These problems motivate a special emphasis on the last of these categories, measurement error, which can have many different causes, such as unclear survey questions, respondents who forget information, or data collection mistakes.

One pernicious type of measurement error is interviewer data fabrication, also known as

curbstoning.[1] There are many reasons that an interviewer might fabricate data: households may be hard to reach; sensitive questions may be uncomfortable to ask; or payment may be based on the number of surveys completed. In one survey from the United States Census Bureau, at least 6.5% of interviewers were found to be falsifying at least some of their data [99]. In another survey, 6% of interviewers admitted to fabricating entire interviews and 13% of interviewers admitted to fabricating at least part of an interview—despite being supervised in a telephone call center [68]. Curbstoning is a major concern of many survey organizations [4, 13, 32, 76]. It has been discovered and reported in several surveys [21, 82, 99], and an "epidemic of suspected interview falsification" almost derailed at least one large epidemiological study [109].

Compounding the problem, the same reasons that make it so hard to gather data from low-resource rural regions in developing countries also make it hard to prevent curbstoning in these areas. Limited supervision, low training, high turnover, and mismatched incentives make overall interviewer error higher in surveys from the developing world [30, 50], and the rate of data fabrication may likely be higher as well.

1.3 My approach to detect interviewer fabrication

In this dissertation, I outline an approach to build software that automatically identifies curbstoning so that supervisors can act to reduce it. To build this software, I draw on two tool sets from computer science, one *algorithmic* and the other *technological*.

On the algorithmic side, I show how two sets of techniques from machine learning, *supervised classification* [116] and *anomaly detection* [27], can be used to automatically identify curbstoning. Both are natural techniques to apply to the problem. In supervised classification, an algorithm builds a model to fit a set of training data and then uses that model to make predictions on new data. Supervised classification has been used successfully for other problems in which the goal of the algorithm is to identify forgeries of some type, such as email spam detection [35].

Supervised classification makes sense if labeled training data is available. I will provide examples in this dissertation illustrating how to obtain such data, but sometimes, time and monetary constraints may preclude this possibility. In such cases, organizations need a technique that does not require training data, and unsupervised anomaly detection, the

[1] The word *curbstoning* brings to mind the image of an interviewer shirking his duties by sitting outside a house, on the curb, making up data instead of actually performing the interview [28]. However, the true origin of this word is unclear.

subfield of data mining dedicated to detecting surprising data, is a natural choice. One of its most important successful applications has been fraud detection [27], another problem that is similar to curbstoning detection.

On the technological side, I show how to modify data collection software running on *mobile electronic devices*—laptops, personal data assistants (PDAs), mobile phones, and tablets—to increase the accuracy of the algorithms. Such software tools have become increasingly popular following the dramatic increase in the use of mobile phones in developing countries. These tools, such as Pendragon Forms [93], EpiSurveyor [45], and Open Data Kit [6, 53, 52], have been used—to name a few examples—by government workers to implement socio-economic surveys, by microfinance institutions to record payments between lenders and borrowers, by indigenous tribes to catalog trees to participate in carbon markets, and by community health workers to manage household visits [53]. Just by making the process of data collection more efficient, they can make survey data more accurate [73, 119].

These tools can also make curbstoning detection algorithms more accurate. To do so, they can record user-interaction traces containing features like timing, change histories, and even the amount that users scroll. When curbstoners interact with their devices, they are not actually interviewing anyone; therefore, their interaction traces may differ from those of interviewers who are collecting real data. Curbstoners may, for example, spend less time on questions or change their answers more frequently. When algorithms are trained to recognize patterns such as these, their accuracy improves.

1.4 Thesis and contributions

I show that my approach of using supervised classification algorithms, unsupervised anomaly detection algorithms, and user trace data recorded by mobile data collection software can detect curbstoning with an accuracy of over 90%. Specifically, my thesis is the following.

> **Thesis:** Supervised classification and unsupervised anomaly detection algorithms can accurately and robustly identify curbstoning, even when interviewers are aware that the algorithms are being used, have some knowledge of how they work, and are incentivized to avoid detection. Furthermore, the accuracy of these algorithms can be improved by using user-trace data recorded by mobile electronic devices during data collection.

While arguing that this thesis is correct, I make several contributions, including to

- Summarize and synthesize related work on detecting curbstoning;
- Develop new anomaly detection algorithms and show that they can detect fabricated, or otherwise unusual, data;
- Show that standard supervised classification algorithms can detect curbstoning on realistic data sets, even when interviewers are aware that the algorithms are being used, have some knowledge of how they work, and are motivated to fabricate data in a way that avoids detection;
- Demonstrate that the accuracy of these algorithms can be improved by using user-trace data recorded by mobile electronic devices during data collection; and
- Provide recommendations on how to apply these techniques in practical data collection situations.

There are at least two limitations of my work. First, although I gather my data carefully and evaluate my algorithms rigorously, my data sets are not large, and therefore it is too early to conclude without further evaluation that my techniques generalize widely. Before using my algorithms in a large survey, implementers should perform incremental evaluations of my approach on their own survey data to ensure that their results are consistent with my own. Second, I focus on *how to detect* interviewer fabrication but not on *what to do* when it is detected; this leaves unexplored practical questions regarding how to communicate possible curbstoning to supervisors and how to deal with suspected curbstoners once they are identified. The answers to these questions depend on the specific contexts of the organizations involved. In my conclusion, I discuss some aspects of these questions, but I leave a detailed exploration for future work.

Because it is an important and well-defined problem, I focus on interviewer fabrication, even though it is not the only type of measurement error that can occur in surveys. Many of the ideas that I present in this dissertation, however, may apply to other data quality issues as well. For example, my algorithms may find instances of interviewer misunderstanding or poorly designed questions. Throughout this dissertation, I highlight how my ideas can apply to these broader data quality issues.

1.5 Organization of dissertation

The remainder of this dissertation is organized as follows.

In Chapter 2, I place my research in the context of two bodies of related work. The first, which is over 60 years old, comes from survey methodology. A major focus of this research has been to develop management practices that prevent curbstoning and to understand what traits tend to characterize fabricated data. My work builds on this research by moving from the question of *what* characterizes fabricated data to the question of *how* to detect it algorithmically. The second body of related work, which is much newer, comes from *Information and Communication Technology for Development (ICTD)*, an interdisciplinary field of technologists and social scientists who explore how modern computing and communication tools can be used to aid health and economic development. A major focus of these researchers has been to design tools that increase both the amount and quality of data collected in low-resource regions. My work—especially the part that shows that data collection software running on mobile electronic devices can be used to improve the accuracy of curbstoning detection algorithms—builds naturally on this area of research.

In Chapter 3, I describe the algorithmic methods that I use in this dissertation. Some of the algorithms that I describe are novel and are a direct contribution of my work. Others are not novel, but are drawn from the state of the art in supervised classification and unsupervised anomaly detection. Many of these algorithms have not been applied to detect curbstoning before; a contribution of my work is to show that they can be. I classify the curbstoning detection algorithms in Chapter 3 using a novel framework. The type of algorithm that should be used depends on the context. Throughout the chapter, I explain what should be considered when choosing the algorithm. I conclude the chapter by describing the evaluation methodologies that I use throughout the rest of the dissertation.

In Chapter 4, I describe my first evaluation of the algorithms developed in Chapter 3. In this evaluation, I measured the algorithms' ability to identify curbstoning in data sets from two East African NGOs. One of the data sets contained *labels* that indicated which data was real and which data was fabricated. To get these labels, I combined data that was known to be real with data that some interviewers fabricated specifically for the study. These labels allowed me to measure the performance of the algorithms quantitatively. I show that the supervised algorithms achieved an accuracy of up to 81% and that the unsupervised anomaly detection algorithms could find fabricated forms at a frequency substantially above chance. The other data sets are unlabeled; they do not provide an opportunity to evaluate the performance of the algorithms quantitatively. However, no part of these data sets was created specifically for the purpose of the study, so, in a sense, they are more authentic than the labeled data. I show that the unsupervised algorithms found compelling anomalies in these data sets. Although it is impossible to know what caused

these anomalies, their discovery points to the usefulness of the algorithms.

In Chapter 5, I describe the design and execution of an in-depth follow-up study that I performed to explore two parts of my thesis: that the curbstoning detection algorithms can work even if interviewers know they are being used and are incentivized to avoid detection; and that the accuracy of these algorithms can be improved by using user-trace data recorded by mobile electronic devices during data collection. To explore the first part, I had interviewers fabricate data in controlled settings. I varied how much they knew about the algorithms and whether or not they received a reward for fabricating data that avoided detection. To explore the second part, I modified the mobile data collection software Open Data Kit Collect [53] to record detailed user trace data during data collection. I outline how I did this and how I chose features to extract from these logs. These choices could be used as a template by others who wish to add similar capabilities to their mobile data collection software.

In Chapter 6, I describe the results of this study. First, I provide an analysis to validate the design of the study. I show that when I gave interviewers specific information on how the curbstoning detection algorithms worked, they changed their behavior in a way that suggests that they understood this information. Second, I show that—even when interviewers knew about the algorithms and were incentivized to avoid detection—supervised classification algorithms could detect curbstoning with an accuracy of up to 87%. When there was less knowledge and motivation from the interviewers, the accuracy of the algorithms was even higher, up to 92%. Third, I show that the user trace information from ODK Collect boosted the accuracy of the algorithms by up to 25%. Fourth, I show through simulations that these algorithms continued to perform well in several realistic scenarios, such as when it is impossible to rely on timing information or when there is noise in the training data labels. Fifth, I show that the unsupervised anomaly detection algorithms could find fabricated forms with an accuracy substantially above chance when the amount of fabrication is small. Sixth, and finally, I broaden the discussion by analyzing what traits tended to characterize fabricated data. This analysis gives intuition, which may generalize beyond the specifics of the study, into how the algorithms were able to distinguish the real data from the fake.

In Chapter 7, I conclude the dissertation. First, I summarize the evidence that I presented to support my thesis. Second, I provide recommendations on how to use my algorithms. In particular, I discuss what kind of practical impact can be expected if they are used, provide guidance on how to choose which algorithm to use, and mention important ethical considerations. Third, and finally, I suggest directions for future research.

Chapter 2
RELATED WORK

The work that I present in this dissertation builds on and extends two bodies of related work: research in survey methodology on curbstoning and research in ICTD on gathering data from low-resource regions. In this chapter, I describe both of these bodies of related work and explain where my contributions fit within their context.

2.1 Research on curbstoning in survey methodology

The academic study of curbstoning was initiated in 1945 by Crespi [36]. He gave several reasons that interviewers might fabricate data, including long questionnaires, complex or intrusive questions, unreasonable expectations, and hard to reach subjects. He proposed a two-pronged solution to remedy the problem: first, making careful survey design and management decisions to make cheating unlikely or unattractive and second, choosing a random sample of the respondents to reinterview to detect curbstoning. These two approaches have formed the basis of most of the work by survey methodologists to combat curbstoning.

The first approach, making design and management decisions to *prevent* cheating, was the focus of most of the early work on curbstoning [10, 11, 20, 46, 102].[1] Standard recommendations on the management side include hiring interviewers with good interpersonal and organizational skills, matching interviewers to respondents by socioeconomic characteristics, hiring experienced interviewers, carefully training interviewers in the concepts and objectives of the survey, and monitoring performance very closely at the beginning of interviewers' tenure. Besides reinterviews, techniques for monitoring interviewer performance include direct observation, audio recordings, call monitoring, and comparison of results to expected distributions [13].

The second approach, performing reinterviews to *detect* cheating, is tied to the need to un-

[1] For a more recent summary of management recommendations to prevent curbstoning, see the manual written in 2003 by the American Association for Public Opinion Research [3].

derstand what traits characterize fabricated data. If certain traits are known to characterize fabricated data, then the algorithm that chooses the respondents to reinterview could do so in a targeted way by searching for data exhibiting these traits [59]. Most of the curbstoning research in survey methodology has focused on what these traits are, instead of on how to design algorithms to detect fabricated data.

One notable exception is the recent paper by Li et al. [76]. In this paper, the authors employed a type of supervised classification algorithm called *logistic regression* (see Section 3.2.1) to find fake survey data in census data. They used this algorithm as a building block to design more targeted reinterview sampling algorithms. They showed, through simulation of their algorithm on historic data, that their new sampling algorithms were able to identify curbstoning more effectively than simple uniform sampling. Other authors have suggested other techniques to find fabricated data, including clustering algorithms [21] and a technique used to monitor quality in manufacturing processes called *statistical process control* [24, 59].

Understanding the characteristics of fabricated data is essential to design algorithms to detect it. In a literature search, I found 13 papers that describe traits of fabricated survey data [21, 33, 24, 59, 61, 65, 69, 76, 82, 95, 101, 108, 109].[2] In most of these papers, the authors had access to data sets from survey organizations in which some amount of fabrication was known; based on this data, they inferred and reported general traits that characterized it. These papers were published between 1981 and 2009. Four of them came from the United States Census Bureau [24, 59, 69, 76], which initiated an interviewer falsification study in 1982 to collect data on suspected and confirmed cases of falsification; four came from other large governmental institutions in the United States and Germany [33, 95, 101, 108]; three came from large-scale epidemiological studies from the developed world [61, 82, 109]; and two came from surveys done at least partly in the developing world [21, 65].

From these 13 papers, I extracted 15 traits of fabricated data that they collectively proposed. Table 2.1 classifies each of these papers according to which traits they propose. Sometimes

[2]To perform this literature search, I examined the first 50 results in Google Scholar for the following searches: "interviewer fabrication surveys," "curbstoning surveys," "interviewer cheating," "interviewer data falsification," and "survey bendfords law." I also searched through all 42 papers found by Google Scholar that cited Crespi's original 1945 paper on curbstoning [36]. I included a paper if it fit the following criteria: it was written after January 1, 1980 and before April 1, 2012; it was in English; it was a research article or technical report (not a book or manual); it was about interviewer fabrication of survey data; and it identified specific traits of the data itself that might characterize fabrication. For each paper that I included, I examined all references in the bibliography of the paper, and I searched through all citations of that paper on Google Scholar. Sometimes, there were very similar versions of a paper available. When this happened, I just chose one representative paper.

Variable	Inciardi (1981) [61]	Hood and Bushery (1997) [59]	Krejsa et al. (1999) [69]	Bushery et al. (1999) [24]	Turner et al. (2002) [109]	Cho et al. (2003) [33, 108]	Porras and English (2004) [95]	Shäfer et al. (2004) [101]	Murphy et al. (2004) [82]	Bredl et al. (2008) [21]	Judge and Schechter (2009) [65]	Li et al. (2009) [76]
Bad fit to Benford's Law						✓	✓	✓		✓	✓	
Missing telephone numbers		✓	✓		✓				✓			
Fast interviews			✓	✓					✓			✓
Low data variance	✓						✓	✓		✓		
Interview surge			✓	✓	✓							
Few missing units					✓				✓	✓		
Unusual data	✓				✓				✓			
Many missing units		✓	✓									
Close to deadline				✓								✓
Rare response combinations							✓		✓			
Short paths through survey		✓								✓		
Many incomplete interviews			✓									
Low time variance									✓			
Unusual time of day			✓									
Slow interviews									✓			

Table 2.1: Traits of fabricated data proposed in related work.

multiple papers propose the same trait, and sometimes different papers propose traits that contradict each other (e.g., *many missing units* and *few missing units*). In what follows, I will describe each of these 15 traits in more detail. In Chapter 5, I will revisit a subset of these traits when I design my own algorithms to detect fabrication.

I classified the 15 traits into five categories. The first two categories are based on the idea that curbstoners are motivated to save effort and not get caught. Hood and Bushery [59] summarized this thinking as follows: "Interviewers who falsify will try to keep it simple and fabricate a minimum of falsified data. They also will try to make it more difficult to check falsified cases."

The traits that are based on the idea that curbstoners are motivated to save effort are:

- *Fast interviews* [24, 69, 76, 82]. Curbstoners may fill out forms more quickly to save

effort and because they are not having real interactions with respondents.

- *Many missing units* [21, 59, 82]. Curbstoners might have a higher percentage of missing or ineligible units since that usually means that either fewer forms need to be filled out or the amount of work per form is smaller.
- *Interview surge* [24, 69, 109]. Fake interviews might occur in bursts of short time periods. This is consistent with an interviewer who wants to get a lot of interviews done at once in order to save time and effort.
- *Close to deadline* [24, 82]. Curbstoners might be motivated to cheat because they have deadline pressure. Therefore, faked interviews might be more likely to occur close to deadlines.
- *Short paths through survey* [21, 59]. Many surveys have branching logic, in which certain answers to some questions may mean that others are skipped. Fake interviews might be more likely to have a larger number of questions skipped because of the branching logic.
- *Unusual time of day* [69]. Curbstoners might fake interviews at home, where it is easier, and therefore fake interviews might happen at unusual times of the day.
- *Many incomplete interviews* [69]. Curbstoners may be more likely to abandon a higher percentage of interviews midway through the process.

The traits that are based on the idea that curbstoners try to avoid getting caught are:

- *Missing telephone numbers* [59, 69, 82, 109]. Many large surveys have reinterview programs, in which supervisors call a random sample of units to verify that interviews took place. Therefore, curbstoners may have a higher percentage of missing or unavailable phone numbers.
- *Low data variance* [21, 61, 95, 101]. To avoid raising suspicion, curbstoners may be more likely to choose typical values for answers.
- *Few missing units* [21, 82, 109]. Since missing units may arouse suspicion, curbstoners may mark fewer of their units as missing or unavailable.

Others base their approaches on the idea that interviews may lack knowledge of what the true population data looks like. Characteristics inspired by this idea are:

- *Bad fit to Benford's Law* [21, 33, 65, 95, 101]. Benford's Law [9, 56] states that in a wide

variety of contexts, the leading digit of numeric quantities tends to follow a certain decreasing distribution. Unless very knowledgable, people are unlikely to make up numbers that fit this law. This idea has been adopted to detect tax fraud [85], and many authors have also suggested (and shown) its utility in finding falsified numeric survey data.

- *Unusual data* [61, 82, 109]. Interviewers may not have a good sense for the true population averages of the quantities that they are measuring, especially when these quantities are stratified by other variables, such as sex and race [82]. Therefore, curbstoners may be found by looking for unusual response distributions.

- *Rare response combinations* [82, 95]. Even if interviewers know the overall population averages, they may choose unusual combinations of answers on individual forms, such as respondents who have done heroin but not smoked a cigarette [82].

Other characteristics proposed are:

- *Low time variance* [82]. Because curbstoners are not interviewing real people, there might be less variance in the amount of time they take to complete forms.

- *Slow interviews* [82]. Unusually slow interviews indicate something out of the ordinary and may therefore indicate cheating.

My work in relation to this research

My work builds on the research on curbstoning in survey methodology by moving from the question of *what characterizes fabricated data* to the question of *how it can be detected*. In this sense, I continue the line of research started in 2009 by Li et al. [76] that systematically and rigorously evaluates the potential of various algorithms to identify fabricated data.

To build on this work, I explore the potential of new algorithms to detect curbstoning, including advanced supervised classification techniques like random forest and the novel unsupervised anomaly detection techniques developed in Chapter 3. On the technological side, I show how user traces recorded by mobile data collection software can be used to dramatically improve the accuracy of the algorithms. My experimental designs are novel: I carefully control the level of knowledge that interviewers have regarding the curbstoning detection algorithms and whether or not they received a reward for fabricating data that avoided detection; doing so allows me to assess how these factors affect the accuracy of the algorithms. Finally, my work is the first to assess the robustness of anomaly detection algorithms to several common situations, such as when there is noise in the training data labels.

2.2 Research on data quality in ICTD

Much of the work by researchers and practitioners in ICTD has focused on technological approaches to increase the efficiency of collecting, digitizing, and organizing data in low-resource regions. This work can be divided into two categories: developing new data collection platforms and innovating the digitization of paper data. In what follows, I summarize this work.[3]

2.2.1 Developing new data collection platforms

Early work in ICTD used Personal Data Assistants (PDAs) as a platform for data collection. CyberTracker, for example, is specialized software that enables illiterate animal trackers to record observations on a PDA [37]. Pendragon Forms, as another example, is general-purpose data collection software that was built for PDAs (and that now runs on smart phones as well) [93]. It has been used to collect data on basic health information [104], tuberculosis [16], and respiratory tract infections [43]. Studies have shown that such software can improve the efficiency of data collection [17, 58, 71].

Later work in ICTD shifted toward using mobile phones as a platform. This shift was a natural consequence of the exponential growth of mobile phones in developing countries: in 2000 there were 11 million mobile phone users in Africa; in 2004 there were 53 million; and in 2008 there were 246 million [5]. Some researchers evaluated the potential of call-in systems for data collection, using either human operators or automatic *Interactive Voice Response* systems [92, 103]. Others developed software that used SMS to transmit data from phones to servers [48, 96, 12]. Application areas of mobile data collection include community health programs [40, 44, 67, 79, 121] (see Section 4.1) and micro-finance groups [88, 90, 97].

More recent work in ICTD has used *smart phones* as a platform. These phones have touch screens, significant processing power, and the ability to record multimedia data, such as photographs, audio, video, and GPS locations. Some researchers have argued that smartphones' combination of computing power, intuitive interfaces, portability, long-lasting batteries, and decreasing cost make them the ideal data collection and computation device in many development scenarios [5, 52]. General-purpose tools for data collection that run on smart phones include Pendragon Forms [93], EpiSurveyor [45], and Open Data Kit [6, 53, 52]. These software tools have been used for many purposes, including to implement socio-economic surveys, to record payments between lenders and borrowers, to

[3]The structure of my summary roughly follows the structure of a related work survey from Carl Hartung's Ph.D. dissertation [52].

catalog trees for participation in carbon markets, and to manage household visits by community health workers [53].

2.2.2 Innovating the digitization of paper data

Despite the rising popularity of new data collection platforms, in many contexts, paper remains the dominant medium for data collection [31, 39]. Researchers have therefore developed new technologies to more efficiently digitize paper data. CAM, for example, mixes paper systems with mobile-phone-based data entry [88, 90]. Users fill out paper forms and then use the phone's camera to scan barcodes on the forms; doing so triggers the appearance of prompts on the phone that allow the user to enter data and send it to a remote database. Shreddr, as another example, is a cloud-hosted service that combines batch processing, computer vision, and verification through crowdsourcing to transform data on paper forms to structured digital data [31]. ODK Scan, as a final example, is software that runs computer vision algorithms on phones with cameras to digitize data on forms with bubbles and checkboxes [39].

Taking a different approach, USHER is a software system that attempts to reduce data-entry errors by creating a probabilistic model of form data and using this model to optimize question ordering and to re-ask questions that seem likely to be incorrect [29]. Experiments have suggested that this approach can improve data quality and reduce entry cost [28].

My work in relation to this research

My work on developing algorithms to detect fabricated survey data can be seen as part of the broader ICTD research program dedicated to innovating the way that data is collected from the developing world. In this light, my work can be seen as a critical step in getting this valuable data: ensuring that it is real.

Although curbstoning detection algorithms can be used on data regardless of how it is collected, they can be especially useful when they are applied to data that is collected through mobile data collection software like Pendragon Forms, EpiSurveyor, or Open Data Kit. First, being able to detect fabricated data is most useful when it can be done as it is collected [109]. Mobile data collection software can make data available for analysis much faster than paper data collection can. Thus, when using the software, supervisors can run curbstoning detection algorithms on a regular basis (e.g., once a week) while data collection is still occurring—instead of waiting until all of the data is collected, digitized, and cleaned. This fine-grained approach may make it possible to fix data quality problems while there is still a chance to do so. Second, as I will show in Chapter 6, when mobile electronic devices are used during data collection, user-trace logs recorded by these tools

can be used to increase the accuracy of curbstoning detection algorithms.

My work also builds on USHER's approach of using advanced algorithmic techniques to improve data quality. Adding to this work, I develop novel unsupervised anomaly detection algorithms and test the algorithms' ability to detect fabricated data.

2.3 Chapter summary

In this chapter, I presented my contributions in the context of two bodies of related work: research in survey methodology on curbstoning and research in ICTD on gathering data from low-resource regions. In the context of survey methodology, I argued that my work brings a new focus to the algorithmic question of *how* to detect curbstoning. In the context of ICTD, I argued that my work fits within a broad research program searching for technological solutions for gathering higher-quality data from the developing world.

In the next chapter, I will develop and explain the curbstoning detection algorithms used in this dissertation.

Chapter 3
ALGORITHMIC METHODS

In this chapter, I describe the curbstoning detection algorithms that I evaluate in this dissertation. I start, in Section 3.1 by presenting a novel framework for classifying these algorithms. Then, in Section 3.2, I describe the supervised classification algorithms, and in Section 3.3, I describe the unsupervised anomaly detection algorithms. For both types of algorithm, I first provide general background information and then delve more deeply into the specific techniques that I employed. I conclude the chapter in Section 3.4 by outlining some of the evaluation techniques that I used to assess the success of these algorithms.

The unsupervised algorithms that I present in Section 3.3.2 are novel and are a direct contribution of this dissertation. The other algorithms that I present are not novel but are drawn from the state of the art in classification and anomaly detection algorithms; for some of these algorithms, their application to curbstoning detection is novel.

3.1 Framework

In this dissertation, I distinguish between two types of survey data. The first type of data, *response data*, is the data from the survey itself—it consists of the responses that are obtained through the interviews. The second type of data, *metadata*, is the data describing when, where, and how the response data was obtained. Metadata can include—to name a few examples—the time of day the interview started, GPS coordinates from where the interview took place, the time it took for each question to be completed, who did the interview, the order the questions were answered, or the amount the user scrolled through the question prompts. Many types of metadata can only be gathered if a mobile electronic device is used to conduct the interviews.

The data from a survey can be visualized as a two-dimensional array, in which each row corresponds to a single interview and each column corresponds to either a question or a metadata attribute. I will use the words *row* and *form* interchangeably, and I will use the words *column* and *feature* interchangeably. I will also use the term *response feature* to refer to

Dimension	Description
(S)upervised vs. (U)nsupervised	Whether the algorithm is a supervised classification algorithm or an unsupervised anomaly detection algorithm.
(F)orm-level vs. (I)nterviewer-level	Whether the algorithm makes predictions at the form level (one row) or the interviewer level (multiple rows).
(U)nivariate vs. (M)ultivariate	Whether the algorithm makes predictions at the question level (one column) or the form level (multiple columns).

Table 3.1: Dimensions of my framework.

a feature containing response data and the term *metafeature* to refer to a feature containing metadata.

I classify questions into four types:

- *numeric* – questions whose responses must be a real number or an integer
- *ordinal* – questions whose responses must be chosen from a finite set of possibilities that are not numeric but still have a natural ordering, e.g., a Likert-type scale [77]
- *categorical* – questions whose responses must be chosen from a finite set of possibilities that are not numeric and do not have a natural ordering, e.g., a `true/false` or a `yes/no/maybe` question
- *free text* – questions whose responses can be any string of alphanumeric characters

Other data types are possible, such as dates, times, or multimedia data, but the four types that I have listed are sufficient to describe the questions in the surveys from this dissertation.

I process ordinal questions by converting their responses to numeric values (see the description of the `ord` features in Section 5.5.1, for example). I do not use free text responses directly, but I do use features extracted from them, such as their length.

My framework divides curbstoning detection algorithms into eight groups based on three binary dimensions: *supervised* vs. *unsupervised*, *form-level* vs. *interviewer-level*, and *univariate* vs. *multivariate*. These dimensions are summarized in Table 3.1.

Curbstoning detection algorithms must predict which data is real and which data is fake. The first dimension of my framework, *supervised* vs. *unsupervised*, relates to *how* the predictions are made. In a supervised classification algorithm, a model is built on labeled training data and then used to make predictions, whereas in an unsupervised anomaly detection algorithm, outliers are identified based on a set of prior assumptions on the data. I discuss supervised classification in more depth in Section 3.2 and unsupervised anomaly detection in more depth in Section 3.3. Supervised algorithms are often more powerful than unsupervised algorithms because instead of relying on prior assumptions to make their predictions, they use training data to learn specific traits of fabricated data that may depend heavily on context. Often, however, getting high-quality training data may be expensive or time consuming. In such cases, unsupervised anomaly detection algorithms may make more sense. These algorithms have the additional advantage that anomalies are often of interest to data users regardless of whether they were caused by fabrication. They might, for example, indicate important phenomena like disease outbreaks.

The second dimension, *form-level* vs. *interviewer-level*, relates to the *unit* for which predictions are made. A form-level algorithm attempts to identify individual *forms* that have been fabricated, whereas an interviewer-level algorithm aggregates forms by interviewer and attempts to identify *interviewers* who are systematically fabricating data. This dimension draws a distinction between algorithms that make predictions for a single row at a time and algorithms that make predictions for multiple rows at a time. Form-level algorithms may be more flexible because they can identify more fine-grained fabrication. There are many reasons that an interviewer might fabricate some, but not all, of her interviews, including fatigue, deadline pressure, or difficult-to-reach households. On the other hand, interviewer-level algorithms may be more powerful because they aggregate information across interviews—doing so may allow them to identify patterns that might be missed otherwise. This advantage of interviewer-level algorithms can be seen in Sections 4.3.1 and 4.3.2 when I compare the performance of form- and interviewer-level algorithms on my data sets.

The third dimension, *univariate* vs. *multivariate*, also relates to the unit for which predictions are made, but concerns the columns instead of the rows. A univariate algorithm attempts to identify individual *questions* whose answers have been fabricated, whereas a multivariate algorithm lumps all questions together and attempts to identify entire *forms* that have been fabricated. Univariate algorithms may be more flexible because—like form-level algorithms—they can identify more fine-grained fabrication. There are many potential causes of question-level fabrication, including difficult or embarrassing questions,

hurried interviewers, or poorly-worded prompts. (Thus, identifying individual questions whose responses have been fabricated may help not only to ensure data quality but also to refine questionnaire design.) On the other hand, multivariate algorithms may be more powerful because they aggregate information across questions. This advantage of multivariate algorithms can be seen in Section 4.3.2 and in Section 6.5 when I compare the performance of univariate and multivariate algorithms on my data sets.

For convenience, I will use three-letter acronyms to to indicate the type of an algorithm according to my framework. I form the acronyms by combining the first letters of the class of the algorithm in each dimension. So, for example, I will describe an unsupervised, form-level, univariate algorithm with the acronym `UFU`, and I will describe a supervised, interviewer-level, multivariate algorithm with the acronym `SIM`.

To summarize, each of these dimension presents a flexibility-versus-power tradeoff. Unsupervised techniques do not require labeled training data but may not be as powerful as supervised techniques; form-level techniques may detect curbstoning when interviewers fabricate only a small portion of their data but may not be as powerful as interviewer-level techniques; and univariate techniques may detect issues at the question level but may not be as powerful as multivariate techniques. The choice of which algorithms to use will depend on the context of the survey organization; I discuss this choice in more detail in Section 7.1.2.

In this dissertation, I explore five of the eight possible types of algorithm. The reasons for not including the remaining three vary. Table 3.2 lists each of the eight types of algorithms. For each of the five types that I cover, the table states in which sections the algorithms are developed and evaluated. For each of the remaining three types, the table states why I did not cover them.

Table 3.3 shows the results of an experiment in which I measured the running times of the five algorithms described in this section. The table shows that—at least on my benchmark data set of 100 forms—the supervised algorithms are roughly an order of magnitude faster than the interviewer-level unsupervised algorithms, and the interviewer-level unsupervised algorithms are roughly two orders of magnitude faster than the form-level unsupervised algorithm LOCI. (It should, however, be kept in mind that the implementation language of the unsupervised algorithms, Python, is much slower than the implementation language of the supervised algorithms, Java.)

Type	Comments
SFU	*Supervised, form-level, univariate algorithms* are not covered in this dissertation. Such algorithms would amount to finding a cutoff, on one response feature, that best separates real data from fake data in a labeled training set. It is unlikely that one response feature would contain enough information to find a cutoff that robustly predicts question-level fabrication.
SFM	*Supervised, form-level, multivariate algorithms* covered in this dissertation include logistic regression (Section 3.2.1) and random forest (Section 3.2.2). They are evaluated in Sections 4.3, 6.2, 6.3, and 6.4.
SIU	*Supervised, interviewer-level, univariate algorithms* are not covered in this dissertation. Such algorithms would be similar to the UIU algorithms developed in Section 3.3.2, except that instead of simply seeking outlier response distributions, they would learn from training data the type of response distributions to seek. The development and investigation of this type of algorithm are left for future work because there are not enough interviewers in the data sets in this dissertation to train such an algorithm.
SIM	*Supervised, interviewer-level, multivariate algorithms* covered in this dissertation include logistic regression (Section 3.2.1) and random forest (Section 3.2.2). In the interviewer-level version of these algorithms, the scores output by the classification algorithm are averaged by interviewer (see Section 3.2). They are evaluated in Section 4.3.
UFU	*Unsupervised, form-level, univariate algorithms* are not covered in this dissertation. Such algorithms would be univariate outlier detection algorithms. These algorithms are already well-developed, and have been suggested for use in data quality control [55, 114].
UFM	*Unsupervised, form-level, multivariate algorithms* covered in this dissertation include LOCI, which is developed in Section 3.3.1 and evaluated in Sections 4.3 and 6.5.
UIU	*Unsupervised, interviewer-level, univariate algorithms* covered in this dissertation include MMA and SVA. These algorithms are developed in Section 3.3.2 and evaluated in Sections 4.3, 4.4, and 6.5.
UIM	*Unsupervised, interviewer-level, multivariate algorithms* covered in this dissertation include MMA and SVA (with the outlier scores averaged across all questions). These algorithms are developed in Section 3.3.2 and evaluated in Sections 4.3, 4.4, and 6.5.

Table 3.2: The eight types of curbstoning detection algorithm.

3.2 Supervised classification algorithms

Supervised classification is a subfield of machine learning and statistics [116]. The task of a supervised classification algorithm is to classify a set of *instances* as belonging to one of a

Algorithm	Time (s)
Logistic regression	0.028
Random forest	0.047
LOCI	14
Multinomial model algorithm (MMA)	0.17
s-value algorithm (SVA)	0.12

Table 3.3: Benchmark times for curbstoning detection algorithms. I performed this test on a MacBook Pro (Early 2011, 15-inch), with a 2.0 GHz quad-core Intel Core i7 processor. The time reported is the average time to build a model for the supervised algorithms and the average time to output outlier scores for the unsupervised algorithms. For each of the iterations, a random subset of 100 forms was chosen from the data set $\mathcal{A}_\ell$ described in Section 4.2.1. I used Weka's Java implementations of the supervised algorithms [51] and my own Python implementations of the unsupervised algorithms. The number of trees used in the random forest was 32 (see Section 3.2.2).

finite, discrete set of *classes*. In this dissertation, the instances are typically forms, and the set of classes is typically $\{\texttt{real}, \texttt{fake}\}$. A set of instances for which these classes are known, called the *training set*, is used to build a *model* of the data. The training algorithm chooses the parameters of the model by attempting to optimize a cost function that quantifies the the quality of the model's fit to the data. After this is done, the model is used to classify instances for which the class is not known. The type of model and the algorithm used to build it—both of which may vary—together define the supervised classification technique.

Supervised classification has been successfully used in a wide variety of practical scenarios, such as speech recognition, spam detection, handwriting recognition, and scientific image classification [81]. It has also been used to detect curbstoning. As described in Section 2.1, Li et al. [76] use logistic regression to detect curbstoning. In the notation of my framework, their approach is an `SFM` algorithm.

When faced with a classification task, choosing which machine learning algorithm to use can be a daunting task. There are many from which to choose (see e.g. [15, 80, 116]), and there is little guidance on how to decide. Because the best performing algorithm varies by data set, the recommended approach is often to try several algorithms and see which performs best.

For my dissertation, I chose two popular supervised classification algorithms: *logistic regression* and *random forest*. By necessity, this choice is ultimately a somewhat arbitrary matter of personal preference. However, there were some reasons for it. First, because

I use supervised classification not to find the absolute best classifier possible but instead to evaluate the potential of the approach *in general*, I decided to try only a small number of popular algorithms. Second, the two algorithms that I chose are complementary. Logistic regression is simple, widely-used in both statistics and machine learning, and fast. Random forest is less simple, but it may be one of the best general purpose classification algorithms [26]. Unlike logistic regression, random forest can efficiently represent non-linear decision boundaries,[1] and it tends to be more efficient than other popular non-linear classifiers, like support vector machines and neural networks.

Both logistic regression and random forest are typically multivariate algorithms. As mentioned earlier, instances are forms; thus, the most straightforward way to use these algorithms is at the form-level (i.e., as `SFM` algorithms). However, it is also possible to use them at the interviewer level (i.e., as `SIM` algorithms). This is because—before making a binary prediction of `real` or `fake`—both of these algorithms make an intermediate prediction in the form of a probability of being `fake`. To get interviewer-level predictions, I take the average of these probabilities over all the forms from each interviewer, and output `real` for the interviewer if this average probability is less than 0.5 and `fake` if this average probability is greater than 0.5.

To evaluate logistic regression and random forest, I used their implementations in Weka [51], a widely-used open-source machine learning library.[2] In Section 3.2.1, I describe how logistic regression works, and in Section 3.2.2, I describe how random forest works. In Section 3.2.3, I describe how I performed *model selection* for each of these two algorithms. Model selection is the important task of tuning parameters of supervised classification algorithms to get the best possible performance.

3.2.1 Logistic regression

Logistic regression [2, 116] is a way to build a binary classifier for points in $\mathbb{R}^n$. Instances have n numeric features, and they can belong to two classes: 0, the *negative* class, or 1, the *positive* class. Throughout this dissertation, I will use the convention that real instances are *negative* and fake instances are *positive*.

Although logistic regression only works for numeric features, there is a standard approach to convert categorical features into numeric features [116]: for each categorical variable

[1] Logistic regression can support non-linear decision boundaries if polynomial features are introduced, but this is not efficient for a large number of features.

[2] Specifically, I used Weka, version 3.6, downloaded in April 2012 from `http://prdownloads.sourceforge.net/weka/weka-3-6-7.zip`.

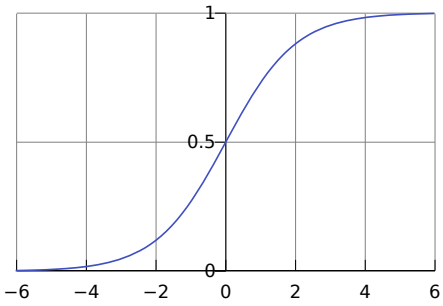


Figure 3.1: The logistic function $f(z) = 1/(1+e^{-z})$.

j that can take on values from the set $X_j = \{x_1, x_2, \ldots x_k\}$, create $k-1$ new variables $j_1, j_2, \ldots j_{k-1}$, where the variable j_ℓ is equal to 1 if j has the value x_ℓ, and 0 otherwise.

The *logistic model* consists of a real-valued $(n+1)$-dimensional vector $(\theta_0, \theta_1, \ldots \theta_n)$. These parameters define a hyperplane, called the *decision boundary*, given by the equation $\theta_0 + \theta_1 y_1 + \cdots \theta_n y_n = 0$. To use this model to predict whether a given instance $y = (y_1, y_2, \ldots, y_n)$ is positive, first the quantity $z = \theta_0 + \theta_1 y_1 + \cdots + \theta_n y_n$ is computed. The sign of z gives the side of the separating hyperplane to which y lies, and the magnitude of z is proportional to its distance from this hyperplane. To convert z, which can take on any value in the range $(-\infty, \infty)$, to a probability $h_\theta(y)$, the *logistic function* is applied to z. This function, defined by $f(z) = 1/(1+e^{-z})$, is plotted in Figure 3.1. Thus, the final prediction of logistic regression, the probability that an instance is positive, is given by

$$h_\theta(y) = f(z) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 y_1 + \cdots + \theta_n y_n)}} \quad .$$

Various optimization algorithms can be used to fit the logistic model to a set of labeled training data. The most straightforward is *gradient ascent to maximize the likelihood* (see, e.g., Section 4.4.1 of [54]). Weka uses a different technique called the *ridge estimator* [74] that is optimized to work well when there are a large number of features. This technique is probably more advanced than necessary for the size of my data sets, but I chose to use Weka's default implementation for simplicity. In any case, the particular optimization algorithm used to fit the model is a relatively minor point, and it is not necessary to understand this algorithm in detail to use logistic regression.

As shown in Table 3.3, logistic regression is a very fast algorithm. It also scales well to larger data sets than the ones that I use in this dissertation—especially if approximate

optimization techniques are used, such as *stochastic gradient ascent* [19].

3.2.2 Random forest

Random forest, introduced in 2001 [23], is a much newer technique than logistic regression. In a comparison to many other popular classification algorithms on multiple benchmark data sets, it was shown to have the best overall accuracy [26]. It is an *ensemble* technique, meaning that it aggregates the predictions made by many sub-models to make a prediction. In the case of random forest, the basic model is a *decision tree*.

In a decision tree, each internal node corresponds to a feature j. Each child of that node corresponds to one of the values of j if j is categorical, or a range of values of j if j is numeric or ordinal. Each leaf node contains the subset of instances in the training set that have feature values specified by the path from the root to the leaf. To make a prediction on a new instance y, the leaf that y would map to is found and a majority vote is taken of the training instances' labels at that leaf.

There are many algorithms to build decision trees, but a particularly simple randomized algorithm is used in the case of random forest. First, a parameter K is chosen that is less than or equal to the total number of features. Each tree is built by recursively splitting the set of instances according to a single feature. To choose a feature on which to split the node, first K features are chosen uniformly at random. Of these K features, the feature that would result in the highest *information gain* after splitting is chosen. (For numeric features, all possible binary splits are considered.) The information gain of splitting on a feature j is defined to be the entropy of the class distribution of the instances at the node minus the average entropy of the class distributions of the child nodes after the split on j. This process is repeated recursively until each leaf node consists only of instances having the same label.

A random forest consists of some number, say I, of these random decision trees. To predict a class for an instance, a majority vote is taken across all I of the component decision trees. (To predict a probability, the fraction of trees voting *positive* is measured.) To build the decision trees, a technique called *bagging* [22] is used. In this technique, each tree is built on a random sample drawn uniformly at random with replacement from the training set. (This is called a *bootstrap* sample.) The size of the training sets are equal in size to the training set, but because they are drawn with replacement, they are somewhat different, with some instances appearing more than once and some instances not appearing at all.

As shown in Table 3.3, random forest is also a fast algorithm, although—at least in Weka's implementation—not quite as fast as logistic regression. It also scales well. Each tree can be

built completely independently, so it is especially amenable to parallel implementations.

3.2.3 Model selection

Model selection is the process of choosing, based on a set of data, the type of model on which to train a classifier. This term can mean different things for different classification techniques. For logistic regression, I use it to mean *feature selection*: choosing the subset of features on which to train the model. If too many features are used to train the model, then it may *over-fit* the training data, and if too few features are used, then it may *under-fit* the training data. In both cases, the model will not generalize well to new data sets; therefore it is important to perform feature selection carefully.[3] For random forest, I use the term model selection to mean the process of choosing the parameters K, the number of features considered at each split, and I, the number of decision trees in the forest.

To perform model selection for logistic regression in this dissertation work, I used a *bottom-up greedy* feature selection algorithm [116]. This algorithm works as follows. Out of all possible features, it chooses the feature that results in the highest accuracy for logistic regression evaluated using 10-fold cross-validation on the training set, averaged over some number of repetitions N. (See Section 3.4 for definitions of these terms.) It adds this feature to the set of features used by the classifier. Denote this set of features by $\mathcal{F}$. Then, out of the remaining features, it finds which feature, when added to $\mathcal{F}$, results in the highest cross-validated accuracy. It continues this process until the accuracy of the classifier cannot be improved by adding any of the remaining features to $\mathcal{F}$.[4]

To perform model selection for the random forest, I first chose a value of I, the number of trees. Prediction accuracy usually increases monotonically with the value of I [23], so my approach in making this choice was to perform some initial experiments to find the lowest value of I for which the accuracy achieved a maximum value. After choosing I, I evaluated a large set of values of K, typically powers of 2 between 2 and the number of features in the data set. I chose the value of K that resulted in the highest accuracy evaluated using 10-fold cross-validation on the training set.

---

[3] There are a number of other techniques to control this tradeoff, sometimes called the *bias–variance tradeoff*, including *regularization* [54], but in this dissertation I keep the discussion simple by focusing only on feature selection.

[4] Actually, to avoid over-fitting, it only adds a new feature if doing so increases the accuracy by at least a certain threshold amount, which I chose to be 0.004 after some experimentation.

3.3 Unsupervised anomaly detection algorithms

Supervised classification algorithms are often very powerful prediction algorithms. However, they require a set of high-quality labeled training data to work. Obtaining such data may sometimes be prohibitively expensive or time consuming. In such cases, unsupervised anomaly detection algorithms may be a viable alternative.

Anomaly detection [27, 57, 91, 94] has been defined as "finding patterns in data that do not conform to expected behavior" [27]. Other terms for it include outlier detection, novelty detection, noise detection, deviation detection, and exception mining [57]. Anomaly detection algorithms—which may be nearest neighbor-based, clustering-based, statistical, information theoretic, or spectral [27]—have been applied to detect a wide variety of surprising occurrences—including network intrusion, credit card fraud, insurance fraud, insider trading, changes in medical condition, unusual images, and ecosystem disturbances [27, 57].

Most anomaly detection techniques assign an *outlier score* to each instance, which indicates how anomalous the instance is. To actually choose a set of anomalies based on these scores, several techniques can be used, including finding the top k outlier scores for some predetermined k, finding all instances that have an outlier score greater than some predetermined cutoff, and finding all instances with an outlier score that differs significantly from a typical outlier score.

Aspects of anomaly detection have appeared before in survey data quality research. Standard practice for cleaning survey data is to look for statistical outliers in numeric data [55, 114]. Using my framework, this approach could be classified as a `UFU` algorithm. (Recall that because this is already a widely studied technique, I do not investigate it further in this dissertation.)

Although I have framed anomaly detection primarily as an alternative to classification when no training data is available, it may make sense to use it even when training data is available. Anomalies are often of interest to data users regardless of whether they were caused by fabrication because they might indicate important patterns in the data like disease outbreaks.

3.3.1 Form-level algorithm (LOCI)

The *Local Correlation Integral (LOCI)* algorithm [87] is a `UFM` algorithm that I used in my work to detect curbstoning. Out of the wide variety of anomaly detection algorithms that could be employed at the form level (see, e.g. [25, 70, 120]), I chose this algorithm primarily for its simplicity and generalizability.

To find anomalous forms, the LOCI algorithm requires there to be a distance function $d(\cdot, \cdot)$ defined between each pair of forms. This function $d(\cdot, \cdot)$ must work for the heterogenous feature spaces that are typical of survey data. I chose to use the *Heterogeneous Euclidean-Overlap Metric (HEOM)* [115], which is is a straightforward way to combine the Euclidean distance for numeric attributes [116] with the *overlap metric* for categorical attributes [106].

Consider two forms y and z, and let Q be the set of features in the survey data. For each feature $j \in Q$, let the value of j for forms y and z be denoted by y_j and z_j. For numeric features j, let $\max_j$ be the maximum value of feature j over all forms, and let $\min_j$ be the minimum. According to HEOM, the distance between y and z is

$$d(y, z) = \sqrt{\left(\sum_{j \in Q} d_j(y_j, z_j) \right)^2} \ ,$$

where

$$d_j(y_j, z_j) = \frac{y_j - z_j}{\max_j - \min_j}$$

if j is numeric, and

$$d_j(y_j, z_j) = \begin{cases} 1 & \text{if } y_j \neq z_j \\ 0 & \text{otherwise} \end{cases}$$

if j is categorical.[5] Although straightforward, the HEOM tends to perform only slightly worse than more sophisticated distance functions [18, 106].

Given the HEOM distances, the LOCI algorithm decides whether a form y is an outlier by comparing the density of forms around y to the mean density of its neighbors. If this quantity is small, then y is given a high outlier score. More precisely, for any positive real number r, let the density of y, denoted $n(y, r)$, be the number of forms within a distance r of y according to the HEOM distance $d(\cdot, \cdot)$. Let $\hat{n}(y, r)$ be the mean of all $n(z, r)$ values for each form z within a distance of r of y (including y itself), and let $\sigma(y, r)$ be the standard deviation of these values. For a range $[r_{min}, r_{max}]$, the outlier score of y is given by

$$\gamma = \max_{r_{min} \leq r \leq r_{max}} \frac{\hat{n}(y, r) - n(y, r)}{\sigma(p_i, r)} \ .$$

The higher γ is, the more of an outlier y is. As suggested by Papadimitriou et al. [87], my implementation of this algorithm sets r_{min} to be the smallest radius that includes at least 20

[5] Note that the dependence of $d_j(y_j, z_j)$ on $\max_j$ and $\min_j$ is sensitive to outliers. A direction for future work is to investigate distance metrics that are more robust to outliers, such as one formed by replacing the denominator with the *inter-quartile range (IQR)*.

points and r_{max} to be the diameter of the entire set of points, the maximum value possible.

As shown in Table 3.3, LOCI is by far the slowest algorithm described in this chapter. It is probably unrealistic to run it on data sets containing more than a few thousand forms. However, there is an approximate version of the algorithm that scales better [87]. Exploring the potential of this approximate algorithm to detect fabrication on larger data sets is an interesting direction for future work.

3.3.2 Interviewer-level algorithms (MMA and SVA)

Existing anomaly detection techniques do not apply as readily to interviewer-level curbstoning algorithms because of the additional level of complexity involved in aggregating the forms by interviewer. In this section, I describe two novel unsupervised interviewer-level algorithms that apply to categorical response data. Both are univariate but can be easily extended to be multivariate. (Thus, in this section, I cover both `UIU` and `UIM` algorithms.)

I start with some notation that will be used to define both techniques. Consider an interviewer i and a question $j \in Q$ with a finite, discrete set of answer values. Let X_j be the range of values that question j can take. For any $x \in X_j$, let $g_j^i(x)$ be the number of times that interviewer i reports the value x for question j. For all i, j, and $x \in X_j$, let

$$f_j^i(x) = \frac{g_j^i(x)}{\sum_{x \in X_j} g_j^i(x)}$$

be the fraction of times that interviewer i reports the value x for question j.

Multinomial Model Algorithm (MMA)

The *Multinomial Model Algorithm (MMA)* computes an outlier score for interviewer i and question j by computing the chance of seeing g_j^i if the answers are chosen according to a multinomial probability distribution with parameters estimated from the other interviewers. The lower this probability is, the the higher the outlier score is.

Specifically, MMA computes a score m_j^i for each interviewer and question as follows. For all $x \in X_j$, let

$$g_j^{-i}(x) = \sum_{i' \neq i} g_j^{i'}(x)$$

be the frequency count for value x over all interviewers besides i. Let

$$f_j^{-i}(x) = \frac{g_j^{-i}(x)}{\sum_{x' \in X_j} g_j^{-i}(x')}$$

be the normalized version of these counts. From this distribution, MMA calculates the *expected* count for value x from interviewer i to be

$$E_j^i(x) = f_j^{-i}(x) \sum_{x' \in X_j} g_j^i(x') \ .$$

These expected counts can form the basis for a χ^2-test giving a p-value for the chance of seeing frequency counts that deviate at least as much as g_j^i, given the null hypothesis that the counts are drawn from a multinomial distribution with parameters given by f_j^{-i}. That is, MMA calculates the statistic

$$R_j^i = \sum_{x \in X_j} \frac{(g_j^i(x) - E_j^i(x))^2}{E_j^i(x)}$$

and determines the probability of seeing a value at least this large in a χ^2 distribution with $|X_j| - 1$ degrees of freedom. A smaller value of this probability indicates a greater amount of surprise for interviewer i's distribution. The final score, m_j^i, is the negative logarithm of this value. That is,

$$m_j^i = -\log \Pr\left[\chi^2_{|X_j|-1} \geq R_j^i\right] \ ,$$

where a higher value indicates a greater amount of surprise. MMA can be made to be multivariate by reporting the single value m^i, the mean of the m_j^i over all categorical questions j.

As shown in Table 3.3, MMA is a fast algorithm that easily scales to data sets of hundreds of interviewers and thousands of forms.

s-Value Algorithm (SVA)

MMA forms an expected distribution for an interviewer by taking the weighted mean of the distributions for all other interviewers. Using the mean as an estimator is sensitive to outliers: if there is a large amount of curbstoning, the expected distribution might not be representative of the true population distribution. This possibility inspired my second unsupervised interviewer-level algorithm, The *S-Value Algorithm (SVA)*, which uses a median instead of a mean because of its decreased sensitivity to outliers [60].

For any vector v, define $\mu_{1/2}(v)$ to be the median of the values in v. For a question j and a value $x \in X_j$, let $f_j(x)$ be the vector of values $f_j^i(x)$ for all interviewers i. The surprise for interviewer i's distribution for question j can be characterized by

$$d_j^i = \sum_{x \in X_j} \left| f_j^i(x) - \mu_{1/2}(f_j(x)) \right| \quad .$$

One could directly use the d_j^i value as the final outlier score for interviewer i and question j, but for some questions, this value could be high for all interviewers. This could happen if, for example, the question's answer depends strongly on geography and interviewers are widely dispersed. For this reason, SVA normalizes the d_j^i values by their median. More precisely, let d_j be the vector of d_j^i values over all i, and let

$$s_j^i = \frac{d_j^i}{\mu_{1/2}(d_j)} \quad .$$

This value, named the *s-value* to indicate surprise, is the final outlier score given by SVA. A higher value indicates a greater amount of surprise. As for the MMA method, SVA can be made to be multivariate by reporting the single value s^i, the mean of the s_j^i over all categorical questions j.

As a side note, SVA is methodologically similar to the recently developed *Tariff Method* of verbal autopsy [63]. Here the goals are reversed, however. Instead of starting with signs and symptoms and hoping that the cause of death can be identified, SVA starts with the survey responses and hopes that it *cannot* identify the interviewer.

As shown in Table 3.3, SVA is even faster than MMA. Like MMA it should scale easily to data sets of hundreds of interviewers and thousands of forms.

3.4 Evaluation methodology

Much of the remainder of this dissertation could be characterized as evaluating how well the algorithms defined in this chapter detect curbstoning. In this section, I overview the methodology I use to perform these evaluations.

3.4.1 Measuring performance

Each of the algorithms that I have described predicts whether each of some *unit* of the survey data is fabricated. For form-level multivariate algorithms, the unit of prediction is the form; for interviewer-level univariate algorithms, the unit of prediction is the interviewer–

Measure	Description
Accuracy	The fraction of instances that are predicted correctly.
True positive rate	The fraction of positive instances that are correctly predicted. A high number is desired.
False positive rate	The fraction of negative instances that are incorrectly predicted. A low number is desired.
Sensitivity	Another name for the true positive rate.
Specificity	Equal to $1 - (\text{false positive rate})$.
Precision	The fraction of instances that are predicted to be positive that are actually positive. A high number is desired.
Recall	Another name for the true positive rate.
F_1-score	Equal to $2 \cdot (\text{precision} \cdot \text{recall}) / (\text{precision} + \text{recall})$. This is a commonly used measure in information retrieval that combines both precision and recall [98]. It may be more meaningful than the accuracy for data sets with a highly skewed label distribution. A high number is desired.

Table 3.4: Measures of classification performance. All numbers range between 0 and 1.

question pair; and for interviewer-level multivariate algorithms, the unit is the set of all of the forms from an interviewer. All of the algorithms that I have defined can express their predictions as a real-valued *score*, where a higher value indicates a stronger belief that the unit is positive (fabricated). For logistic regression, this number is the probability output by the logistic function; for random forest, it is the fraction of trees that predict *positive*; and for the unsupervised methods, it is the outlier score. In order to translate this score into a binary prediction, a *cutoff score* must be chosen. For algorithms in which the score is a probability, a natural cutoff is 0.5, although other cutoffs might make sense in certain situations. As described in Section 3.3, there are many ways to choose a cutoff score for the unsupervised methods.

Once a cutoff score is chosen, there are various ways of measuring the accuracy of the algorithm's predictions. In this dissertation, I use eight commonly used and related measures: *accuracy*, *true positive rate*, *false positive rate*, *sensitivity*, *specificity*, *precision*, *recall*, and *F_1-score*. These measures are defined in Table 3.4. The value of these measures depends on the cutoff score chosen. As the cutoff score increases, the specificity increases; the true positive rate, false positive rate, sensitivity, and recall decrease; and the accuracy, precision, and F_1-score neither increase nor decrease in general.

To explore the tradeoff between sensitivity and specificity as the cutoff score varies, I often employ a graphical technique called Receiver Operating Characteristic (ROC) curves [47] (see, e.g., Figure 4.1). These curves plot the true positive rate versus the false positive rate for all possible cutoffs. A point on the diagonal line $y = x$ is what one would expect if the classifier was guessing randomly with a fixed probability of outputting positive. If a classifier has an ROC curve that is substantially above this line, then this is evidence that it is making predictions that are more accurate than would be expected by chance. The more an ROC curve is above this line, the better the classifier's predictive ability is.

3.4.2 Training vs. test data

As mentioned in Section 3.2.3, supervised classification may suffer from *over-fitting*, which occurs when the classifier is fit so closely to the training data that it does not generalize well to new data. To avoid reporting performance numbers that are artificially high, common practice is to divide labeled data into a *training* and *test* set [116]. The model is fit to the training data, but the measures of performance that are reported are for the test set.

Sometimes, it is not possible to strictly separate a test set from the training data. In this case, an approximation of using a test set, called *10-fold cross-validation*,[6] can be used [116]. In this technique, the data is randomly partitioned into 10 equal-sized pieces. Nine of the pieces are used to train a classifier, and its performance is measured on the tenth piece. This process is repeated 10 times, once for each possible test set. In this way, the performance of the classifier is measured for each instance of the data, but no instance was used to train the classifier on which it was tested.

In this dissertation, I use cross-validation in Chapter 4 and separate training and test sets in Chapters 5 and 6. As discussed in Section 3.2.3, I also use cross-validation on the training set (or the entire data set, if there is no segregated training data) for model selection.

3.5 Chapter summary

In this chapter, I described five curbstoning detection algorithms that I evaluate later in this thesis: logistic regression, random forest, LOCI, MMA, and SVA. I showed how these algorithms can be placed in a framework that helps to articulate the tradeoffs to consider when choosing an algorithm to detect curbstoning. The algorithms MMA and SVA are novel; they are a direct contribution of my dissertation. The other algorithms are not novel, but they were drawn from the state of the art in supervised classification and unsupervised

[6] There is nothing particularly special about the number 10, but using it seems to be standard practice.

anomaly detection. I concluded the section by describing the evaluation methodology I use in this dissertation.

In the next section, I perform my first evaluation of these algorithms. I show that, in data from two East African health organizations, they are able to detect curbstoning with high accuracy.

Chapter 4
A PRELIMINARY STUDY IN EAST AFRICA

In this chapter, I begin to evaluate how well the algorithms described in Chapter 3 can detect curbstoning. To do so, I run them on three data sets collected from two community health programs in East Africa. I start, in Section 4.1, by explaining what community health programs are and by motivating the need that they have for effective data quality control. Then, in Section 4.2, I give more information on the data sets that I use in this chapter.

One of the three data sets is *labeled*. That is, it is annotated to indicate which part of it is real and which part of it is fabricated. To get these labels, I combined data that was known to be real with data that some interviewers fabricated during a "fake data party" (see Section 4.2.1 for details). These labels allowed me to measure the performance of both the supervised and unsupervised curbstoning detection algorithms quantitatively. I report the results of this evaluation in Section 4.3. I show that the supervised algorithms achieved an accuracy of up to 81% at the form-level and up to 97% at the interviewer level. I also show by using ROC curves that both the univariate and multivariate unsupervised algorithms identified curbstoning at a level substantially above random chance.

The other two data sets are unlabeled; they do not provide an opportunity to evaluate the performance of the algorithms quantitatively. However, these data sets are more closely tied to real field operations, so, in a sense, they are more authentic than the labeled data. In Section 4.4, I conclude the chapter by summarizing the results of running one of the unsupervised anomaly detection algorithms on these unlabeled data sets. I show that it found compelling anomalies in these data sets. Although it is impossible to know what caused these anomalies, their discovery points to the usefulness of the unsupervised algorithms.

The work that I present in this chapter is based on work that led to an earlier publication [14].

4.1 Background on community health worker programs

In a *Community Health Worker (CHW)* program, trained members of the community—who are not healthcare professionals—travel between homes in a given area to check up on

clients, perform routine care, promote healthy behavior, and make referrals to health facilities when necessary. Researchers have shown that CHW programs can positively affect health outcomes but that they require strong supervision and support to be effective [75]. To address some of these needs, researchers and developers have created software tools that run on mobile phones carried by CHWs [41, 44, 67, 79, 121]. CommCare [41, 79], for example, is an open-source software platform for CHWs that is used to register new clients, structure interviews, and record responses directly into the phone. The mobile-phone component of the software uploads its data to a central server called CommCareHQ [34]. Supervisors can use CommCareHQ to track cases and generate summary reports of their organization's operations.

CommCare removes the need for CHWs to manually enter data into a database after recording it on paper. This change in workflow can increase workers' efficiency [79]. It provides other advantages as well. First, it can make the data more quickly available to doctors and epidemiologists, who can use it for health surveillance. Second, it can make it easier for supervisors to manage their operations effectively. For example, a supervisor can use CommCareHQ to monitor the number of visits that CHWs are making to ensure that program objectives are being met. Or, automated SMS reminder systems built on top of CommCareHQ can be used to encourage CHWs to follow up with clients when needed. In fact, it was recently shown that one such system decreased the average number of days a CHW's client were overdue for a follow up visit by 86% [42].

However, the data management capabilities of CommCare are only useful if the software is managing high-quality data—and this might not always be the case. At one CommCare deployment at an organization that I will call *Organization A*, surprising patterns in the data were discovered. Further investigation revealed that a significant number of CHWs were curbstoning. At a different program at an organization that I will call *Organization B*, I found that supervisors were performing extensive data quality control measures that exhibited the degree to which they were concerned about curbstoning and other data quality problems.

Curbstoning in a system like CommCare negates the intended benefit of providing decision makers with a sound basis for their decisions. Even more important to those in community health, it hinders the ability of supervisors to manage their operations effectively, and without effective supervision, the health of the client population may suffer. Supervisors already spend a portion of their limited time addressing data quality issues; automatically detecting data quality problems like curbstoning is of primary importance.

4.2 Description of data sets used

There are three data sets in this chapter. The first, which I denote by $\mathcal{A}_\ell$, is a labeled set from the CHW program called *Organization A* in the previous section. This organization was operating in Tanzania until it was shut down due to a lack of funding in 2011. The second, which I denote by $\mathcal{A}_u$, is a large unlabeled data set from the same organization. The third, which I denote $\mathcal{B}$, is an unlabeled set from the CHW program called *Organization B* in the previous section. This organization is in Uganda, and as of June 2012, it is still operating.

4.2.1 Data set $\mathcal{A}_\ell$

Data set $\mathcal{A}_\ell$ is from Organization A, a CommCare-supported CHW program in Tanzania that specialized in maternal and neonatal health. The workers in this organization travelled between approximately 100 households in the period of a month, asking basic health and demographic questions, such as whether anyone had a cough, what kind of family planning methods were being used, and whether there were any children under 2 years old. If a CHW found an urgent health problem, she would refer the household to a clinic.

The data consists of 846 forms filled out by 40 CHWs. There were 12 questions in total, of which one was numerical ("How many people are in your household?") and 11 were categorical. Apart from this response data, CommCare recorded one piece of metadata, how long it took to complete each form. Of the 846 rows in $\mathcal{A}_\ell$, 317 were labeled `real` (from 15 unique CHWs) and 529 were labeled `fake` (from 25 unique CHWs).[1]

The `real` part of $\mathcal{A}_\ell$ came from a study measuring how often different interviewers reported the same answers on surveys of the same household [40]. Unlike normal day-to-day operations at Organization A, these interviews were conducted in a central location to which mothers from households in the area would travel. The interviews were conducted on seven different days in November and December of 2010. Because of the unusually high amount of supervision during these interviews, I have high confidence that the interviewers were faithfully interviewing the clients and recording their responses accurately.

The `fake` part of $\mathcal{A}_\ell$ was obtained from a "fake data party" that my collaborators and I set up for the purpose of this study. Twenty-five CHWs from Organization A were gathered and given a small payment to fill out several copies of the same form used for the real data. They did not actually interview any households while filling out these forms. Instead, they

[1] There was some overlap between the CHWs filling out the `real` and `fake` data, but the system of recording user identifiers was different for each of the labels, so I do not have a way to match the CHWs between labels. Thus, for simplicity, I am treating them as distinct groups of CHWs.

were instructed in Swahili to "pretend they were a client and answer as a client would." Over two and a half hours, these 25 CHWs faked a total of 529 forms. A brief manual inspection of this data revealed no obvious misunderstandings and no obvious shortcuts taken by the CHWs to fill out the forms.

It is important to note that the labels of $\mathcal{A}_\ell$ are not perfect. It is impossible to be certain that every form in the real data set is accurate, and a CHW who is fabricating data in a fake data party might do so differently than a CHW who is fabricating data in the field. I believe, however, that the labels are accurate enough and the data realistic enough to provide a meaningful first evaluation of the algorithms described in Chapter 3. Furthermore, some of the concerns regarding the artificial nature of the fake data party are addressed in Chapters 5 and 6.

4.2.2 Data set $\mathcal{A}_u$

Data set $\mathcal{A}_u$ is an unlabeled data set from Organization A. This data set is more realistic than the labeled data set $\mathcal{A}_\ell$ because it comes from actual CHW field operations during June and July of 2011. The form used is similar to the form in the labeled set. It has 15 questions, all of which are categorical. I filtered the data to include rows only from CHWs who submitted at least 20 forms during the two-month period. This resulted in a set of 4321 forms submitted by 33 unique CHWs.

4.2.3 Data set $\mathcal{B}$

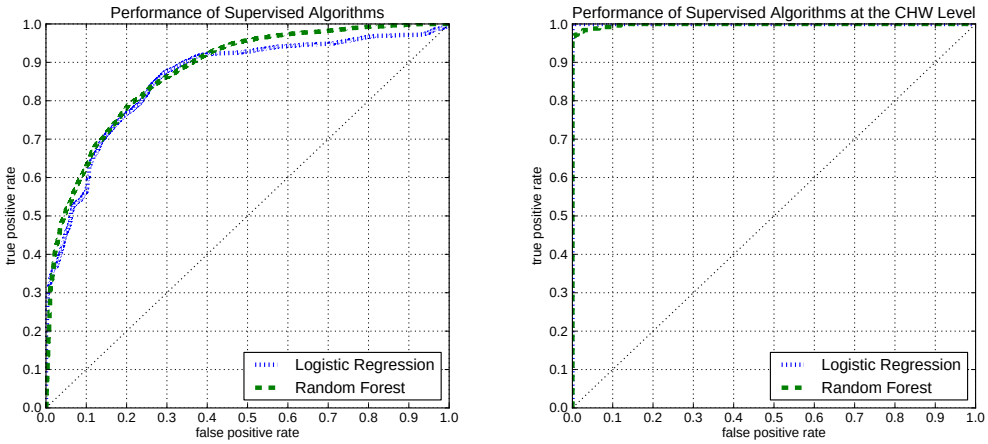
Data set $\mathcal{B}$ comes from the field operations of Organization B, a CHW program in Uganda. A number of questions in this data set were "check all that apply." To transform these into a multiple choice format, I created a separate `yes/no` question for each possibility. This resulted in 103 different questions. The raw data set consists of 328 rows submitted by 42 interviewers. I filtered this to include rows only from interviewers who submitted at least 8 forms, which resulted in a final set of 265 forms submitted by 26 interviewers.

4.3 Evaluation on labeled data

Because the data set $\mathcal{A}_\ell$ is labeled, it can be used to evaluate the performance of both supervised and unsupervised algorithms. In Section 4.3.1, I use it to evaluate the supervised classification algorithms logistic regression and random forest. In Section 4.3.2, I use it to evaluate the unsupervised anomaly detection algorithms LOCI, MMA, and SVA.

Algorithm	Prec.	Rec.	Spec.	F_1	**Acc.**
Logistic Regression	0.84	0.86	0.72	0.85	**0.81**
Random Forest	0.84	0.85	0.72	0.84	**0.81**

Table 4.1: Form-level performance of supervised algorithms on $\mathcal{A}_\ell$. This table shows the precision, recall, specificity, F_1-score, and accuracy for both logistic regression and random forest.



(a) Form-level (b) Interviewer-level

Figure 4.1: ROC curves for supervised algorithms on $\mathcal{A}_\ell$.

4.3.1 Supervised algorithms

To optimize the performance of the supervised algorithms, I first performed model selection, as outlined in Section 3.2.3. For logistic regression, I performed bottom-up greedy feature selection with cross-validated accuracy (averaged over 40 repetitions) as the optimization criterion. This resulted in 4 of the 13 features being chosen. In order, they were `total-time` (the last timestamp on the form minus the first timestamp on the form), `diarrhea` ("Does anyone in the house have diarrhea?"), `death` ("Has anyone in the house died?"), and `new-born` ("Is there a new born in the house?"). Of these features, `total-time` was negatively correlated with fabrication, and `diarrhea`, `death`, and `new-born` were positively correlated with fabrication. Thus, *curbstoners completed forms more quickly and tended to overestimate the prevalence of rare events in the households they interviewed*.

The fact that interviewers fabricated data more quickly (a mean of 148 seconds for the

Algorithm	Prec.	Rec.	Spec.	F_1	**Acc.**
Logistic Regression	0.88	1.00	0.77	0.94	**0.92**
Random Forest	1.00	0.96	1.00	0.98	**0.97**

Table 4.2: Interviewer-level performance of supervised algorithms on $\mathcal{A}_\ell$. This table shows the precision, recall, specificity, F_1-score, and accuracy for both logistic regression and random forest.

fabricated data versus a mean of 240 seconds for the real data) is consistent with the *fast interviews* hypothesis from related work [24, 69, 76, 82], and the fact that they misestimated the population distribution of responses is consistent with the *unusual data* hypothesis from related work [61, 82, 109]. (Both of these hypotheses were described in Section 2.1.)

For random forest model selection, I set $I = 32$ and tried all values of K in $\{2, 4, 8, 13\}$, using the cross-validated accuracy (averaged over 10 repetitions) as an optimization criterion. This resulted in a choice of $K = 8$.

Table 4.1 shows performance measures for both algorithms obtained via 10-fold cross-validation. Both algorithms have similarly good performance, achieving an accuracy of 81%. Figure 4.1(a) shows ROC curves for both of these algorithms. These curves show a good—and very similar—set of sensitivity-specificity tradeoffs for both algorithms. Thus, the supervised classification algorithms can accurately identify curbstoning at the form level.

One could also ask how well the algorithms work at the interviewer level. That is, if one aggregates the predictions for each form by interviewer, does this accurately predict which of the interviewers are fabricating data? To answer this question, I performed the following experiment for both logistic regression and random forests. Using 10-fold cross-validation, I obtained a probability for each form that it was fabricated. Then, for each interviewer i, I averaged these probabilities over all of i's forms. I interpreted this number to be the probability that i was fabricating data. If it was greater than 0.5, the prediction was that i was a *positive* (fake) example, and if it was less than 0.5, the prediction was that i was a *negative* (real) example. I repeated this 10 times to get 10 predictions for each interviewer, or 400 predictions total.

Table 4.2 shows the performance measures from these predictions. It shows that aggregating information by interviewer is very effective. The accuracy of the predictions made this way is 92% for logistic regression and 97% for random forest. Figure 4.1(b) shows ROC

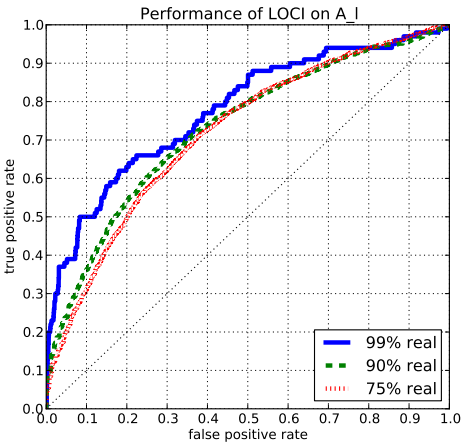


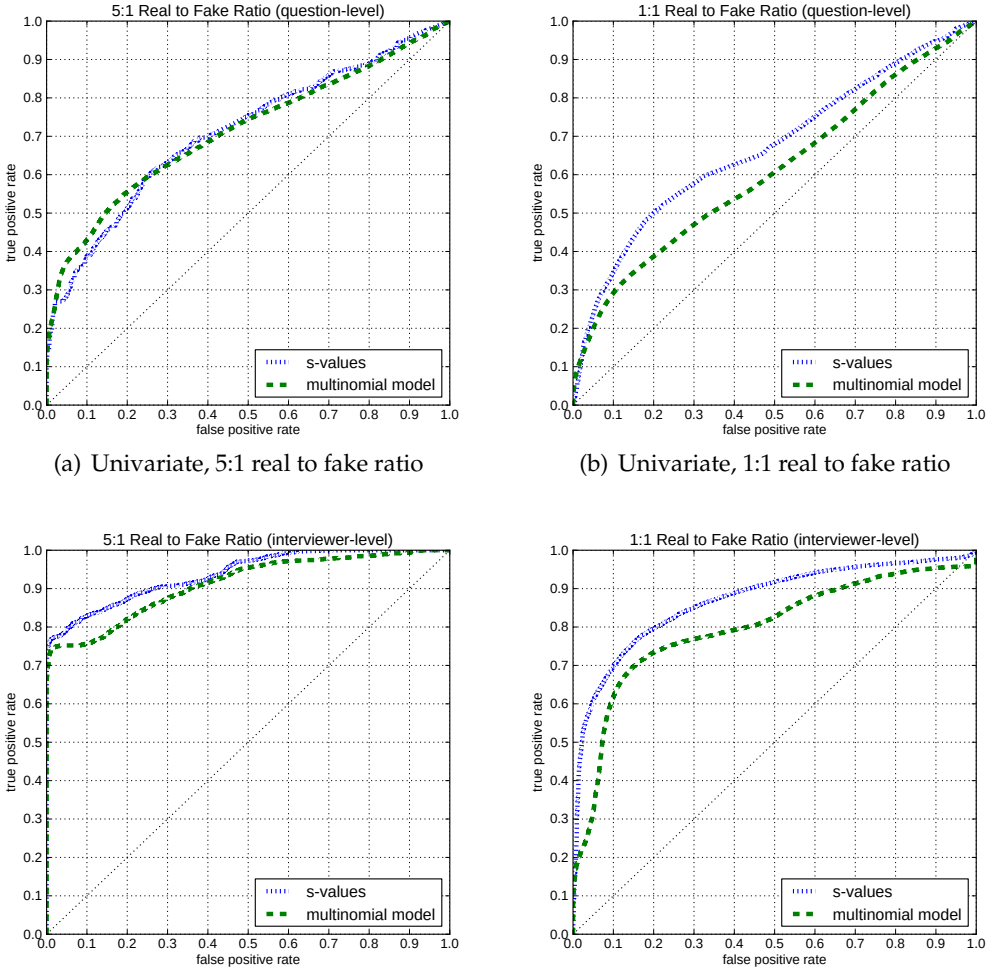
Figure 4.2: Performance of LOCI on $\mathcal{A}_\ell$.

curves for these predictions. These ROC curves are extremely good, and in fact, the curve for logistic regression is *perfect*, meaning that there is a number z such that all of the interviewers who fabricated data had a score greater than z and all of the interviewers who did not fabricate data had a score less than z.[2] Thus, when making predictions at the interviewer level, the supervised algorithms are even more effective at identifying curbstoning. Although in practice it is unlikely that interviewers will fabricate either all or none of their data, this analysis suggests that interviewers who habitually fabricate a lot of data could be identified by unusually high average scores from the supervised classification algorithms.

4.3.2 Unsupervised algorithms

A complication of evaluating an anomaly detection algorithm on $\mathcal{A}_\ell$ is that the algorithm should perform better when there is more real data than fake data, but the data set $\mathcal{A}_\ell$ has a fixed fraction of fabricated data (63%). To work around this, I used random subsets of $\mathcal{A}_\ell$ with smaller—and varying—amounts of fabricated data. For LOCI, I ran the following experiment once for each value of $p \in \{75, 90, 99\}$: for 100 repetitions, I randomly selected p real forms and $100 - p$ fake forms. Then I had LOCI compute outlier scores for each of the chosen forms. I used these scores, aggregated over the 100 repetitions, to create ROC curves.

[2] It turns out that this number z could be anything between 0.52 and 0.54. This explains the seeming discrepancy between the performance of logistic regression in Table 4.2 and Figure 4.1(b): Table 4.2 reports the performance when a cutoff score of 0.5 is used, which is below the minimum value of z needed for perfect separation.



(a) Univariate, 5:1 real to fake ratio

(b) Univariate, 1:1 real to fake ratio

(c) Multivariate, 5:1 real to fake ratio

(d) Multivariate, 1:1 real to fake ratio

Figure 4.3: ROC curves for the unsupervised anomaly detection algorithms MMA and SVA on $\mathcal{A}_\ell$.

Figure 4.2 shows the ROC curves generated in this way. It shows that the algorithm tended to do fairly well at predicting fabricated data, though, as a comparison to Figure 4.1(a) shows, not as well as the supervised algorithms, even when 99% of the data was real. As one would expect, the performance of LOCI decreases when there is more fabricated data. However, the difference in performance between the 90% real and 75% real curves is small, suggesting that this effect is minor for moderate rates of fabrication.

Next, I evaluated both the univariate and multivariate versions of MMA and SVA, starting

with the univariate versions. Just as with LOCI, I used random subsets of $\mathcal{A}_\ell$ with varying amounts of fabrication. Since MMA and SVA make predictions at the interviewer level, I chose subsets of *interviewers* instead of forms. There were two conditions: a 5:1 ratio of real to fake data and a 1:1 ratio of real to fake data. Specifically, for each value of $p \in \{3, 9\}$, for 1000 repetitions, I chose p CHWs who were fabricating data and $18 - p$ CHWs who were not fabricating data. I formed a data set by choosing all of the forms made by these 18 CHWs. Then I had SVA and MMA compute outlier scores for each interviewer-question pair. I used these scores, aggregated over the 1000 repetitions, to create ROC curves.

Figures 4.3(a) and 4.3(b) show the ROC curves generated in this way. These curves show a high level of performance, similar to that of LOCI at the form level. Even though MMA and SVA are interviewer-level, it is perhaps not surprising that their univariate versions have a performance similar to the form-level algorithm LOCI because LOCI can leverage the information across all questions, whereas univariate MMA and SVA can use only a single question.

As expected, the performance of MMA and SVA is better when the proportion of fake data is lower, but—perhaps surprisingly—*even when the amount of fake data is the same as the amount of real data, the performance of these algorithms is substantially above chance.* I speculate that this is because in aggregate, the CHWs come up with a fairly accurate estimate of the population averages, but the fake CHWs tend to deviate more from this estimate than the real CHWs. Also, note that MMA and SVA have similar performance at the 5:1 real to fake ratio, but SVA has a higher performance than MMA at the 1:1 real to fake ratio. This could be because the use of medians by SVA to determine the expected distributions is less sensitive to the extra noise introduced by the large number of CHWs fabricating data.

Figures 4.3(c) and 4.3(d) show the ROC curves generated by the multivariate versions of MMA and SVA for the same real to fake ratios. These figures show that, not surprisingly, both algorithms perform better in the multivariate case. SVA tends to do slightly better than MMA, a difference that is more pronounced for the 1:1 real to fake case.

4.4 Evaluation on unlabeled data

Although the labels in data set $\mathcal{A}_\ell$ provide a ground truth with which to evaluate the performance of curbstoning detection algorithms, this data set suffers from the limitation that interviewers who were fabricating data were not given an incentive to fabricate data realistically. Therefore, the fabricated data may be artificially easy to detect, and the performance

Data Set $\mathcal{A}_{\mathbf{u}}$

```
**********
Question 1 (score = 412.2): Do you want
to know more about family planning?
                            No   Yes
Interviewer 1 (312 forms):  11%  89%
Everyone (3131 forms):      71%  29%

**********
Question 2 (score = 335.4): Is the client
available?
                            No   Yes
Interviewer 2 (258 forms):  91%  9%
Everyone (4321 forms):      28%  72%

**********
Question 3 (score = 101.9): Do you want
to know more about family planning?
                            No   Yes
Interviewer 3 (100 forms):  11%  89%
Everyone (3131 forms):      71%  29%

**********
Question 4 (score = 95.5): Did you give
any referral in this household?
                            No   Yes
Interviewer 3 (100 forms):  62%  38%
Everyone (3131 forms):      93%  7%

**********
Question 5 (score = 85.6): Did anyone in
the household have a fever yesterday or
today?
                            No   Yes
Interviewer 3 (100 forms):  68%  32%
Everyone (3131 forms):      94%  6%
```

Data Set $\mathcal{B}$

```
**********
Question 1 (score = 41.3): If yes,
which of these marketing activities do
you do with this group? (Choosing crops
or varieties to grow and market)
                             No   Yes
Interviewer 1 (16 forms):    44%  56%
Everyone (265 forms):        94%  6%

**********
Question 2 (score = 36.5): Why did you
find yourself without food? (There was
no food distribution)
                            No   Yes
Interviewer 2 (8 forms):    0%   100%
Everyone (265 forms):       94%  6%

**********
Question 3 (score = 35.5): Why did you
find yourself without food? (No one was
willing to offer us some food)
                            No   Yes
Interviewer 2 (8 forms):    0%   100%
Everyone (265 forms):       93%  7%

**********
Question 4 (score = 32.1): What did
you do with the information you got
from the FIELD OFFICER? (I started
storing my produce at a warehouse or
collection point)
                            No   Yes
Interviewer 3 (9 forms):    11%  89%
Everyone (265 forms):       89%  11%

**********
Question 5 (score = 31.7): What
information have you ever got from a
FIELD OFFICER? (Supplying to WFP)
                            No   Yes
Interviewer 3 (9 forms):    0%   100%
Everyone (265 forms):       84%  16%
```

Figure 4.4: The most anomalous question distributions according to MMA in data sets $\mathcal{A}_u$ and $\mathcal{B}$.

reported in Section 4.3 may be overly optimistic. In this section, I address this limitation

by evaluating the unsupervised algorithm MMA on the data sets $\mathcal{A}_u$ and $\mathcal{B}$.[3] These data sets are not labeled, so it is hard to *quantitatively* evaluate MMA on them. However, they are from actual field operations of Organizations A and B, so it is possible to *qualitatively* assess whether MMA can find compelling outliers in these realistic data sets.

Figure 4.4 shows, for both data sets, the five most surprising interviewer-question pairs according to MMA. Each entry in the figure shows the question text, along with the interviewer's distribution compared to the overall distribution from all interviewers. For example, the figure shows that the most surprising distribution from an interviewer at Organization A, with an m_j^i value of 412.2, is "Do you want to know more about family planning?" Interviewer 1 said that 89% of her clients wanted to learn more about family planning, whereas the proportion of clients, over all interviewers, who responded this way was only 29%. Although there could be many causes for this type of anomaly besides curbstoning, it seems clear that a supervisor would want to follow up on it. Indeed, after I showed a similar table to supervisors at Organization B, they informed me that they found information like this very useful and were interested in the possibility of using algorithms such as MMA as part of their operations.

One could ask whether the types of anomalies shown in Figure 4.4 are what would be expected from chance fluctuations alone. This does not seem to be the case. A justification is shown in Figure 4.5, which helps to visualize interviewer correlations for the m_j^i values in data set $\mathcal{B}$. Each bar in this figure corresponds to an interviewer; the height is the number of questions for which the interviewer's distribution has an m_j^i value in the top 95th percentile of the values for all interviewer-question pairs. The bars are sorted from left to right according to this frequency, and only the top 15 interviewers are shown.

Figure 4.5 shows that two interviewers have many more surprising answer distributions than other interviewers. If the m_j^i values were independently and identically distributed, then for a given interviewer, the chance that a particular question distribution would make it in the 95th percentile is 0.05. Thus, for the first interviewer in Figure 4.5, the chance that at least 27 questions would randomly appear in the 95th percentile is the chance that a binomial random variable with parameters $n = 103$ (the number of questions) and $p = 0.05$ would achieve a value of at least 27, which is 8.5×10^{-13}. Since

[3]Because these data sets are unlabeled, it is impossible to evaluate supervised algorithms on them. Of the unsupervised algorithms, I focus on MMA in this section. I do not evaluate LOCI because it is hard to qualitatively assess how compelling individual form outliers are. I do not evaluate SVA because when the number of forms per interviewer varies significantly and there are some questions with very few non-blank forms from certain interviewers, as is true for $\mathcal{B}$, SVA can have a tendency to give artificially high scores to interviewer-question pairs in which the number of forms is small.

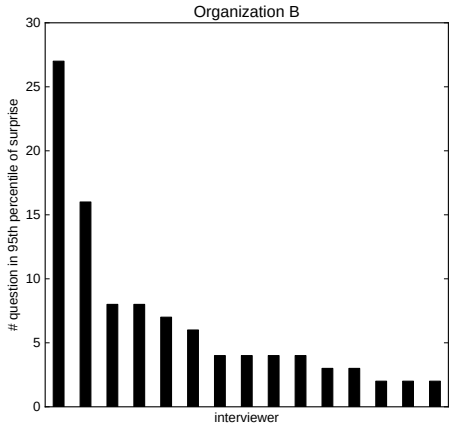


Figure 4.5: Number of anomalous questions by interviewer in data set $\mathcal{B}$. Each bar shows the number of questions per interviewer that have m_j^i values in the 95th percentile. The interviewers are sorted from left to right according to this number. (Only the top 15 interviewers are shown.) The abnormally high frequency of the first two interviewers suggests that the m_j^i values are not due to chance alone.

there are 26 interviewers, the chance that this would occur for any interviewer is less than $26 \cdot 8.5 \times 10^{-13} = 2.2 \times 10^{-11}$. Hence, it seems highly likely that something besides chance alone is causing this interviewer's distributions to be so different; a supervisor would be well-advised to investigate further.

4.5 Chapter summary

In this chapter, I described a preliminary investigation into the feasibility of using supervised classification and unsupervised anomaly detection algorithms to identify curbstoning. To do so, I ran the algorithms on data sets from two CHW organizations in East Africa. First, I showed that on the labeled data set $\mathcal{A}_\ell$, the supervised algorithms achieve an accuracy of up to 81% at the form-level and 97% at the interviewer level, and that the unsupervised algorithms identified curbstoning with good sensitivity and specificity. Second, I showed that one of the unsupervised interviewer-level techniques, MMA, was able to find compelling anomalies in more realistic, but unlabeled, data sets from the organizations.

Although these initial results are promising, there are two main limitations of the work presented so far. First, as mentioned before, interviewers in the fake data party were not given an incentive to fabricate data realistically. Thus, the quantitative evaluations in Sec-

tion 4.3 may be overly optimistic. Second, the potential of user trace metadata features to detect curbstoning was left largely unexplored. I did show that the `total-time` feature was predictive of fabricated data, but this feature just scratches the surface of the types of rich metadata features that could be recorded by mobile devices. In the next two chapters, I describe a study that addresses both of these limitations.

Chapter 5
STUDY HABITS SURVEY: DESIGN AND EXECUTION

The research that I presented in Chapter 4 provided preliminary evidence that supervised classification and unsupervised anomaly detection algorithms could be used to detect curbstoning, but it did little to address two major parts of my thesis: first, that the algorithms can work even when interviewers are aware that they are being used and are incentivized to avoid detection; and second, that their accuracy can be improved by using user-trace data recorded by mobile electronic devices during data collection. In this chapter, I describe a study that I performed to simultaneously explore both of these aspects of my thesis.

In the study, I had participants conduct a short survey on study habits using software called Open Data Kit (ODK) Collect installed on their Android mobile phones. As in the work described in Chapter 4, the interviewer participants not only collected real data, but also created fabricated data in controlled settings. Unlike the work in Chapter 4, however, interviewers in this study were given information about how the algorithms worked and were incentivized to fabricate data in a way that avoided detection. They also used a modified version of ODK Collect that recorded extensive user trace logs. These two features of the study allowed me to delve more deeply into the questions posed by my thesis.

In this chapter, I discuss the design and execution of the study, deferring the discussion of the data collected to the next chapter. I start in Section 5.1 by describing the survey on study habits that the interviewers conducted for this study. I discuss the design considerations that led to the survey, and I give a general description of the questions in it. (The entire content of the survey is given in Table A.1.) In Section 5.2, I give relevant background on ODK Collect. In Section 5.3, I describe the protocol of the study. I show how both participants' knowledge of the algorithms and the incentives provided for realistically fabricating data were carefully controlled and varied so that the effect of such knowledge and motivation could be studied. (Some further technical detail on the protocol, including the text of scripts that were used, can be found in Appendix B.) In Section 5.4, I describe the structure of the log files that recorded user interaction traces on ODK Collect. For these raw log

files to be useful, numeric and categorical features needed to be extracted from them. In Section 5.5, I motivate and define the features that I extracted from the logs. I conclude the chapter, in Section 5.6, by summarizing the actual execution of this study. I describe how 28 participants from the University of Washington were recruited to interview 256 respondents and generate 448 fabricated forms. I also discuss data quality indicators, how I cleaned the data, and how I separated the data into training and test sets. This provides the background needed to delve into the data in the next chapter and explore whether it supports my thesis.

5.1 The Study Habits survey

The survey on study habits that I used for this study contained 44 questions: 27 categorical, 7 numeric, and 10 free text. The survey was designed to take between 5 and 10 minutes to complete. It started with a series of questions to determine whether the respondent was eligible for the survey. To be eligible, a respondent had to be a university student, be between age 18 and age 25, and have never taken the survey before.

Once the respondent was determined to be eligible, the survey asked for details about her major. It then asked for details about the hardest class she had taken in the last year, including its name, when she took it, whether she liked it, and how many hours a week she spent on it. Next, it asked how much time she spent on obligations outside of school, including paid work, volunteer work, research, and family. Following that, it asked whether and how much she would be willing to pay for a major she was interested in if it cost extra. Then it asked a series of questions to determine how often the respondent sought help in her studies from faculty, TAs, tutors, and advisors. It concluded by asking whether the respondent believed that she studied more than the average student, procrastinated more than the average student, and had more obligations outside of school than the average student.

For conciseness, I will refer to questions by a *unique identifier*. These identifiers, along with the entire content of the survey, are given in Table A.1. This table shows, for example, that Question 22 has the identifier `average-work` and the text "On average, how many hours do you spend studying?"

Unlike the survey in Chapter 4, the Study Habits survey was designed specifically for the study on curbstoning detection. I considered several factors in its design:

- *Variety of answer types*. To be general, the survey included a substantial number of categorical, numeric, and free text responses.

- *Variance in answer difficulty*. Some of the questions were straightforward for the interviewer and respondent, like `sex` ("What is your sex?") and `major` ("What is your major?") while some required more thought, like `average-work` ("On average, how many hours do you spend studying?") or `faculty` ("How often in a quarter do you typically seek help from faculty for your course work?"). My hypothesis was that the time to complete harder questions would vary more substantially between real and fake data than the time to complete easier questions would. Having both hard and easy questions would allow me to test this hypothesis.

- *Branching logic*. Many of the questions in the survey were asked only if an earlier question was answered in a particular way. For example, respondents were only asked what their major was if they had indicated earlier that they had decided on a major. The number of such conditional questions was 17, so depending on how questions were answered, the number of questions that a respondent was actually asked varied from 27 to 44. I included such a large number of conditional questions to test the *short paths through survey* hypothesis, described in Section 2.1, that posits that curbstoners tend to ask fewer conditional questions [21, 59].

- *Pertinent questions*. Although I created the Study Habits survey specifically to evaluate curbstoning detection algorithms, I did not want the interviewers and respondents to take it less seriously because of that. To lend credibility to the questions, I chose questions in consultation with the undergraduate advisors at the University of Washington Department of Computer Science & Engineering (CSE). The involvement of the CSE advisors with the design of the survey was communicated in writing and verbally to the interviewers and respondents.

The survey questions were not tested with the target population before they were finalized, and as a result, some of the questions turned out to be ambiguous. For example, many people did not think it made sense for a major to be both "intended" and "declared," which was an option provided by the question `intended-major`. However, the purpose of my research was to study curbstoning—not to draw conclusions based on the response data; therefore, such ambiguities do not affect the validity of this study.

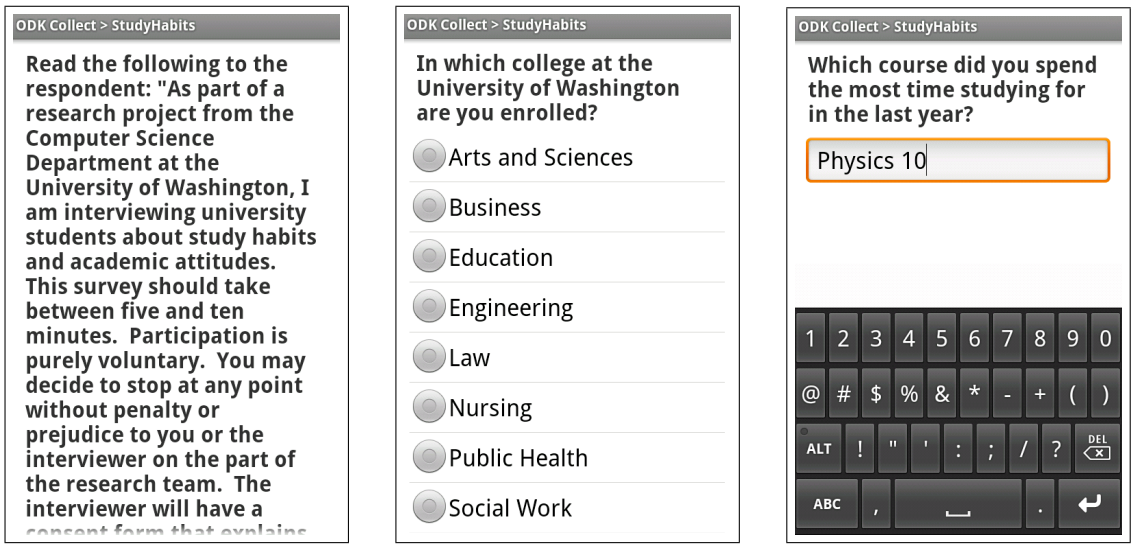


Figure 5.1: Screenshots from Study Habits survey on ODK Collect. On the left is a note prompt (`consent-note`); in the middle is a categorical question prompt (`uw-college`); and on the right is a free text question prompt (`major`).

5.2 Background on Open Data Kit

Open Data Kit (ODK) is "an extensible, open-source suite of tools designed to build information services for developing regions" [53]. For this study, I used one of the tools from the suite called Collect, which is data collection software for mobile devices that use the Android operating system. Although Collect will run on both tablets and mobile phones, participants in this study used it only on phones. Collect supports many different data types, including categorical, numeric, and free text. It also supports multimedia types like images, audio, and video, but this functionality was not required for this study.

In Collect, a survey consists of *prompts*, which can be either read-only *note prompts* or interactive *question prompts* for data entry. (In addition to the 44 question prompts in the Study Habits survey, there are another 8 note prompts, which are also listed in Table A.1.) To move forward through the survey, users *swipe* the touch screen with their fingers, a motion similar to turning the pages of a book. To move forward, users swipe their finger to the left, and to move backward, they swipe their finger to the right. To choose a response for a categorical question, users click on the desired response. To enter a numeric or free text response, they use the keyboard on their phone (either a physical keyboard, if one exists, or the "soft" keyboard that appears on the touch screen.) If the entire prompt is too big to

Phase	Components	Duration
Training session	Introduction to study Demonstration Generation of uninformed fake data ($\texttt{fake}_0$) Generation of first real data	1-2 hours
Interview period	Interviews of 10 respondents	1 week
Follow up session	Generation of informed fake data ($\texttt{fake}_1$) Generation of better-informed fake data ($\texttt{fake}_2$)	1-2 hours

Table 5.1: Summary of study protocol.

fit in the phone's screen, users can scroll through the prompt vertically with their fingers. Figure 5.1 shows screenshots from three different prompts in the Study Habits survey.

5.3 Study protocol

In this section, I describe the protocol that the interviewer participants followed, both to conduct interviews with actual respondents and to fabricate data in controlled settings. Participation in the study lasted one week and consisted of three phases: (1) the *training session*, which took place in the lab; (2) the *interview period*, which took place on participants' own time; and (3) the *follow up session*, which took place in the lab.

Participants fabricated data during both the training session and the follow up session. Their knowledge of the purpose of the study and the curbstoning detection algorithms increased over time. In the training session, they fabricated data, but they were not told why they were doing so. As in the study from Chapter 4, they were told only to "pretend they were interviewing someone and answer as he or she would." In the follow up session, they fabricated data in two additional rounds. In the first round, they were told the the purpose of the study; in the second round, this knowledge was supplemented with personalized feedback about how the algorithm was identifying their fabricated data. In both rounds, they were given a monetary incentive if they fabricated data realistically.

In the rest of this section, I describe this protocol in more detail. Table 5.1 summarizes this protocol at a high level. (I discuss the details of data collection, including how I recruited participants, in Section 5.6.)

5.3.1 Training session

Participants began the study by coming into the lab for a training session, sometimes alone and sometimes in small groups. In this session, participants signed consent forms, were given a demonstration of how to carry out a survey, and generated their first real and fabricated data. Although participants were told that that one of the purposes of the study was to test new features of ODK involving data quality, they were never told anything more specific. In particular, *during the training session, the explicit purpose of the study—to test algorithms to identify curbstoning—was not mentioned.*

The training sessions were structured as follows.

Part 1 – Introduction to study

First, participants signed the consent form. Next, I told them that my research group was running a survey about study habits on behalf of the CSE undergraduate advisors at the University of Washington. I told them that they would be interviewers for this survey, and that for the week between the training and follow up sessions, they were to administer the survey using ODK Collect on any 10 people they could find, as long as they fit the requirements of the study given in Section 5.1. I told them that I was interested both in the data itself and in testing some new unspecified features of ODK Collect involving data quality.

Part 2 – Demonstration

After the introduction, I demonstrated how interviews should be conducted using ODK Collect. First, I outlined the required procedure for obtaining informed consent from respondents. Then, I gave the participants consent forms for the respondents and told them that they were to read the `consent-note` prompt aloud before getting a signature from the respondent.

Next, I walked through each question on an emulated Android phone projected for everyone to see. I explained that participants were to read from and enter data into the phone themselves, instead of letting the respondents self-administer the survey. I took particular care to emphasize the branching logic of the survey. I encouraged participants to ask questions during this demonstration.

At the end of the demonstration, I walked through the `name-phone-note` prompt and explained that because data quality was important to me in this study, I would be performing reinterviews on a random subset of the respondents who consented to leave their name and phone number. I stressed that consent should be obtained carefully before respondents

left their numbers.

Part 3 – Generation of uninformed fake data
After the demonstration, I asked the respondents to fill out five different forms, telling them to "pretend they were interviewing five different people and answer as they would." The data collected during this portion of the training session was the first fabricated data created by the participants. Instead of telling the participants the true purpose of this fabrication, I implied that it was to gain familiarity with the survey. For this reason, I refer to this data, labeled $\texttt{fake}_0$, as the *uninformed fake* data.

Part 4 – Generation of first real data
After all of the participants finished generating the $\texttt{fake}_0$ data, I told them that to practice performing the survey more realistically, they would take turns interviewing each other. They were to do everything they would in a normal survey, including reading the `consent-note` prompt and getting a signed consent form from the respondent. This data was the first data labeled `real`. If there was only one participant in the training session, I skipped this portion of the training session. At the end of the training session, participants were given a $20 gift card as compensation for their time.

5.3.2 Interview period

As I explained to participants, following the training session was a one-week interview period. I asked each participant to administer the survey to 10 eligible respondents during this period. There was, however, no enforcement of this request, and compensation was not dependent on the number of respondents interviewed.

During the interview period, I had no contact with the participants, except to answer questions and schedule the follow up session. Participants conducted the surveys whenever they wanted, using their own Android phones or a loaner Android phone if they did not have their own. The data collected during this period was given the label `real`. Because of my lack of contact with the participants, data quality was a concern. I describe how I addressed this concern in Section 5.6.2.

5.3.3 Follow up session

After the interview period, I asked participants to come back to the lab for a follow up session. As for the training session, participants were allowed to attend alone or in groups, and the session was designed to take between 1 and 2 hours. In the follow up session, I told participants the true purpose of the study and gave them two more rounds to fabri-

cate data. In both rounds, a monetary reward was given to the participants who fabricated data particularly realistically. The primary difference between the first and second round was that in the first round, participants had little knowledge of how the curbstoning detection algorithm worked, whereas in the second round personal feedback was given to participants describing how the algorithm was identifying their fabricated data. The purpose of doing this was to study how robust the curbstoning detection algorithms were to adaptions from interviewers as they learned how the algorithms worked.

The follow up sessions were structured as follows.

Round 1 – Generation of informed fake data
First, participants uploaded the `real` data collected during the week. Next, I read participants a script informing them that the true focus of this study was to validate algorithms to identify fabricated survey data. Then, I told them that they would have 40 minutes to fabricate between 4 and 10 more forms. This allowed participants to spend between 4 and 10 minutes per survey, which turned out to easily contain typical survey times seen during the collection of real data. I provided this flexibility so that participants would not face time constraints that could artificially differentiate the fabricated data from the real data.

To incentivize the participants to fabricate data realistically, I told them that they would be given a *fabrication score* indicating how well they faked data; if the score was good enough, they would receive an extra $10 gift card. The particular requirement to receive the extra compensation depended on whether there was one or more than one participant in the follow up session. If there was one participant, then a participant had to do better than a predetermined cutoff, chosen to be the approximate median score seen during the pilot period. (See Section 5.6 for details on the pilot period.) If there was more than one participant, then a participant had to get the best score in the session.

The way scores were calculated changed slightly between the pilot period and main data gathering period. In both cases, a participant's score was based on the probability of being fake that a simple supervised classifier assigned to her fabricated instances. The lower these probabilities were in aggregate, the better. Details of the scoring mechanism and the complete text of the scripts read to the participants can be found in Appendix B. The data collected during Round 1 was called the *informed fake* data and given the label $\texttt{fake}_1$.

Round 2 – Generation of better-informed fake data
After participants finished fabricating their Round 1 data, I ran a script that extracted 9 features for each form, called the *feedback-eligible features*. I chose these features before I

Our algorithm logs how many seconds you spend on the screens in which you are supposed to read something to the respondent. For each submission you gave us, it found the total number of seconds you spent on these screens. This number tended to be higher in your fabricated data than in the real data.

Our algorithm logs whenever you move backwards through the form (by swiping to the right). For each submission you gave us, it computed the total number of times you moved backwards. This number tended to be lower in your fabricated data than in the real data.

Our algorithm logs the total amount you scrolled up or down on a screen (in pixels) on each submission you gave us. This number tended to be lower in your fabricated data than in the real data.

Figure 5.2: Example feedback email sent to respondent.

started collecting data because—based on my previous experience and prior research—I thought they would be both predictive of fabrication and easy to explain to participants. Table 5.2 shows each of the feedback eligible features, along with how they were explained to participants when applicable. I provide a full definition of all features, including the feedback eligible features, in Section 5.5.

After the script extracted the feedback eligible features, it trained a simple classifier on these features and used that script to generate the fabrication scores. (I describe more details of this classifier in Appendix B.) Next, I read the scores for each participant aloud to the group. I immediately gave the $10 gift card to the winning participant, if there was one.

After that, I read participants a script that explained that during the next few minutes, I would be sending each of them an email that contained the three feedback-eligible features that were most correlated with their own data being fake, along with the direction of that correlation. I told them that they should think of these features as the "clues" that the algorithm used to determine that their fabricated data was actually fake. The emails were generated using the explanations given in Table 5.2. Figure 5.2 shows an example of an actual email sent during one of the follow up sessions.

After they received these emails, they had another 40 minutes to fabricate between 4 and 10 surveys. I told them that, to fabricate data more realistically, they should keep in mind their "clues." As in the first round, I gave them a reward of $10 if they fabricated data real-

Feature	Explanation
`max-select-one-edits`	"Our algorithm looks at the number of times you change a response to each multiple choice question. For each submission you gave us, it found the maximum number of times a multiple choice question response was changed."
`mean-select-one-edits`	"Our algorithm looks at the number of times you change a response to each multiple choice question. For each submission you gave us, it found the mean (average) number of times a multiple choice question response was changed."
`mean-string-length`	"Our algorithm looks at the length of all text responses that you give. (This does not include multiple choice questions or questions with a numeric value.) For each submission you gave us, it found the mean (average) length of the text responses."
`median-answer-time`	"Our algorithm logs how many seconds you spend on each question. For each submission you gave us, it found the median (average) number of seconds you spent on all of the questions."
`note-time`	"Our algorithm logs how many seconds you spend on the screens in which you are supposed to read something to the respondent. For each submission you gave us, it found the total number of seconds you spent on these screens."
`num-conditional`	"Some of the questions in the survey are conditional: whether they appear depends on the responses to earlier questions. Our algorithm logs whenever one of these conditional questions appears. For each submission you gave us, it computed the total number of conditional questions that appeared."
`num-previous`	"Our algorithm logs whenever you move backwards through the form (by swiping to the right). For each submission you gave us, it computed the total number of times you moved backwards."
`num-swipes`	"Our algorithm logs whenever you move from one question to another (going either forward or backward). For each submission you gave us, it computed the total number of times you moved from one question to another."
`total-scrolled`	"Our algorithm logs the total amount you scrolled up or down on a screen (in pixels) on each submission you gave us."

Table 5.2: Feedback-eligible features. Each participant received three of the explanations, corresponding to the three of these features that were most correlated with the participant's fabricated data. Each explanation was followed by "This number tended to be _____ in your fabricated data than in the real data," where the blank was either "higher" or "lower," depending on whether the feature was positively or negatively correlated with fabrication.

Label	Name	Description
`real`	*real*	Real interviews between participants and respondents, including the interviews that occurred at the end of the training sessions.
$\texttt{fake}_0$	*uninformed fake*	Fabricated interviews from the training session, in which participants were told to "pretend they were interviewing five different people, and answer as they would." For this data, the participants were not informed of the true purpose of the study and were not given an incentive to fabricate data realistically.
$\texttt{fake}_1$	*informed fake*	Fabricated interviews from the first round of the follow up session. For this data, the participants knew the true purpose of the study and were given a monetary incentive to fabricate data realistically, but they did not have any information about how the fabrication detection algorithm worked.
$\texttt{fake}_2$	*better-informed fake*	Fabricated interviews from the second round of the follow up session. For this data, not only did the participants know the true purpose of the study and receive an incentive to fabricate data realistically, but also they had been given some feedback on features that the algorithm used to identify their fabricated data.

Table 5.3: Summary of data labels.

istically enough. If there was more than one participant in the session, the requirement to get the $10 was again to get the best score in the session. If there was only one participant, then the requirement was to improve the score the participant got in Round 1 by a factor of two.

The data collected during Round 2 was called the *better-informed fake* data and given the label $\texttt{fake}_2$. At the end of the follow up session, participants were given a $30 gift card—in addition to any of the additional award money received—as compensation for their time.

Table 5.3 summarizes the four different types of real and fabricated data that were obtained in all three phases of the study.

Event type	Description	Parameters
answer selected	An answer is selected for a categorical question prompt. This could happen more than once for a given prompt if the user changes the response.	The form identifier, the prompt name, the response selected, the position of the response selected.
data saved	A form is saved.	The form identifier.
form deleted	A form is deleted.	The form identifier.
instance loaded	A form is loaded.	The form identifier.
next	A prompt is reached from a forward (left) swipe.	The form identifier, the prompt name.
previous	A prompt is reached from a backward (right) swipe.	The form identifier, the prompt name.
scroll	The user scrolls up or down on a prompt.	The form identifier, the prompt name, the amount scrolled in pixels (where a positive number means a scroll up and a negative number means a scroll down).
text changed	The text is changed for a numeric or free text question prompt.	The form identifier, the prompt name, the answer text prior to the change, the answer text after the change.

Table 5.4: User-trace log event types. In addition to the parameters listed in this table, each event was also tagged with a timestamp.

5.4 User trace logs

By default, the only metadata that ODK Collect records is the time that each form was created. To test the utility of more extensive user-trace metadata, I modified the source code of Collect to implement detailed user trace logging.

In my implementation, each log entry consisted of an *event type* and a timestamp. Various parameters were also recorded depending on the event type. For example, many log entries were tagged with a form and prompt identifier to indicate where the interviewer was in the survey when the event occurred. Table 5.4 lists all of the event types, a description of what they signified, and the parameters that were recorded along with them.[1]

The logs were stored as SQLite files on participants' phones, which were uploaded to a

[1] I actually recorded more types of events then those listed in Table 5.4, but these types were the only ones that I ended up using for feature extraction.

server during the training and follow up sessions. They were identified on the server by a hash of the phone's device identifier, which made it possible to associate the logs by interviewer. By looking at the most recent *answer selected* and *text changed* events for each form, the actual responses for that form could be reconstructed. Because of this, the traditional XML storage mechanism implemented by ODK Collect was bypassed and the only data uploaded from participants' phones were the log files.

5.5 Feature extraction

The log files described in the previous section were linear recordings of a series of events in time, but the algorithms described in Chapter 3 require numeric or categorical features. The parameters recorded in the log entries, including the timestamps, made it possible to extract a wide variety of numeric and categorical features from the log files. For example, to get the amount of time a user spent on a given prompt on a given form, an algorithm could first search for all events that are tagged with the form and prompt identifier. Let the set of these events be S. Then, supposing that all of the log events are numbered in sequential order, the algorithm could add up the differences between the timestamp of each event $i \in S$ and the timestamp of its successor event $i + 1$ (which may or may not be in S.) The result of this sum is the time spent on the given prompt.

In this section, I define all of the features that I extracted from the logs. My motivation for choosing features varied. Some of the features were inspired by the related work described in Section 2.1 that suggested various traits that might characterize fabricated data, such as *fast interviews*, *short paths through survey*, and *low data variance*. Other features were not inspired by a particular hypothesis about how they would predict curbstoning, but they seemed to me to be potentially useful.

There were some features that I could not include in the study. GPS data, for example, could be a strong predictor of fabrication, but because for this study, all data fabrication occurred in the same physical location, it was not appropriate to include it. The particulars of the study format also ruled out many of the remaining features described in Section 2.1, especially the time-of-day-based features such as *interview surge*, *close to deadline*, and *unusual time of day*. In Section 7.2.3, I discuss the possibility of using more sophisticated user-trace logging as a direction for future research.

The features extracted from the logs are divided into three categories: *prompt-level features*, which pertain to individual prompts within a form; *form-level features*, which pertain to the

entire form; and *interviewer-normalized features*, which are features that have been adjusted by what is typical for the interviewer. I discuss each in turn, and conclude in Section 5.5.5 by describing the naming convention that I use to refer to features from this study. Note that some of these features have already been discussed in the context of the feedback-eligible features of Table 5.2.

5.5.1 Prompt-level features

I extracted the following prompt-level features:

- `response` – the actual value of the response for the prompt, if it was a question prompt. (This is the only thing that would typically be recorded by ODK Collect for the question.)
- `time` – the total number of milliseconds spent on the prompt.
- `num-next` – the total number of times, within a single form session, that the user swiped forward (left) to the prompt.
- `num-previous` – the total number of times, within a single form session, that the user swiped backward (right) to the prompt.
- `delay-to-first-edit` – the number of milliseconds between when the prompt was first swiped to and when the the first edit was made.
- `num-edits` – the number of times the user changed the response for the prompt, if it was a categorical, numerical, or free text question prompt. This number is the sum of two sub-features:
  - `num-contiguous-edits` – the number of times the user re-edited the response contiguously without changing prompts in between.
  - `num-non-contiguous-edits` – the number of times the user edited the response immediately after swiping to the prompt.

  As an example to illustrate the meaning of these features, suppose that the sequence of edits for a question prompt was as follows. First, the user swiped forward to the prompt, chose a response, then chose another response, and then swiped forward to the next question. Later, the user swiped backward to the same prompt and chose a different answer before swiping forward again. For this prompt, `num-edits` is 3, `num-contiguous-edits` is 1, and `num-non-contiguous-edits` is 2.

- `scrolldown` – the total number of pixels scrolled down on the prompt.
- `scrollup` – the total number of pixels scrolled up on the prompt.
- `ord` – for ordinal questions (`when-hardest-course`, `average-work`, `faculty`, `tas`, `tutors`, `advisors`), a positive integer indicating the response's position in the order.

5.5.2 Form-level features

I extracted the following form-level features:

- `total-time` – the value of the last timestamp seen in the logs for the form minus the value of the first timestamp seen in the logs for the form. Note that because the set of possible timestamps includes timestamps from re-editing sessions, the value of this feature could be much higher than the actual time spent actively editing the form.
- `total-answer-time` (`median-answer-time`, `min-answer-time`) – the sum (median, minimum) of the `time` features for all question and note prompts.[2]
- `total-delay-to-first-edit` (`median-delay-to-first-edit`, `min-delay-to-first-edit`) – the sum (median, minimum) of `delay-to-first-edit` features for all question prompts that were edited at least once.[2]
- `mean-string-length` – the mean length of all non-empty free text responses apart from `phone` ("What is your phone number?").
- `note-time` – the total time in milliseconds spent on note prompts.
- `num-conditional` – how many of the 17 conditionally-appearing question prompts were answered.
- `total-time-conditional` – the total time spent on conditionally-appearing question prompts.
- `mean-select-one-edits` (`max-select-one-edits`) – the mean (max) of the `num-edits` feature values for the categorical question prompts that were edited at least once.[3]
- `mean-select-one-contiguous-edits` (`max-select-one-contiguous-edits`) – the mean

[2] I also experimented with using the mean instead of the median, but preliminary experiments showed that the `mean-` version of this feature was typically dominated as a predictor by either the `median-` or `total-` version.

[3] These features were feedback-eligible features, but they were removed during all subsequent analysis because of their linear dependence on the `select-one-contiguous-edits` and `select-one-non-contiguous-edits` features.

(max) of the num-contiguous-edits feature values for the categorical question prompts that were edited at least once.

- mean-select-one-non-contiguous-edits (max-select-one-non-contiguous-edits) – the mean (max) of the num-non-contiguous-edits features values for the categorical question prompts that were edited at least once.
- num-swipes – the total number of forward and backward swipes on the form.
- num-previous – the number of backward swipes on the form.
- total-scrolled – the total number of pixels scrolled up and down while the form was being filled out.
- times-old-response-checked – the number of times the following event happened: when filling out the response to a question, the user moved backward some number of questions and then forward to the original question, without changing any response along the way.
- times-why-not-avoided – the number of times the following event happened: first, the user chooses the response "zero" to one of the question prompts faculty, tas, tutors, or advisors; next, she swipes forward to the follow up -why-not question that appears only when "zero" is selected; finally, she swipes backwards to the original question prompt and changes the response to something besides "zero." (The idea for this feature came from observing this behavior in the logs and hypothesizing that it might be a strategy to avoid extra work that could indicate curbstoning.)
- total-hours-on-activities – the sum of the response features for the question prompts paid-work-hours, volunteer-work-hours, research-hours, family-hours, and other-obligation-hours.
- num-activities-more-than-one-hour – the number of response features for the question prompts paid-work-hours, volunteer-work-hours, research-hours, family-hours, and other-obligation-hours that had a value greater than zero.

Note that these last three features—times-why-not-avoided, total-hours-on-activities, and num-activities-more-than-one-hour—only make sense in the context of the Study Habits survey. It is not immediately clear how they would generalize to other surveys. However, it turns out that these three features are not very strong predictors of fabrication. (I discuss this result further in Section 6.6.) Therefore, my recommendation to someone attempting to follow this model of feature extraction for her own survey is to avoid using

form-specific features like these. (I discuss this topic further in Section 7.1.2.)

5.5.3 Interviewer-normalized features

All of the features described so far are *raw* in the sense that they are compared on the same scale regardless of who the interviewer was. This naive comparison ignores important information—what might be a high value of a feature for one interviewer might be low for another. Because of this, I added *interviewer-normalized features* that adjusted feature values based on what was typical for the interviewer.

Specifically, I calculated the mean and standard deviation across each interviewer for the following features:

- `response` for each numeric question prompt;
- `time`, `delay-to-first-edit`, `num-contiguous-edits`, `num-non-contiguous-edits`, `ord` for each question prompt for which the feature was defined; and
- all of the form-level features except `mean-select-one-edits` and `max-select-one-edits`.

Then for each form and each feature, I computed the difference and squared difference, measured in standard deviations, between the value of that feature for the form and the mean value of that feature for all forms computed by the interviewer. I denote these features by prefixing the feature name by either `diff-` or `sqdiff-`, depending on whether they correspond to the difference or the squared difference.[4] For example, the value of the feature `sqdiff-total-delay-to-first-edit` on a form f from interviewer i is the squared difference in standard deviations between the `total-delay-to-first-edit` feature for form f and the mean value of the `total-delay-to-first-edit` feature over all of interviewer i's forms.

5.5.4 Summary of feature extraction

The feature extraction steps just outlined resulted in a total of 641 numeric and categorical features: 209 prompt-level, 22 form-level, and 410 interviewer-normalized. Expanding the 26 categorical features according to the procedure outlined in Section 3.2.1 resulted in a total of 667 features. Three of these features did not vary over the training set, so in effect

[4]Initially, I performed the same procedure for the median and inter-quartile range (IQR), but a preliminary investigation revealed that the interviewer-normalized features based on these values were not as predictive.

there were a total 664 features.[5]

5.5.5 Feature naming conventions

I use the following conventions to refer to the features defined in this section. Prompt-level features are prefixed with the name of the prompt, whereas form-level features have no prefix. For example, `more-costly-num-previous` refers to the number of times the `more-costly` prompt was reached via a backward swipe, whereas `num-previous` refers to the total number of backward swipes throughout the editing of the entire form.

As mentioned in the previous section, the interviewer-normalized features are be prefixed by either `diff-` or `sqdiff-`, depending on whether they correspond to the difference or the squared difference. This prefix comes before any applicable prompt-level prefix. So, for example, the feature `diff-taken-before-num-non-contiguous-edits` refers to the difference, in standard deviations, between the value of `num-non-contiguous-edits` for the `taken-before` prompt on a given form and the mean of that value for all of the interviewer's forms.

5.6 Execution of study

Data for this study was collected between April 12, 2012 and May 14, 2012. Twenty-eight interviewers participated during this time frame, collectively interviewing 256 respondents and generating 448 fabricated forms.

5.6.1 Participant recruiting

I recruited interviewer participants through email advertisements and posters. Because the participants had to play the role of a respondent during Part 4 of the training session, they also had to be university students between the ages of 18 and 25.[6] My advertisements stated that participants must own their own Android phone onto which I could install ODK Collect. Three of the 28 participants did not have their own phone but instead used a loaner phone that I provided. Of the 28 participants, 18 were male and 10 were female. All were undergraduate or Masters students at the University of Washington. Most of them were recruited from engineering majors.

The first six interviewers comprised the *pilot group* of participants. During their participa-

[5]These features are far from independent, however. A *principal components analysis* [64] revealed that 99% of the variation of this data set was contained in a space having dimension 305.

[6]However, I found out during the course of the study that one of the male participants was 27 and one of the female participants was 26.

tion, as detailed in Appendix B, I tweaked the scripts used and the technical details of how the fabrication score was computed. Because these changes were minor, I did not separate these interviewers' data from the rest.

There were no mid-study dropouts. That is, all of the 28 participants who attended a training session also came to a follow up session. There were 11 training sessions and 11 follow up sessions. Three of the training sessions had only one participant, and the maximum size of a training session was 5; only one of the follow up sessions had one participant, and the maximum size of a follow up session was 4.

Although each interviewer's involvement with the study was intended to last exactly one week, because of last-minute scheduling changes, this period actually varied from a minimum of 5 days to a maximum of 9 days. There were also some discrepancies from the protocol in the number of forms per interviewer having each label. For example, the number of forms per interviewer labeled `real` varied from 3 to 16 and the number of forms per interviewer labeled $\texttt{fake}_0$ varied from 1 to 5. However, as Figure 5.3 shows, this number did not typically vary substantially from what the protocol specified.

5.6.2 Data quality indicators

In order to accurately test the ability of algorithms to detect curbstoning, it was important that the data labeled `fake` actually be fake and the data labeled `real` actually be real. For this study, that the `fake` data was actually fake is not in question since that data was generated in the lab under my supervision. However, that the `real` data was actually real is a different matter—it is possible that the interviewers in this study committed curbstoning when they were not supposed to.

I took two steps to minimize the possibility. First, as discussed in Section 5.3, I informed interviewers that there would be random reinterviews on a subset of the respondents. I performed these reinterviews according to the following protocol. For each interviewer, I listed in random order all of the respondents who left a name and phone number. Starting from the top of the list, I called the respondents to verify that the interview took place. I tried to reach each respondent up to two times, leaving a voicemail if he or she did not answer. If I could not reach a respondent after two tries, I proceeded to the next respondent on the list, continuing until I successfully contacted someone or ran out of respondents on the list. Once I contacted a respondent for a given interviewer, I verified that the interview did indeed take place and concluded the reinterview process for the interviewer.

Each of the interviewers had at least one respondent who left his or her name and phone number. For 26 of the 28 interviewers, I reached the first respondent that I tried and veri-

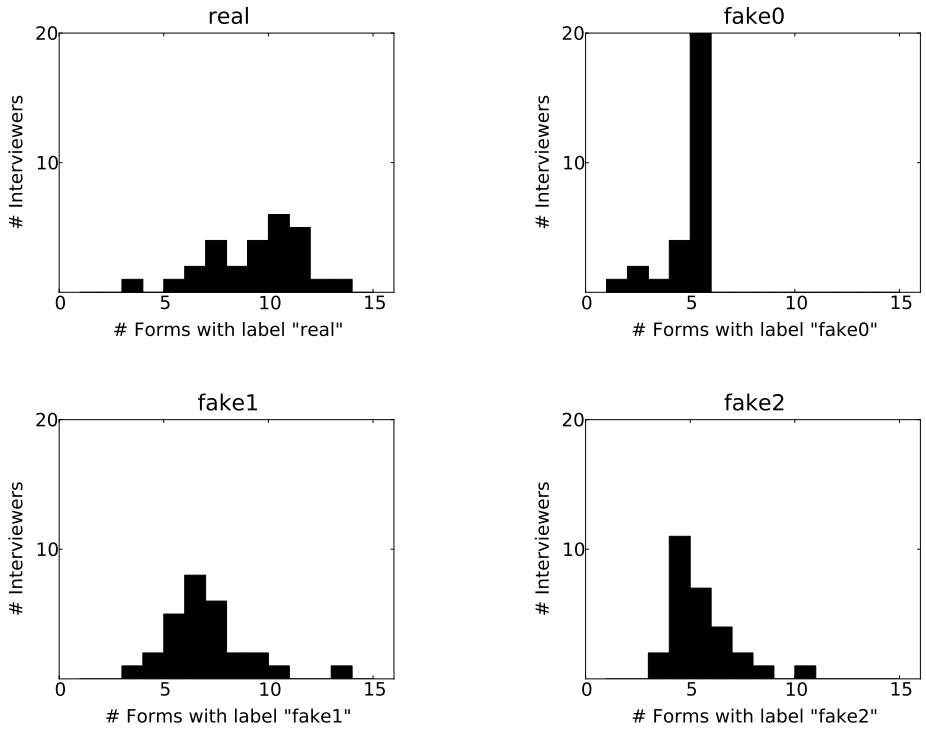


Figure 5.3: Histograms of number of forms per interviewer. Each plot shows a histogram for a different label.

fied that the interview did indeed take place. For the remaining two, there was only one respondent listed, and I was unable to reach that respondent in two phone calls.

The second step that I took to minimize the possibility of curbstoning on the data labeled `real` was to conduct a debriefing at the end of the follow up session in which I invited participants to share any irregularities or fabrication that occurred during the interview period. I stressed how important it was that I know exactly which data is real and which data is fake. I told them that if they informed me of anything, I would keep it confidential and it would not affect their compensation in any way. I also did my best to convey that I would actually be grateful for the help it would provide. I mentioned that if they were embarrassed to tell me something in person, they could email me afterward.

Given this invitation, most participants said that they collected data exactly as they were supposed to. However, a few respondents did share important information with me. Three of them told me that for a total of 8 forms, they collected data on paper during the interview

and entered it into the phone later; three of them told me that for a total of 5 forms, they handed the phone to the respondent to self-administer; and finally, one of them told me that he fabricated 2 forms in order to reach the goal of 10 interviews that I requested. As described in the next section, I removed all 15 of these forms during data cleaning.

Because of the measures I took to ensure data quality, I believe that, with possibly only a few exceptions, there was not a substantial amount of curbstoning in the `real` data. Further evidence for this is given by the positive nature of the results that I will present in the next chapter. That is, one could argue that if there was a large amount of curbstoning in the data labeled `real`, then it would have been more difficult than it was to distinguish it from the data known to be fabricated.

5.6.3 Data cleaning

During data cleaning, forms were removed for various reasons:

- I could not figure out the label for the form because of an error in the code (3 forms from 3 interviewers).
- The interviewer told me that he fabricated forms that were supposed to be real (2 forms from 1 interviewer).
- The interviewers collected data on paper and then entered it into ODK Collect later (8 forms from 3 interviewers).
- The interviewers handed the phone to respondents to self-administer surveys (5 forms from 3 interviewers).
- The respondent was ineligible to complete the survey (15 forms from 10 interviewers).
- The interview was not completed: less than 20 substantive questions were answered (8 forms from 4 interviewers).

This left 704 forms total. I divided these forms into three groups. First, I created a *hold out* group. None of the forms in this group were used for any of the experiments described in this dissertation.[7] To form the hold out group, I performed the following procedure. For each interviewer i and for each label ℓ in $\{\texttt{real}, \texttt{fake}_0, \texttt{fake}_1, \texttt{fake}_2\}$, I looked at all of the forms completed by interviewer i having label ℓ. If the number of these forms was between 0 and 2, I did not remove any of the forms for the hold out group. If it was between 3 and

[7]Later, I will use this group as a test set for publications based on this dissertation work.

Label	Training	Test
`real`	156 (38%)	47 (36%)
$\texttt{fake}_0$	74 (18%)	24 (18%)
$\texttt{fake}_1$	101 (25%)	33 (25%)
$\texttt{fake}_2$	78 (19%)	28 (21%)

Table 5.5: Distribution of labels for training and test sets. There were 409 forms in the training set and 132 forms in the test set.

5, I randomly chose 1 of the forms to be in the hold out group. If it was between 6 and 10, I randomly chose 2 of the forms to be in the hold out group. Otherwise, if it was some number $r > 10$, I randomly chose $\lceil r/3 \rceil$ forms to be in the hold out group.

This procedure gave me a hold out group of size 163. Next, I divided the remaining 541 forms into a training and test set to use for the experiments in this dissertation. To do so, I followed the same procedure that I used to create the hold out group. This gave me a training set of size 409 and a test set of size 132. Table 5.5 shows the number of forms having each label for both the training and test set.

5.7 Chapter summary

In this chapter, I discussed the design and execution of the study using the Study Habits survey. In Section 5.1, I discussed the design considerations that led to the survey, and I gave a general description of the questions in it. In Section 5.2, I gave relevant background on ODK Collect. In Section 5.3, I described the study protocol. I showed how both participants' knowledge of the algorithms and the incentives provided for realistically fabricating data were carefully controlled and varied so that the effect of such knowledge and motivation could be studied. In Section 5.4, I described the structure of the log files that recorded user-interaction traces on ODK Collect. In Section 5.5, I motivated and defined the features that I extracted from the logs. In Section 5.6, I summarized the actual execution of this study.

In the next chapter, I describe an in-depth investigation of the data from the Study Habits survey.

Chapter 6
STUDY HABITS SURVEY: RESULTS

In the previous chapter, I described the design and execution of the research study using the Study Habits survey. In this chapter, I analyze the data collected in that study.

First, before delving into the analysis, I perform an important verification of a central assumption of the study design: that the *better-informed* fabricated data ($\texttt{fake}_2$) is appropriately named. That is, although the purpose of giving feedback to interviewers before Round 2 of the follow up session was to simulate what happens as interviewers learn and adapt to curbstoning detection algorithms, just giving them feedback does not guarantee that they will understand it and react to it. However, in Section 6.1, I provide quantitative evidence that this did indeed happen, thus lending credibility to the conclusions I draw later based on the $\texttt{fake}_2$ data.

After performing this verification, in Section 6.2, I evaluate the performance of the supervised classification algorithms on the Study Habits data. I show that these algorithms obtain up to 92% accuracy for the uninformed fabricated data. Importantly, I show that the supervised algorithms accurately identify not only the uninformed fabricated data, but also the informed fabricated data (89% accuracy) and better-informed fabricated data (87% accuracy). These results support the part of my thesis that posits that curbstoning detection algorithms can work even when interviewers are aware that they are being used and are incentivized to avoid detection.

In Section 6.3, I explore another question posed by my thesis: whether and how much curbstoning detection algorithms can be improved by leveraging user-trace metadata. First, I remove all of the metadata features and show that the performance of the algorithms is worse without the metadata. Next, I delve more deeply into the question of *how much* detail it is necessary to record in the user trace data. By looking at the performance of classifiers using feature sets that require successively more detail in the logs, I argue that more detailed user logs lead to improved performance. Both of these analyses support the part of my thesis that argues that the accuracy of curbstoning detection algorithms can be improved when mobile electronic devices are used during data collection.

Another part of my thesis is that curbstoning algorithms can *robustly* identify fabricated data. In Section 6.4, I examine various aspects of the robustness of the curbstoning detection algorithms. I motivate some possibilities that could be expected in realistic data collection scenarios, such as being unable to rely on timing information, not knowing who the interviewer is for each survey, or not having completely accurate training labels. I show that the supervised algorithms continue to be accurate in these situations, demonstrating the robustness of my approach.

In Section 6.5, I move from supervised classification algorithms to unsupervised anomaly detection algorithms. I test the performance of both form-level and interviewer-level algorithms on the Study Habits data. I show that although the unsupervised algorithms are less accurate than the supervised algorithms, they can still be useful when there is a small amount of fabrication and training data cannot be obtained.

I conclude the chapter in Section 6.6, in which I broaden the discussion from the specific algorithms used to general traits of fabrication in the Study Habits data. Specifically, I measure the correlation of all of the features with fabrication. My analysis shows how much timing-based features dominate the others in predicting fabrication. In order to get a sense for how the other features fit in, I next remove the timing-based features and examine the correlations of the remaining features. Based on my findings, I give examples of certain traits that may tend to characterize fabricated data.

Preliminaries

Before beginning, I define some notation that will be useful throughout the chapter. Let $\mathcal{D}$ be the set of 541 non-hold-out forms defined in the previous chapter. Let $\mathcal{D}^0$ be the subset of $\mathcal{D}$ consisting of the 409 forms in the training set, and let $\mathcal{D}^t$ be the subset of $\mathcal{D}$ consisting of the 132 forms in the test set. In order to test the ability to recognize each of the three types of fake data—$\texttt{fake}_0$, $\texttt{fake}_1$, and $\texttt{fake}_2$—it will be useful to create three data sets from $\mathcal{D}$. Each of these data sets consists of the real data and one of the types of fake data. More precisely, for any $\ell \in \{0, 1, 2\}$ and $z \in \{0, t\}$, let $\mathcal{D}^z_\ell$ indicate the subset of forms in $\mathcal{D}^z$ having a label of either $\texttt{real}$ or $\texttt{fake}_\ell$. So, for example, the notation $\mathcal{D}^0_0$ denotes the set of all forms in the training set having a label of $\texttt{real}$ or $\texttt{fake}_0$, and the notation $\mathcal{D}^t_1$ denotes the set of all forms in the test set having a label of $\texttt{real}$ or $\texttt{fake}_1$,

6.1 Response to feedback during follow up session

The purpose of giving feedback between Rounds 1 and 2 of the follow up session was to simulate what happens as interviewers learn and adapt to curbstoning detection algorithms. However, just because the interviewers received personalized feedback does not mean that they necessarily understood it or reacted to it. In this section, I analyze changes in the data between Round 1 and Round 2 that give quantitative evidence that the interviewers did indeed react to the feedback.

Recall from Section 5.3.3 that the interviewers were sent an email that told them which 3 of the 9 feedback-eligible features were personally most correlated with fabrication, along with the direction of the correlation. Call this group of features the interviewer's *feedback group*. There were 28 interviewers, so there were a total of $28 \times 3 = 84$ interviewer-feature pairs (i, j) that were part of a feedback group.

If the interviewers understood and reacted to this feedback, then for each such pair (i, j), the mean value of feature j for interviewer i should change in the *opposite* direction of the initial correlation. That is, if interviewer i was told that feature j was negatively correlated with fabrication in Round 1, then the average value of feature j should increase in Round 2—and vice versa if the interviewer was told the opposite. The null hypothesis—that interviewers either did not understand or react to the feedback—would imply that the average value of feature j would be just as likely to go up as it would be to go down, regardless of whether the feature was negatively or positively correlated with fabrication.

An examination of the data shows that, in fact, 71 times out of 84, the mean value of j changed in the opposite direction of the initial correlation. Thus, for a two-sided p-value of less than 10^{-10}, the null hypothesis can be rejected. Hence, it appears very likely that interviewers did react to the feedback they were provided between Rounds 1 and 2. Figure 6.1 shows a more detailed graphical illustration of this trend.

As a side note, an interesting feature in the figure is the interviewer who had an extremely high mean value of `mean-select-one-edits` and `max-select-one-edits` in Round 2. Although the number of edits increased so dramatically between Rounds 1 and 2, these features were not a part of his feedback group, which consisted of `num-conditional`, `mean-string-length`, and `median-answer-time`, all of which were negatively correlated with fabrication in Round 1.

A glance through the raw log files of this interviewer illustrates this unusual behavior. For example, in one form, he changed his answer to `more-costly` 13 times in a row, al-

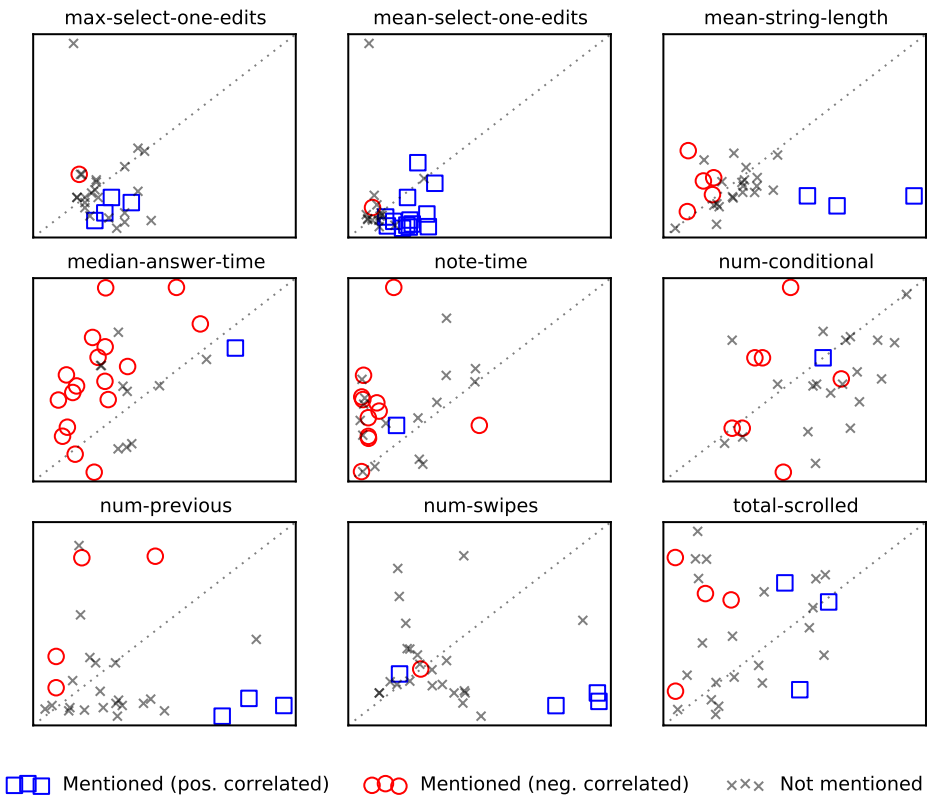


Figure 6.1: Feedback response trends. There is one plot for each feedback-eligible feature j. Within each plot, each marker represents one of the 28 interviewers i. The horizontal position of the marker indicates the mean value of feature j for interviewer i in Round 1 of the follow up session. The vertical position of the marker indicates the mean value of feature j for interviewer i in Round 2. Both axes have the same scale and are stretched to fit the maximum and minimum values of the particular feature. Hence, markers above the diagonal line indicate an increase in mean between the two rounds, and markers below the diagonal line indicate a decrease in mean between the two rounds. The shape of each marker indicates whether feature j was in interviewer i's feedback group, and, if it was, whether feature j was positively or negatively correlated with fabrication. Squares indicate that the feature was in the feedback group and that it was positively correlated with fabrication; circles indicate that the feature was in the feedback group and that it was negatively correlated with fabrication; and crosses indicate that the feature was not in the feedback group. The large fraction of squares that are above the diagonal and circles that are below the diagonal show a significant trend of interviewers reacting to the feedback that they received.

ternating between "yes," "no," and "maybe" with no apparent pattern. Although it is impossible to confirm what caused this behavior, an intriguing speculation is that it was a pacing strategy—either conscious or unconscious—to increase the amount of time spent on questions. That is, in order to rectify a behavior that he knew was getting him caught—spending too little time on each question—he overdid another behavior that was also getting him caught even though he did not know it—the number of edits per question. This points to an advantage of using many different metadata features: *even when interviewers know that their user-traces are being monitored, it may be hard during fabrication to simulate normal behavior along multiple dimensions at once*.

Another interesting pattern evident in Figure 6.1 is the general downward trend in `max-select-one-edits`, `mean-select-one-edits`, and `num-previous`, and the general upward trend in `total-scrolled`. I speculate that the downward trend in the first three features was due to fatigue and the the upward trend in `total-scrolled` was because the high number of interviewers who were told to slow down started scrolling through the prompts more as they simulated reading them to respondents. (Indeed, in $\mathcal{D}$, there are highly significant cross-correlations of 0.40 between `total-scrolled` and `note-time` and of 0.37 between `total-scrolled` and `median-answer-time`.)

6.2 Overall evaluation of supervised algorithms

In this section, I describe my evaluation of the performance of logistic regression and random forest in predicting fabrication on the data sets $\mathcal{D}_0$, $\mathcal{D}_1$, and $\mathcal{D}_2$. Running the algorithms separately for the three data sets allowed me to assess the effect of an increasing level of interviewer knowledge on algorithm accuracy. I show that the supervised algorithms were quite accurate for all three data sets. This result supports the part of my thesis that argues that curbstoning detection algorithms can work even when interviewers are aware they are being used and are incentivized to avoid detection.

First, I followed the procedures outlined in Section 3.2.3 to perform model selection for both logistic regression and random forest. That is, I used the bottom-up greedy feature selection algorithm to get a subset of features for logistic regression, and I performed a search to find the best parameters I and K for random forest (defined in Section 3.2.2). The features selected for logistic regression are listed in Tables C.1 C.2, and C.3, and the parameters (I, K) chosen for random forest were $(256, 16)$ for $\mathcal{D}_0$, $(256, 32)$ for $\mathcal{D}_1$, and

	Logistic Regression					*Random Forests*				
Data	Prec.	Rec.	Spec.	F_1	**Acc.**	Prec.	Rec.	Spec.	F_1	**Acc.**
$\mathcal{D}_0$	0.90	0.71	0.96	0.79	**0.87**	0.88	0.88	0.94	0.88	**0.92**
$\mathcal{D}_1$	0.90	0.82	0.94	0.86	**0.89**	0.88	0.85	0.92	0.86	**0.89**
$\mathcal{D}_2$	0.80	0.71	0.89	0.76	**0.83**	0.85	0.79	0.92	0.81	**0.87**

Table 6.1: Performance of supervised methods on Study Habits data. This table shows the precision, recall, specificity, F_1-score, and accuracy for both logistic regression and random forest.

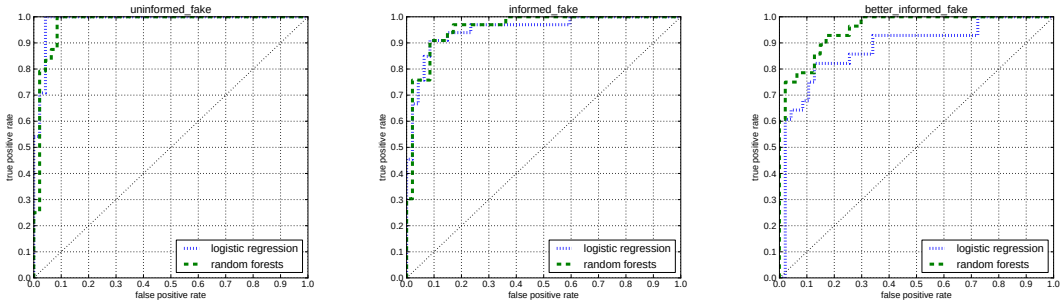


Figure 6.2: ROC curves for supervised methods on Study Habits data.

$(256, 512)$ for $\mathcal{D}_2$.[1]

Once model selection was complete, I trained the two classifiers using the supplied parameters on the training set $\mathcal{D}^0$ and evaluated their performance on the test set $\mathcal{D}^t$. Table 6.1 summarizes the performance measures obtained by both algorithms on all three data sets. It shows that both algorithms perform quite well, achieving accuracies between 83% and 92%. As might be expected, the performance of the algorithms was better when interviewers had less knowledge and were less motivated: the accuracy was highest for the uninformed fabricated data in $\mathcal{D}_0$, second-highest for the informed fabricated data in $\mathcal{D}_1$, and worst for the better-informed fabricated data in $\mathcal{D}_2$. However, *the algorithms were still quite accurate even for the data sets $\mathcal{D}_1$ and $\mathcal{D}_2$ in which interviewers were aware the algorithms were being used, were given specific feedback about how the algorithms worked, and were incentivized to avoid detection.* Random forest performed slightly better than logistic regression.

[1] It is hard to find an intuitive reason that some values of K work better than others. I am including these values only to make it possible to repeat these experiments.

	Logistic Regression					*Random Forests*				
Data	Prec.	Rec.	Spec.	F_1	**Acc.**	Prec.	Rec.	Spec.	F_1	**Acc.**
$\mathcal{D}_0$	0.85	0.46	0.96	0.59	**0.79**	1.00	0.42	1.00	0.59	**0.80**
$\mathcal{D}_1$	0.64	0.21	0.92	0.32	**0.63**	0.60	0.36	0.83	0.45	**0.64**
$\mathcal{D}_2$	0.69	0.32	0.92	0.44	**0.69**	0.50	0.07	0.96	0.13	**0.63**

Table 6.2: Performance of supervised methods on Study Habits response data. This table shows the precision, recall, specificity, F_1-score, and accuracy for both logistic regression and random forest.

Figure 6.2 shows ROC curves for the same classifiers and data sets. These curves show a good set of sensitivity-specificity tradeoffs for both algorithms. They also show that random forest performed much better than logistic regression for $\mathcal{D}_2$.

6.3 Utility of user-trace data

My thesis posits that the accuracy of curbstoning detection algorithms can be improved by leveraging user-trace data recorded by mobile data collection devices. In this section, I provide evidence to support this claim.

First, I describe an evaluation of how well the supervised classification algorithms worked when they were only allowed to use response data. I show that, although they still had some predictive ability, they performed much worse than when they were allowed to use metadata in addition to response data. Second, I show that the amount of detail recorded by the logs is important. I describe an evaluation of the accuracy of classifiers using feature sets that require successively more detail in the logs, and show that more detailed user logs led to improved performance. Both of these analyses point to the importance of user-trace data in identifying curbstoning.

Of the 641 total features, 47 of them are response data features: 33 actual `-response` features (not including responses that do not correspond to interesting survey data, such as `interviewer-id-response` or `phone-response`); 7 `diff-` interviewer-normalized features; and 7 `sqdiff-` interviewer-normalized features. To use logistic regression and random forest on this feature set, I followed the model selection procedure outlined in Section 3.2.3. The features selected for logistic regression are listed in Tables C.4 C.5, and C.6, and the parameters (I, K) chosen for random forest were $(256, 8)$ for $\mathcal{D}_0$, $(256, 512)$ for $\mathcal{D}_1$, and $(256, 2)$ for $\mathcal{D}_2$.

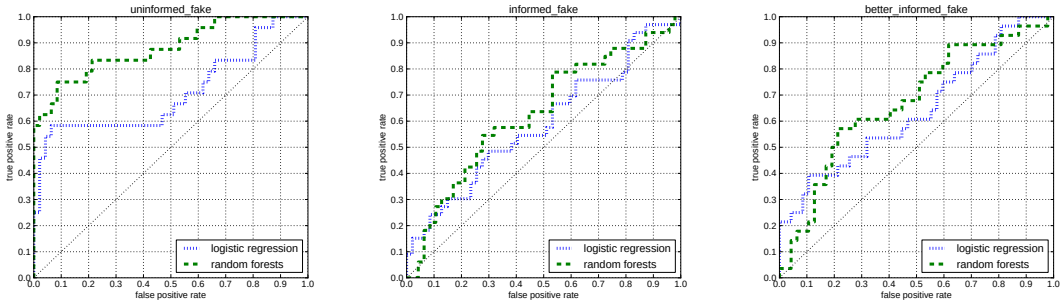


Figure 6.3: ROC curves for supervised methods on Study Habits response data.

Table 6.2 shows the performance measures and Figure 6.3 shows ROC curves for both algorithms on all three data sets. Although the algorithms were still able to predict fabrication with a non-trivial accuracy—especially for $\mathcal{D}_0$—compared to Table 6.1, the performance was much worse. This result shows just how much metadata derived from user-trace data can help in identifying fabricated data.

Given that user-trace metadata can help, one could ask how much detail needs to be recorded in the traces to make the most effective predictions. One possibility is that it is sufficient to record just the time spent on each question. If this were true, it would mean simpler implementations and smaller log files. Thus, it is important to justify the increased complexity that is required to record more detailed user trace logs, with entries for events like edits. I argue here that this level of detail really does help.

Evidence for this is provided by Figure 6.4. This figure shows the error rate of three different logistic regression classifiers, each of which is based on an increasing level of detail in the user trace logs:

- The first classifier, *response only*, is the logistic regression classifier just described that uses a set of response data obtained through bottom-up greedy feature selection. Since this classifier requires *no* user-trace data at all, it acts as a baseline for the other classifiers.

- The second classifier is a logistic regression classifier using the single feature `diff-total-answer-time`. This classifier requires some user-trace data, but only question-level timing data. Specifically, it does not explicitly require detailed user-trace data on question editing. For all three data sets, the error rate of this second classifier is

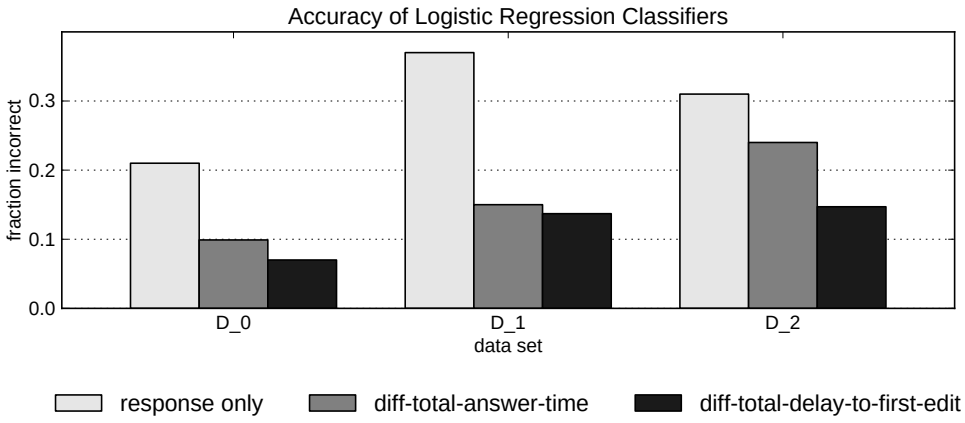


Figure 6.4: Utility of detailed interaction logs. This plot shows the error rates of three logistic regression classifiers, each requiring an increased level of detail in the user trace logs, and each tending to perform better than the last.

much lower than the error rate of the response only classifier.

- The third classifier is a logistic regression classifier that also uses only a single feature. This time, instead of `diff-total-answer-time`, the feature is `diff-total-delay-to-first-edit`. As I will elaborate on in Section 6.6, this feature turns out to be one of the best single predictors of fabrication. It also requires a higher level of detail in the logs than `diff-total-answer-time`: to extract it, for each question, it must be recorded when the user first saw the prompt and when she first made an edit to the response. As shown in the figure, the error rate for this classifier tends to be smaller than for the `diff-total-answer-time` classifier, especially in the data set $\mathcal{D}_2$, for which most interviewers were informed that timing was being used to predict fabrication.

Each successive classifier in this group requires an increased level of detail in the user trace logs, and each successive classifier has a lower overall error rate. This fact supports the argument that user-trace data can substantially improve the accuracy of curbstoning detection algorithms, and that, in general, the more detailed these logs are, the better the algorithms can be.

	Logistic Regression					*Random Forests*				
Data	Prec.	Rec.	Spec.	F_1	**Acc.**	Prec.	Rec.	Spec.	F_1	**Acc.**
$\mathcal{D}_0$	0.79	0.63	0.92	0.70	**0.82**	1.00	0.46	1.00	0.63	**0.82**
$\mathcal{D}_1$	0.67	0.48	0.83	0.56	**0.69**	0.70	0.48	0.85	0.57	**0.70**
$\mathcal{D}_2$	0.59	0.36	0.85	0.44	**0.67**	0.85	0.39	0.96	0.54	**0.75**

Table 6.3: Performance of supervised methods on Study Habits data with no timing-based features. This table shows the precision, recall, specificity, F_1-score, and accuracy for both logistic regression and random forest.

6.4 Robustness analysis

A component of my thesis is that curbstoning algorithms can *robustly* identify fabricated data. In this section, I examine various aspects of the robustness of the curbstoning detection algorithms. First, I explore some realistic scenarios in which certain sets of features may not be available or reliable. In Sections 6.4.1 and 6.4.2, I motivate two such scenarios: a lack of timing information and a lack of interviewer labels for forms (which makes it impossible to use interviewer-normalized features). The best classifiers that I have been able to create relied extensively on both of these types of features. To see how robust supervised classification is in curbstoning detection, I removed these features and evaluated how well the algorithms performed. In Section 6.4.1, I show that although timing is important, reasonably good classifiers can be created when this information is unavailable or unreliable. In Section 6.4.2, I show that my approach is even more robust to unavailable or unreliable interviewer labels. In particular, even if interviewer-normalized features cannot be used, classifiers can be constructed that predict curbstoning with accuracies greater than 80% for all three data sets.

It can be difficult to obtain highly accurate labels on which to train a classifier. For my approach to be most useful, it must be somewhat robust to *label noise*. That is, a small amount of inaccuracy in the labels used to train the classifier must not cause a drastic decrease in predictive accuracy. In Section 6.4.3, I describe an experiment that simulates various levels of label noise and evaluates the resulting loss in accuracy. I show that the supervised algorithms retain a high accuracy even when the label noise is up to 10%.

6.4.1 No timing information

There are many reasons that timing information may be unavailable or unreliable. First, timestamps on phones can change unpredictably as the phones lose power, reset, or connect to towers with inaccurate clocks. This can be an especially pernicious problem in

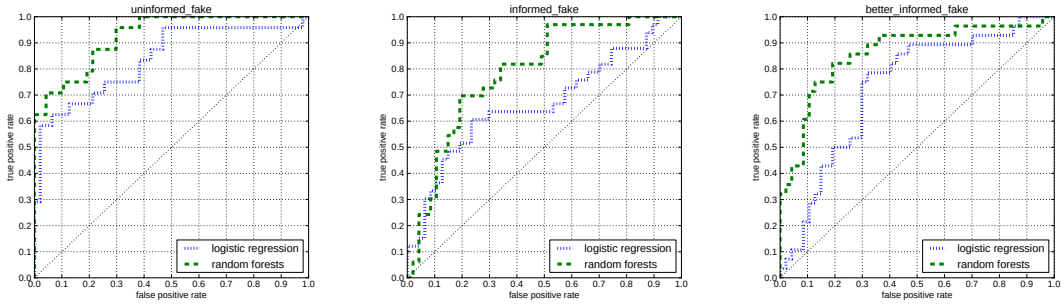


Figure 6.5: ROC curves for supervised methods on Study Habits data with no timing-based features.

low-resource regions that may have less reliable infrastructure. Second, even if timing information is accurate, it may not make sense to use it as a predictor for fabrication. For example, interviewers may gather their data on paper and enter it on a phone later, which may cause data to be entered much faster than if respondents were actually being interviewed during data entry. In some scenarios, it may make sense to restrict this behavior, but in others it may not. For these scenarios, using timing information to look for fabrication might be inaccurate.

In this section, I describe an evaluation of the supervised algorithms when all of the timing-based features are removed. Of the 641 features, 291 of them were derived from the time stamps on the logs, which left 350 features that were not. To use logistic regression and random forest on these 350 features, I followed the model selection procedures that I outlined in Section 3.2.3. The results of the bottom-up greedy feature selection algorithm for logistic regression are shown in Tables C.7, C.8, and C.9, and the parameters (I, K) chosen for random forest were $(256, 32)$ for $\mathcal{D}_0$, $(256, 512)$ for $\mathcal{D}_1$, and $(256, 512)$ for $\mathcal{D}_2$.

Table 6.3 shows the performance measures and Figure 6.5 shows ROC curves for both algorithms on all three data sets. They show that when timing data was removed but other metadata features are not, the performance lied between the response-only classifier and the classifier with full access to all features. Thus, although timing data should certainly be used if it can, fairly accurate classifiers can be created if it is not available or is unreliable.

6.4.2 Lack of interviewer labels

There are many reasons that having interviewer labels on forms might not be possible or reliable. First, interviewers may not be tied to a unique device. Or perhaps, in the case of

	Logistic Regression					*Random Forests*				
Data	Prec.	Rec.	Spec.	F_1	**Acc.**	Prec.	Rec.	Spec.	F_1	**Acc.**
$\mathcal{D}_0$	0.87	0.83	0.94	0.85	**0.90**	0.87	0.83	0.94	0.85	**0.90**
$\mathcal{D}_1$	0.88	0.85	0.92	0.86	**0.89**	0.85	0.85	0.89	0.85	**0.88**
$\mathcal{D}_2$	0.88	0.75	0.94	0.81	**0.87**	0.82	0.64	0.92	0.72	**0.81**

Table 6.4: Performance of supervised methods on Study Habits data with no interviewer-normalized features. This table shows the precision, recall, specificity, F_1-score, and accuracy for both logistic regression and random forest.

surveys in dangerous regions or about sensitive topics, retaining interviewer anonymity is important for its own sake. Finally, maybe interviewer labels can be obtained, but each interviewer performs so few interviews that there is too much noise in the interviewer-normalized features derived from them. In all such cases, using the interviewer-normalized features—which have been crucial to the classifiers described so far in this chapter—may not be possible.

To evaluate how well the algorithms worked without interviewer-normalized features, I removed the 410 interviewer-normalized features from the feature set and evaluated the performance of the resulting classifiers. As before, I followed the model selection procedures that I outlined in Section 3.2.3. The results of the bottom-up greedy feature selection algorithm for logistic regression are shown in Tables C.10, C.11, and C.12, and the parameters (I, K) chosen for random forest were $(256, 16)$ for $\mathcal{D}_0$, $(256, 32)$ for $\mathcal{D}_1$, and $(256, 512)$ for $\mathcal{D}_2$.

Table 6.4 shows the performance achieved by the algorithms without access to the interviewer-normalized features and Figure 6.6 shows the ROC curves. They show that, although the performance was not quite as high as it was when interviewer-normalized features could be used, it was still very good. In particular, both algorithms achieved accuracies of over 80% for all three data sets.

6.4.3 Label noise

When obtaining training data for the supervised algorithms, it can be hard to know that all of the `real` data is actually real. Therefore, it is important that the algorithms be robust to some amount of label noise in the real data. In this section, I describe a simulation that I performed to evaluate how robust the algorithms are to label noise.

I performed three experiments, one for each data set $\mathcal{D}_1$, $\mathcal{D}_2$, and $\mathcal{D}_3$. For each data set,

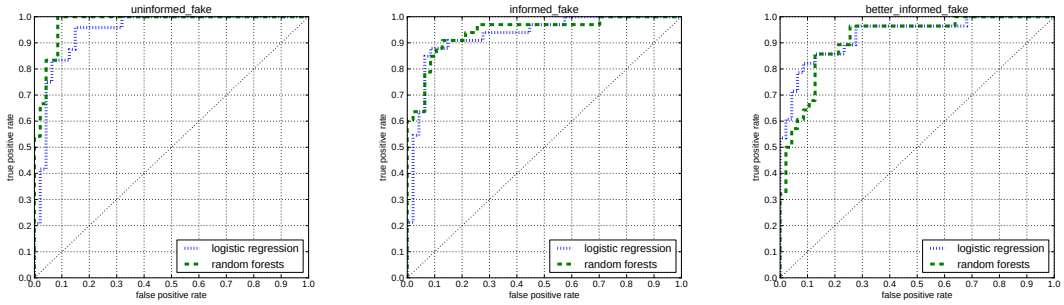


Figure 6.6: ROC curves for supervised methods on Study Habits data with no interviewer-normalized features.

I randomly chose a fraction of the fake data in the training set and switched its label to real. Then I trained logistic regression and random forest classifiers on this (partially mislabeled) training data and evaluated the accuracy on the (correctly labeled) test data. To evaluate how much badly labeled data the algorithms could tolerate, I chose three different values of this fraction: for each value of f in $\{0.05, 0.10, 0.25\}$, I switched the label on exactly the amount of fake data needed to make f fraction of the real data mislabeled. To average out random noise, I repeated this experiment 5 times for each value f.

Figure 6.7 shows the results of this experiment. It shows virtually no difference in error rate when 5% of the real data was mislabeled, a very small difference in error rate when 10% of the real data was mislabeled, and a much larger difference in error rate when 25% of the real data was mislabeled. Thus, the supervised algorithms are robust to at least a moderate amount of label noise.

6.5 Performance of unsupervised algorithms

In this section, I describe an evaluation of the unsupervised anomaly detection algorithms from Section 3.3 on the Study Habits data. Let $\mathcal{D}_{1,2} = \mathcal{D}_1 \cup \mathcal{D}_2$. I evaluated LOCI and SVA on the data sets $\mathcal{D}_0$ and $\mathcal{D}_{1,2}$. (I do not discuss the performance separately for $\mathcal{D}_1$ and $\mathcal{D}_2$ because it was very similar for the two data sets.)

First, I evaluated the ability of LOCI to predict fabrication at the form level. For each of the two data sets $\mathcal{D}_0$ and $\mathcal{D}_{1,2}$, I ran the following experiment once for each value of $p \in \{75, 90, 99\}$: for 100 repetitions, I randomly selected p real forms and $100 - p$ fake

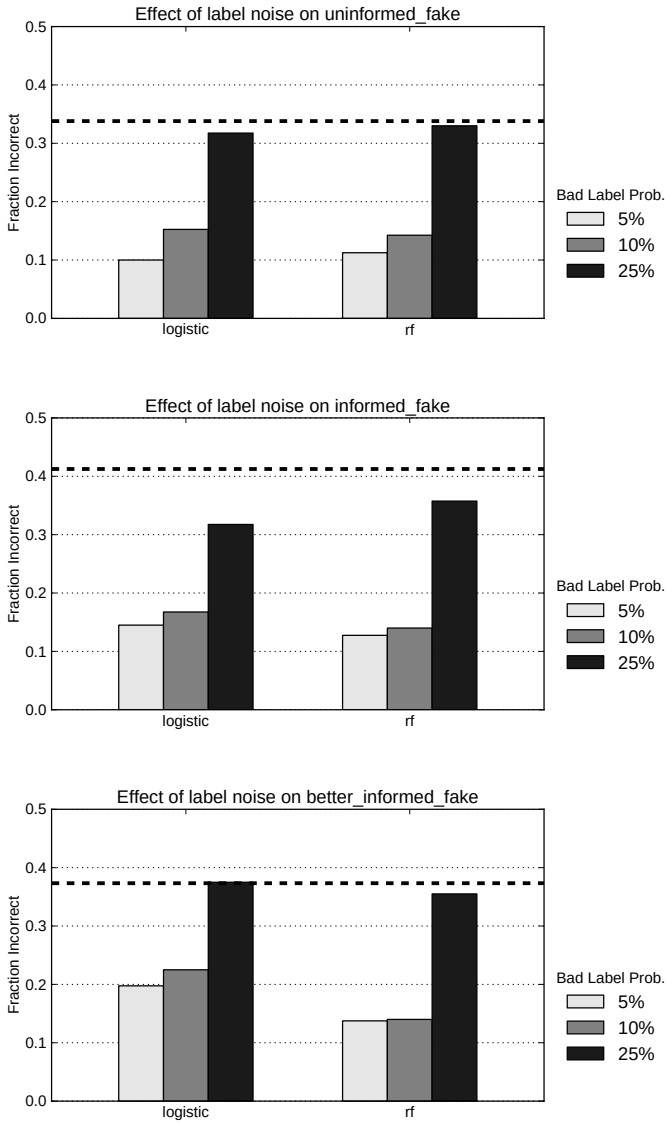


Figure 6.7: Label noise experiment. Each plot corresponds to one of the data sets $\mathcal{D}_0$, $\mathcal{D}_1$, or $\mathcal{D}_2$ and shows the error rate of logistic regression and random forest classifiers for varying levels of mislabeled `real` data. The dotted line shows the baseline accuracy achieved by a classifier that just outputs `real` for each form.

forms. Then I had LOCI compute outlier scores for each of the chosen forms. I used these scores, aggregated over the 100 repetitions, to create ROC curves. I did this for three different feature sets: the feedback-eligible features described in Section 5.3.3, the response

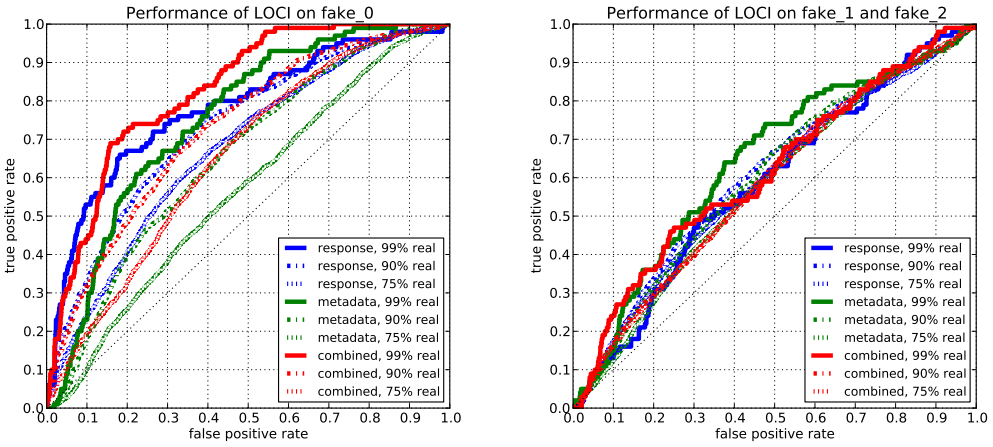


Figure 6.8: Performance of LOCI algorithm on $\mathcal{D}_0$ and $\mathcal{D}_{1,2}$. Each curve is the aggregate over 100 repetitions of sampling 100 of the forms.

data, and a combination of the feedback-eligible features and the response data.

The ROC curves generated from this experiment are shown in Figure 6.8. This figure shows that LOCI tended to predict fabrication for all conditions, but only at a level marginally above chance for the data set $\mathcal{D}_{1,2}$. Thus, unlike the case of the supervised algorithms, it appears that the performance of LOCI got much worse when interviewers gained knowledge and incentive to avoid detection. For $\mathcal{D}_0$, it did better, but the performance depended substantially on the fraction of fabricated data in the data set. When the fraction was low, LOCI did relatively well, and when the fraction was high, LOCI did relatively poorly. The figure also shows that in contrast to the supervised algorithms, when the features were the response data, LOCI did somewhat better than when the features were metadata from user traces. Thus, it seems that for the Study Habits data, the metadata features were grouped into clusters for both the real and fabricated data, whereas the response data features tended to be more clustered for the real data than for the fabricated data.

Next, I evaluated both the univariate and multivariate versions of SVA on the categorical response data from $\mathcal{D}_{1,2}$.[2] SVA is an interviewer-level algorithm, but in the Study Habits data, each interviewer had both real and fabricated data. To work around this, I divided

[2]I did not evaluate an interviewer-level unsupervised algorithm on $\mathcal{D}_0$ because of the small number of forms from each interviewer with the label $\texttt{fake}_0$. I did not evaluate MMA because the number of forms from each interviewer was too small for a reliable χ^2-test.

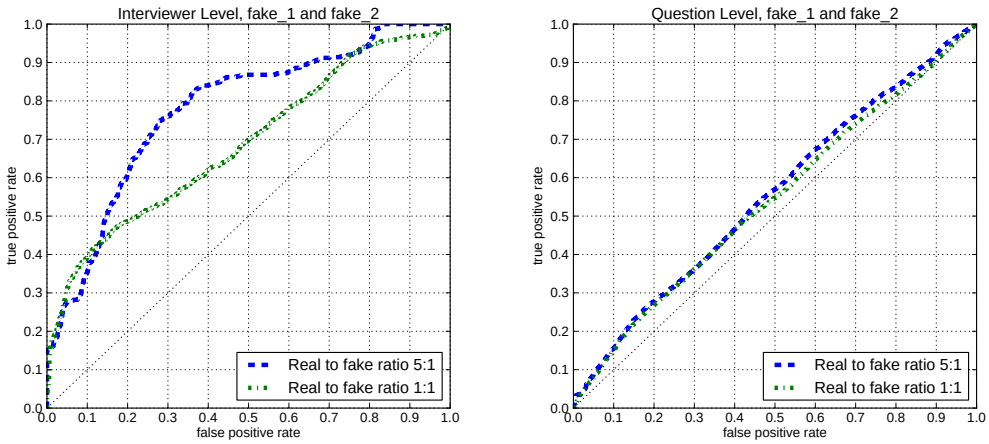


Figure 6.9: Performance of SVA on Study Habits data. Each curve is the aggregate over 100 repetitions. On the left is the performance of the multivariate version of SVA, and on the right is the performance of the univariate version.

each interviewer i into two simulated interviewers i_{real}, who had all of the forms labeled `real` from interviewer i and i_{fake}, who had all of the forms labeled $\texttt{fake}_1$ or $\texttt{fake}_2$ from interviewer i. Similar to the LOCI experiment, I did the following for each value of $p \in \{15, 25\}$: for 100 repetitions, I randomly selected p "real" interviewers and $30 - p$ "fake" interviewers. Then I had SVA compute outlier scores for each of the interviewers (in the multivariate case) and each of the interviewer-question pairs (in the univariate case). I used these scores, aggregated over the 100 repetitions, to create ROC curves, which are shown in Figure 6.9.

This figure shows that SVA did not perform as well for $\mathcal{D}_{1,2}$ as it did for the data in Chapter 4. In particular, it barely performed above chance in the univariate case. The performance in the multivariate case was substantially better, but it did not match the performance of the algorithm on the data in Chapter 4. This discrepancy could be because the interviewers were more careful about the responses they chose for $\mathcal{D}_{1,2}$ since they knew that curbstoning detection algorithms were being employed and that they would be rewarded if they created realistic data. Note that just as in Chapter 4, the performance of the algorithms improved somewhat when the proportion of fabricated data was lowered.

Rank	Feature	r-value
1	`diff-median-answer-time`	-0.75
2	`diff-median-delay-to-first-edit`	-0.73
3	`diff-total-answer-time`	-0.72
4	`diff-total-delay-to-first-edit`	-0.71
5	`diff-more-costly-time`	-0.65
6	`diff-total-time-conditional`	-0.62
7	`diff-advisors-time`	-0.61
8	`diff-tas-time`	-0.60
9	`diff-advisors-delay-to-first-edit`	-0.58
10	`diff-faculty-time`	-0.57

Table 6.5: Top 10 interviewer-normalized features most correlated with $\texttt{fake}_0$.

6.6 Feature correlations

In this section, I broaden the discussion from the specific algorithms used to general traits of fabrication in the Study Habits data. To gain insight into the usefulness of the features, I found how correlated each of the interviewer-normalized features were with forms having the various fake labels. (I focused on the interviewer-normalized features because they typically dominated the other features as predictors for fabrication.) More specifically, for each $\ell \in \{0, 1, 2\}$ let I_ℓ be an *indicator variable* that is 1 if the form has label $\texttt{fake}_\ell$ and 0 otherwise. For each ℓ and interviewer-normalized feature j, I computed the *Pearson product-moment correlation coefficient* [38], or r-value, of j and I_ℓ on the data set $\mathcal{D}_\ell$. The 10 of these features that had the highest value of $|r|$ are shown in Table 6.5 ($\texttt{fake}_0$), Table 6.6 ($\texttt{fake}_1$), and Table 6.7 ($\texttt{fake}_2$).

The most striking feature of Tables 6.5, 6.6 and 6.7 is that they consist entirely of timing-based features. That is, features that are derived from the timestamp data correlate much more strongly with fabrication than do the rest of the features. The non-timing-based features account for 216 of the 410 interviewer-normalized features, but when all of the interviewer-normalized features are ranked by $|r|$, the highest rank for one of these features is only 33 for $\mathcal{D}_0$ (`diff-num-conditional`), 53 for $\mathcal{D}_1$ (`diff-mean-select-one-contiguous-edits`), and 42 for $\mathcal{D}_2$ (`diff-num-conditional`). The direction of the correlation of all of these timing-based features is negative. That is, as expected, *timing is an important predictor of fabrication, and fabricated data tends to be created more quickly*.

It is also interesting to note that the most predictive features in these tables are the interviewer-normalized version of the *form-level* features. That is, features that aggregated timing data

Rank	Feature	r-value
1	`diff-median-delay-to-first-edit`	-0.71
2	`diff-total-delay-to-first-edit`	-0.66
3	`diff-total-answer-time`	-0.65
4	`diff-median-answer-time`	-0.65
5	`diff-more-costly-time`	-0.59
6	`diff-more-costly-delay-to-first-edit`	-0.54
7	`diff-average-work-time`	-0.54
8	`diff-faculty-delay-to-first-edit`	-0.54
9	`diff-more-costly-how-much-delay-to-first-edit`	-0.54
10	`diff-hardest-course-hours-time`	-0.54

Table 6.6: Top 10 interviewer-normalized features most correlated with $\texttt{fake}_1$.

Rank	Feature	r-value
1	`diff-total-delay-to-first-edit`	-0.56
2	`diff-median-delay-to-first-edit`	-0.51
3	`diff-total-answer-time`	-0.48
4	`diff-hardest-course-hours-delay-to-first-edit`	-0.46
5	`diff-faculty-delay-to-first-edit`	-0.44
6	`diff-hardest-course-hours-time`	-0.43
7	`diff-faculty-time`	-0.42
8	`diff-hardest-course-time`	-0.41
9	`diff-advisors-why-not-delay-to-first-edit`	-0.41
10	`diff-total-time-conditional`	-0.41

Table 6.7: Top 10 interviewer-normalized features most correlated with $\texttt{fake}_2$.

across the entire form, like `diff-total-answer-time` and `diff-total-delay-to-first-edit`, did better as predictors than the timing of individual prompt-level features. However, of the prompt-level timing features, the ones that did best in predicting fabrication corresponded to questions that could be expected to require a higher-than-average amount of thought to answer, such as `more-costly`, `average-work`, or `hardest-course`. This suggests that *timing information from hard questions predicts fabrication more accurately than timing information from easy questions*.

A final important feature of these tables is the substantial drop in the correlation of the timing-based features between $\texttt{fake}_1$ and $\texttt{fake}_2$. This fact, which is consistent with the analysis in Section 6.1, suggests that the interviewers realized that timing was an impor-

	fake$_0$			fake$_1$			fake$_2$		
Feature	r	p	k	r	p	k	r	p	k
median-delay-to-first-edit	-0.73	0^*	2	-0.71	0^*	1	-0.51	0^*	2
total-delay-to-first-edit	-0.71	0^*	4	-0.66	0^*	2	-0.56	0^*	1
median-answer-time	-0.75	0^*	1	-0.65	0^*	3	-0.39	0^*	5
total-answer-time	-0.72	0^*	3	-0.65	0^*	4	-0.48	0^*	3
total-time-conditional	-0.62	0^*	5	-0.45	0^*	6	-0.41	0^*	4
min-delay-to-first-edit	-0.53	0^*	6	-0.45	0^*	5	-0.20	0^*	10
total-time	-0.36	0^*	12	-0.44	0^*	7	-0.39	0^*	6
note-time	-0.51	0^*	7	-0.41	0^*	8	-0.13	0.02	13
num-conditional	-0.48	0^*	8	-0.15	0.01	14	-0.26	0^*	7
mean-string-length	-0.37	0^*	11	-0.16	0^*	13	-0.21	0^*	9
min-answer-time	-0.45	0^*	9	-0.30	0^*	10	0.06	0.29	17
mean-select-one-contiguous-edits	0.28	0^*	13	0.35	0^*	9	0.08	0.16	15
num-swipes	-0.25	0^*	14	-0.07	0.20	16	-0.25	0^*	8
total-scrolled	-0.42	0^*	10	-0.20	0^*	12	0.02	0.73	22
max-select-one-contiguous-edits	0.18	0^*	16	0.26	0^*	11	0.05	0.38	19
times-old-response-checked	-0.20	0^*	15	-0.06	0.27	17	-0.08	0.16	16
mean-select-one-non-contiguous-edits	-0.06	0.30	20	0.12	0.03	15	-0.11	0.05	14
num-previous	-0.12	0.04	17	0^*	1.00	22	-0.18	0^*	12
max-select-one-non-contiguous-edits	-0.03	0.60	22	0.03	0.58	20	-0.19	0^*	11
times-why-not-avoided	0.05	0.39	21	-0.04	0.46	18	-0.06	0.29	18
total-hours-on-activities	0.07	0.23	19	-0.04	0.46	19	-0.03	0.60	20
num-activities-more-than-one-hour	0.10	0.08	18	-0.03	0.58	21	-0.02	0.73	21

Table 6.8: Correlations of form-level features. This table lists each of the diff- versions of the 22 interviewer-normalized form-level features. For each feature, and for each label in $\{\texttt{fake}_0, \texttt{fake}_1, \texttt{fake}_2\}$, it shows the r-value of correlation, the 2-sided p-value of the significance of this correlation, and k, which is the rank of that feature's value of $|r|$ compared to the other form-level features. (p-values of less than 0.01 are denoted by 0^*.) Features are listed in increasing order of the average of the k values.

tant clue for the curbstoning detection algorithm and adjusted their behavior accordingly. It is interesting to note, however, that even though 22 of the 28 participants were given feedback on either median-answer-time or note-time, the magnitude of the correlation with fake$_2$ of the top time timing-based feature, diff-total-delay-to-first-edit, remained 0.56. This suggests that *timing might be a robust predictor of fabrication even when interviewers know that it is being used*.

Although timing-based features dominated the other features as predictors of fabrication, other metadata features were useful as well. Table 6.8 shows the correlations of each of the diff- versions of the 22 form-level features with fabrication for each of the three datasets $\mathcal{D}_0$, $\mathcal{D}_1$, and $\mathcal{D}_2$. In the table, each feature j is given a number k for each dataset $\mathcal{D}_\ell$ that indicates the rank, out of all 22 features, of j's value of $|r|$ on $\mathcal{D}_\ell$. The features are ordered in this table by the average value of k. Thus, features toward the top of the table tended to

Rank	Feature	r-value
33	diff-num-conditional	-0.48
56	diff-total-scrolled	-0.42
59	diff-tutors-ord	0.39
61	diff-tutors-why-not-num-non-contiguous-edits	-0.39
65	diff-mean-string-length	-0.37
68	diff-cse-major-num-non-contiguous-edits	-0.35
69	sqdiff-uw-college-num-non-contiguous-edits	0.34
73	diff-uw-college-num-non-contiguous-edits	-0.32
74	diff-intended-major-num-non-contiguous-edits	-0.31
75	sqdiff-cse-major-num-non-contiguous-edits	0.31
76	diff-tutors-why-not-num-contiguous-edits	-0.31
77	diff-major-num-contiguous-edits	-0.30
81	diff-major-num-non-contiguous-edits	-0.29
84	diff-mean-select-one-contiguous-edits	0.28
88	sqdiff-intended-major-num-non-contiguous-edits	0.26
91	diff-num-swipes	-0.25
93	sqdiff-age-years-response	0.24
94	sqdiff-num-conditional	0.24
98	sqdiff-major-num-non-contiguous-edits	0.22
104	sqdiff-tutors-ord	0.20
106	diff-times-old-response-checked	-0.20
107	diff-average-work-ord	0.19
108	diff-volunteer-work-num-contiguous-edits	0.19

Table 6.9: Highly significant correlations with $\texttt{fake}_0$. All non-time-based interviewer-normalized features that have a highly significant correlation with having the label $\texttt{fake}_0$ ($p < 0.001$).

predict fabrication better.

Table 6.8 shows that some of the most correlated non-timing-based features were `diff-num-conditional` (negatively correlated), `diff-mean-string-length` (negatively correlated), `diff-mean-select-one-contiguous-edits` (positively correlated), `diff-num-swipes` (negatively correlated), and `diff-total-scrolled` (negatively correlated). *Thus, the data suggests that curbstoners choose shorter paths through the survey; choose shorter free text responses; make more changes to their data; move back and forth through the survey less; and scroll less.*

To get a sense for all of the significantly correlated non-timing-based features—not just the form-level questions—Tables 6.9, 6.10, and 6.11 show, for each data set $\mathcal{D}_0$, $\mathcal{D}_1$, and

Rank	Feature	r-value
53	diff-mean-select-one-contiguous-edits	0.35
66	diff-max-select-one-contiguous-edits	0.26
92	diff-total-scrolled	-0.20
96	sqdiff-mean-select-one-contiguous-edits	0.19
100	diff-taken-before-num-non-contiguous-edits	0.19

Table 6.10: Highly significant correlations with $\texttt{fake}_1$. All non-time-based interviewer-normalized features that have a highly significant correlation with having the label $\texttt{fake}_1$ ($p < 0.001$).

Rank	Feature	r-value
42	diff-num-conditional	-0.26
44	diff-num-swipes	-0.25
51	sqdiff-average-work-ord	-0.23
67	diff-mean-string-length	-0.21
70	diff-tutors-why-not-num-contiguous-edits	-0.20
76	diff-cse-major-num-non-contiguous-edits	-0.20
85	diff-tas-why-not-num-non-contiguous-edits	-0.19
88	diff-max-select-one-non-contiguous-edits	-0.19

Table 6.11: Highly significant correlations with $\texttt{fake}_2$. All non-time-based interviewer-normalized features that have a highly significant correlation with having the label $\texttt{fake}_2$ ($p < 0.001$).

$\mathcal{D}_2$, each one of the 410 features that has a highly significant ($p < 0.001$) correlation with fabrication.

Table 6.9 contains a lot of correlations, but many of them are expressions of the same underlying phenomenon. For example, four of the features in Table 6.9 involve either the `tutors` or `tutors-why-not` prompt. All of these correlations are caused by the single fact that before the interviewers actually gathered data, they overestimated the amount of help respondents sought from tutors. This caused `diff-tutors-ord` to be high and the number of edits on `tutors-why-not`, which only appeared if the respondent chose "zero times" for `tutors`, to be small. As another example, the large number of correlations of features related to the questions `major`, `intended-major`, and `cse-major`, are caused by the single fact that interviewers overestimated the number of "no" responses to `decided-major`. Once an interviewer made this choice, all three of the following questions are skipped, and thus the number of edits they received tended to be smaller. Note that it is entirely plausible that

interviewers were choosing "no" to `decided-major` not out of ignorance of the population, but out of a desire to save the time needed to fill in the conditionally-appearing followup questions. The fact that these response data features were important in $\mathcal{D}_0$ but not $\mathcal{D}_1$ or $\mathcal{D}_2$ suggests that *as interviewers gain experience and motivation to fabricate data well, response data becomes a less important indicator of fabrication.*

Tables 6.9, 6.10, and 6.11 show that form-level features—such as `diff-num-conditional`, `diff-mean-string-length`, `diff-mean-select-one-contiguous-edits`, `diff-num-swipes`, and `diff-total-scrolled`—tended to dominate the prompt-level features, apart from the ones previously mentioned. The fact that most of the features in the tables are form-level is even more striking when one recalls that there are 209 prompt-level features, compared to only 22 form-level features. Thus, the data suggests that *to predict fabrication, it is better to use aggregated form-level features than individual prompt-level features*. I discuss this idea further in Section 7.1.2.

Finally, it is interesting to note that `sqdiff-average-work-ord` was negatively correlated with fabrication in $\mathcal{D}_2$ (Table 6.11). This means that once interviewers had fabricated a lot of data, they tended to deviate *less* from the mean for `average-work` when they were fabricating data. This is consistent with the widely held *low data variance* hypothesis [21, 61, 95, 101] described in Section 2.1.

I conclude this section with an illustration of how observations in this section can be leveraged to make judgements about data quality if other information is not available. In Section 5.6.2, I suggested that some of the data labeled `real` could actually be fabricated. Although I took steps to reduce the chance of this occurring, based on the insights described in this chapter, I believe that there may be at least one or two forms that were labeled `real` but actually fabricated.

Figure 6.10 gives a graphical illustration of why I believe one of the forms labeled `real` is actually fabricated. During cross-validation on $\mathcal{D}_0$, this form was determined to be a false positive. However, a more in-depth investigation of this form's features calls this judgement into question. The figure shows the values of the `note-time`, `total-answer-time`, and `total-scrolled` features for that form compared to the values from all of the other forms from the same interviewer. The plot makes it clear what an outlier this particular form is: `note-time` is more than 4 times lower than the next lowest value; `total-answer-time` is more than 2 times lower than the next lowest value; and `total-scrolled` is more than 20 times lower than the next lowest value. With the exception of the `note-time` feature from one other form, the feature values from all of the other forms are much more tightly

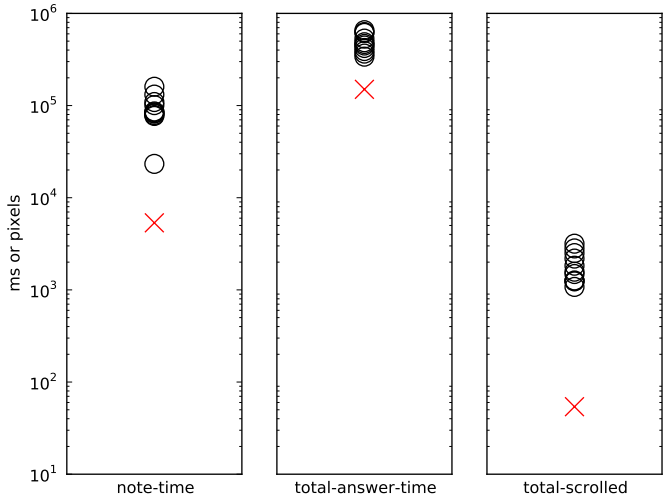


Figure 6.10: Suspicious survey labeled `real`. The feature values for the form in question are indicated by a red X. The feature values from the other forms from the same interviewer are indicated by black circles. Note the logarithmic scale on the y-axis.

clustered. Since low values of `note-time`, `total-answer-time`, and `total-scrolled` correlate strongly with fabrication, I believe that this offers strong circumstantial evidence that this form was not created according to the established protocol for the `real` forms. It is impossible to know the truth without being able to follow up with the interviewer, but information like this forms a strong basis for investigating the quality of data more thoroughly.

6.7 Chapter summary

In this chapter, I described the results from the Study Habits survey. First, I showed that when I gave interviewers specific information on how the curbstoning detection algorithms worked, they changed their behavior in a way that suggests that they understood this information. Second, I showed that—even when interviewers knew about the algorithms and were incentivized to avoid detection—supervised classification algorithms could detect curbstoning with an accuracy of up to 87%. Third, I showed that the user trace information from ODK Collect boosted the accuracy of the algorithms by up to 25%. Fourth, I showed that these algorithms continued to perform well in several realistic sce-

narios, such as when it is impossible to rely on timing information, when interviewer-normalized features could not be used, or when there is noise in the training data labels. Fifth, I showed that the unsupervised anomaly detection algorithms could find fabricated forms with an accuracy substantially above chance when the amount of fabrication was small. Sixth, and finally, I broadened the discussion by analyzing what traits tended to characterize fabricated data.

Chapter 7
CONCLUSION AND FUTURE WORK

My thesis was:

> Supervised classification and unsupervised anomaly detection algorithms can accurately and robustly identify curbstoning, even when interviewers are aware that the algorithms are being used, have some knowledge of how they work, and are incentivized to avoid detection. Furthermore, the accuracy of these algorithms can be improved by using user-trace data recorded by mobile electronic devices during data collection.

In this dissertation, I provided evidence to support this thesis, of which there were four distinct parts.

The first part of my thesis was that the algorithms can identify curbstoning *accurately*. I presented evidence supporting this part in Chapters 4 and 6. In Chapter 4, I showed, using a data set from an East African community health program, that supervised classification algorithms could identify fabricated data at the form level with an accuracy of up to 81% and at the interviewer level with an accuracy of up to 97%. I also showed, by using ROC curves, that the anomaly detection algorithms LOCI, MMA, and SVA could identify curbstoning at a level substantially above random chance. In Chapter 6, I showed, using data from my Study Habits survey, that supervised classification algorithms could identify fabricated data at the form level with an accuracy of up to 92%. I also showed, by again using ROC curves, that the algorithms LOCI and SVA could identify curbstoning at a level substantially above random chance.

The second part of my thesis was that the algorithms can identify curbstoning *robustly*. I presented evidence supporting this part in Chapter 6. I showed that the curbstoning algorithms continued to perform well in several realistic scenarios: when the data contained no timing information, I showed that the accuracy of the supervised classification algorithms was still as high as 82%; when the data was not tagged by the interviewer collecting the data, I showed that the accuracy of these algorithms was still as high as 90%; and when the data had noisy labels—with 10% of the data labeled `real` actually being `fake`—I showed

that the accuracy of these algorithms was still as high as 87%.

The third part of my thesis was that the algorithms can identify curbstoning even when interviewers are aware that the algorithms are being used, have some knowledge of how they work, and are incentivized to avoid detection. I presented evidence supporting this part in Chapter 6. I measured the ability of the algorithm to identify the *informed fake* data $\texttt{fake}_1$ and *better-informed fake* data $\texttt{fake}_2$. I obtained these data sets through a novel experimental design, described in Chapter 5, in which I informed interviewers that curbstoning detection algorithms were being used, provided personalized information about how the algorithms worked, and gave them a monetary reward for fabricating data well. I showed that—even when the algorithms were challenged in this way—they were able to identify fabricated data with an accuracy of up to 87%.

The fourth part of my thesis was that the accuracy of the curbstoning detection algorithms can be improved by using user-trace data recorded by mobile electronic devices. I presented evidence supporting this part in Chapter 6. I compared the performance of the supervised classification algorithms when they were allowed to use user-trace metadata to when they were not. I showed that using this user-trace metadata boosted the accuracy by up to 25%.

I made two contributions in this dissertation that do not pertain to my thesis directly. First, in Chapter 2, I summarized and synthesized related work on detecting curbstoning. From a set of 13 papers, I extracted a set of 15 traits that have been suggested as characterizing fabricated data. This summary can help to inform future investigation into curbstoning. Second, in Chapter 3, I developed two novel anomaly detection algorithms, MMA and SVA. Besides being able to detect curbstoning, these algorithms may have other uses. I discuss this possibility more in Section 7.2.2.

I conclude my dissertation in this chapter. In Section 7.1, I provide some recommendations on how to use the algorithms that I have presented in this dissertation. I discuss what kind of practical impact can be expected if they are used; I provide guidance on how to extract a set of useful features and how to choose which algorithm to use; and I discuss ethical considerations that should be kept in mind when using these algorithms. In Section 7.2, I suggest directions for future research based on what I have presented here. In Section 7.3, I provide final remarks.

7.1 Usage and ethics

In this section, I move the discussion from the academic to the practical: given that my algorithms *can* be used to detect fabrication, *how* should they by used? First, in Section 7.1.1, I consider the question of what benefit can be expected if they are used; second, in Section 7.1.2, I provide guidance on what to consider when choosing which features the algorithms should use; and third, in Section 7.1.3, I discuss some ethical considerations regarding the use of these algorithms.

7.1.1 Potential impact

There may be several reasons to use a curbstoning detection algorithm. For example, in the community health worker context that I described in Section 4.1, a supervisor might want to detect curbstoning to ensure that important home visits are actually occurring. Here, the goal may be more focused on ensuring operational effectiveness than on maintaining data quality *per se*. Usually, however, the reason to use a curbstoning detection algorithm is to ensure that estimates obtained from the data are accurate. In verbal autopsy, for example, the goal is to obtain accurate population-level estimates of cause-of-death frequencies. These estimates may be used for important, and costly, resource allocation decisions. Thus, in this case, supervisors might be interested in using curbstoning detection algorithms to ensure the accuracy of these estimates.

Given that a supervisor's ultimate goal is to ensure that the estimates from the survey are accurate—and not merely to identify curbstoning—the decision of whether to use a curbstoning detection algorithm depends on at least two factors. First, it depends on how prevalent curbstoning is. If it is a large survey and only 0.05% of interviewers fabricate data, then detecting curbstoning may not be particularly important.[1] If, on the other hand, it is a small survey and 30% of the interviewers are fabricating data, then detecting curbstoning may be much more important. Second, the decision of whether to use a curbstoning detection algorithm depends on how accurate the algorithm is. In this dissertation, I have shown that supervised classification algorithms can achieve accuracies higher than 90%. Of course, this number may vary based on the context; the higher it is, the more helpful it is to use a curbstoning detection algorithm.

To give an idea of how these tradeoffs interact, consider a hypothetical example based on results that I reported in Chapter 4. In the survey from that chapter, one of the questions

[1] However, even if the rate of fabrication is currently low, it may still be useful for interviewers to know that effective curbstoning detection algorithms are being used; this knowledge may prevent them from fabricating more data.

Fabrication rate \ True positive rate	**0.10**	**0.20**	**0.30**	**0.40**	**0.50**	**0.60**	**0.70**	**0.80**	**0.90**
0.05	0.00	0.00	0.00	0.01	0.01	0.01	0.01	0.01	0.01
0.10	0.00	0.01	0.01	0.01	0.01	0.02	0.02	0.02	0.03
0.15	0.00	0.01	0.01	0.02	0.02	0.03	0.03	0.04	0.04
0.20	0.01	0.01	0.02	0.02	0.03	0.03	0.04	0.05	0.05
0.25	0.01	0.01	0.02	0.03	0.03	0.04	0.05	0.06	0.07
0.30	0.01	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08
0.35	0.01	0.02	0.02	0.03	0.04	0.05	0.07	0.08	0.09
0.40	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.09	0.10
0.45	0.01	0.02	0.03	0.04	0.05	0.06	0.08	0.10	0.12
0.50	0.01	0.02	0.03	0.04	0.05	0.07	0.08	0.10	0.13

Table 7.1: Hypothetical estimate improvements of using curbstoning detection algorithm. These numbers are from an analysis that assumes, hypothetically, that the real prevalence of a disease is 0.04 and that the prevalence reported in fabricated data is 0.35. For various levels of data fabrication and true positive rates, it shows the difference in the disease prevalence estimate obtained in two ways: (1) without using a curbstoning detection algorithm and (2) using a curbstoning detection algorithm to remove all discovered fabricated responses.

was "Does anyone in the house have diarrhea?" In the data set $\mathcal{A}_\ell$, the response distribution to this question varied dramatically between the real and fake data: the interviewers gathering real data reported a prevalence of diarrhea of 0.04, whereas the interviewers fabricating data reported a prevalence of 0.35.

Suppose that a supervisor is in charge of a survey that measures, among other things, the prevalence of diarrhea in a region. She is worried about curbstoning, and she believes that—as occurred in my data—curbstoners might misreport the prevalence by up to 0.3. What kind of improvement in accuracy can she expect for various levels of curbstoning prevalence and algorithm accuracy? The answer, for this example, is shown in Table 7.1. It assumes that all fabricated data that was identified was removed, and it shows the difference in diarrhea prevalence estimates between before the fabricated data was removed and after it was removed. It shows, for example, that it may not be worth using a curbstoning detection algorithm if the prevalence of curbstoning is less than 30% and the true positive rate of the algorithm is less than 20%. However, for higher—but still realistic—values of the curbstoning prevalence and true positive rate, the situation may be different. For example, if the fabrication rate is 0.25 and the true positive rate of the curbstoning detection algorithm is 0.90, then the estimate of prevalence would change by 0.07. Considering that

the true prevalence of diarrhea in this example is only 0.04, this improvement could have major consequences.

Note that Table 7.1 should not be considered to be an authoritative guide on whether to use curbstoning detection algorithms. To get concrete numbers, I needed to rely on a lot of somewhat arbitrary assumptions. Nevertheless, it illustrates the type of thinking that can be useful when deciding whether to use a curbstoning detection algorithm. Other considerations, such as budget and time limits, will depend heavily on the survey organization's context.

7.1.2 Practical recommendations

Once a supervisor in a survey organization decides to use a curbstoning detection algorithm, there are two questions that she needs to consider: first, which algorithm—or set of algorithms—to use and second, which features to extract for the algorithm.

Choice of algorithm

When considering the choice of algorithm, it can be helpful to recall the tradeoffs of the different algorithm types that I discussed at the end of Section 3.1. There, I classified curbstoning detection algorithms along three binary dimensions: *supervised* vs. *unsupervised*, *form-level* vs. *interviewer-level*, and *univariate* vs. *multivariate*. Each of these dimensions presents a tradeoff. Supervised algorithms are more accurate than unsupervised algorithms, but they require labeled training data—which might be prohibitively time consuming or expensive to obtain. Interviewer-level algorithms are more accurate than form-level algorithms, but they do not have the flexibility to detect when interviewers fabricate only a small amount of the their data. Multivariate algorithms are more accurate than univariate algorithms, but they do not have the flexibility to detect when interviewers fabricate the answers to only some questions. All three of these tradeoffs were apparent in my analyses in Chapters 4 and 6.

Thus, my recommendation to this supervisor depends on what kind of resources she has available and what kind of fabrication she believes may be occurring. If she has the resources to get labeled training data, perhaps through a "fake data party" like I described in Section 4.2.1 and in Section 5.3, my recommendation is to use one of the supervised form-level algorithms from Section 3.2: either logistic regression or random forest. Logistic regression is implemented in many statistical packages and is widely-known; thus, it might be the natural choice. However, there are two advantages of random forest. First, it was more accurate on my data. Second, the model selection procedures that I outlined in Section 3.2.3 were much faster for random forest than for logistic regression. Thus, if

the supervisor has access to a library implementing random forest, I recommend it over logistic regression.

Regardless of whether the supervisor can get labeled training data, I recommend using one of the interviewer-level unsupervised algorithms defined in Section 3.3.2: either MMA or SVA. If she cannot get labeled training data, then algorithms like MMA or SVA may provide the only available automatic approach to detect fabrication; if she can, then they can be used as a complement to supervised algorithms. As I showed in Section 4.4, the unsupervised algorithms can find compelling interviewer-level anomalies in data that may indicate curbstoning, misunderstanding, or even true anomalies in the data such as disease outbreaks. This capability is useful regardless of whether supervised techniques can be used.

I recommend using both the univariate and multivariate versions of MMA and SVA together. The univariate versions can highlight anomalous response distributions to particular questions, and the multivariate versions can highlight interviewers who have anomalous data across many questions. Both MMA and SVA performed similarly, so which to choose is mostly a matter of personal preference. However, it should be kept in mind that SVA does not take into account the number of forms per interviewer; thus, if this number varies significantly from interviewer to interviewer, MMA may be the better choice. LOCI can also be used to identify form-level anomalies, but its slow running time (Table 3.3) makes it less practical.

In all cases, I recommend that my algorithms be used as a supplement to more traditional data-quality control measures, such as numeric outlier detection (`UFU` algorithms in the parlance of Section 3.1) and manual inspection of the data. These older techniques can discover data quality issues that my algorithms cannot, such as isolated data-entry errors or out-of-range values.

Choice of features

Once the supervisor chooses which algorithm, or set of algorithms, to use, the next choice she must make is which *features* the algorithm will use. MMA and SVA only work for categorical response data; thus, for them, the choice is clear: use categorical response data. The supervised algorithms work for categorical and numeric response data and metadata; for them, the choice is more involved. In Section 5.5, I described a large set of features that I extracted from the Study Habits data. Not all of these features were equally useful. Based on my analysis of feature correlations with fabrication in Section 6.6, I have the following recommendations for feature extraction.

By far the most useful features were the timing-based ones. Of these, the *form-level* features, such as the median time spent on each answer (`median-answer-time`) or the median delay to the first edit (`median-delay-to-first-edit`), were the most useful. These features have the advantage that they generalize easily from one survey to the next. Thus, my strongest recommendation is to extract these features if possible. Other form-level features that proved to be particularly useful were the number of conditionally-appearing questions that were answered (`num-conditional`), the mean number of edits per question (`num-edits`), the amount scrolled (`total-scrolled`), and the number of swipes (`num-swipes`). These features are also quite general; I recommend extracting them if possible.

Some of the features that I discussed in Section 5.5—such as the number of times a "why not?" question was avoided (`times-why-not-avoided`) and the total hours spent on activities—were specific to the Study Habits survey. How these features would generalize to a different survey is not immediately clear. However, it turned out that these features were not particularly predictive of fabrication. Therefore, my recommendation is to avoid form-specific features like these. Doing so should not significantly affect the accuracy of the algorithms, and by avoiding form-specific features, it makes the process of feature extraction simpler.

7.1.3 Ethical considerations

Curbstoning detection algorithms can improve data quality, but they cannot replace careful supervision. At their core, all of the curbstoning detection algorithms in this thesis find data that does not fit an expected model. For the supervised algorithms, this model is learned from training data; for the unsupervised algorithms it is based on prior expectations. In both cases, it is an approximation of what high-quality data looks like, but—like all models—it is imperfect. Data fabrication is a serious accusation, especially when someone's reputation or employment is at stake. Therefore, *the predictions of a curbstoning detection algorithm should never be used as the sole basis for any decision.*

Instead, the curbstoning detection algorithms that I have described should be seen as one of many tools that a supervisor has at her disposal to ensure the effectiveness of the survey. If the algorithms detect unusual data, all possible causes should be investigated—including misunderstanding, miscommunication, data entry error, poorly-designed questions, and true anomalies in the data. If, after investigating other potential causes, curbstoning does indeed seem to be occurring, supervisors should attempt to obtain a deep understanding of the evidence before approaching an interviewer to discuss the possibility of fabrication. Doing so will help to ensure that these situations are handled as fairly as possible.

7.2 Future work

The research that I have presented in this dissertation leaves open a variety of directions for future work. In this section, I discuss five categories of potential future work: performing larger studies in different contexts, exploring other applications of the unsupervised algorithms, implementing and exploring more sophisticated user-trace logging, building effective interfaces, and investigating supervisor tuning.

7.2.1 Performing larger studies in different contexts

As I mentioned in Section 1.4, because my data sets are not large, it is too early to conclude without further evaluation that my techniques generalize widely. One possibility, for example, is that because the interviewers for the Study Habits survey only interviewed 10 people, they did not have enough experience to fabricate data realistically. An important direction of future work is to evaluate my algorithms on a larger scale and in different contexts.

Such studies will help to assess whether my results generalize, and they may have other advantages as well. For example, often the best way to improve the performance of a supervised classification algorithm is simply to obtain more training data [7]. Therefore, the performance of curbstoning algorithms may improve when larger data sets are used. As another example, larger data sets may make it possible to explore new types of algorithms, such as supervised, interviewer-level, univariate algorithms (`SIU` algorithms—see Section 3.1 for a further explanation).

7.2.2 Exploring other applications of the unsupervised algorithms

The novel algorithms MMA and SVA that I developed in Section 3.3.2 may be useful in ways beyond merely detecting curbstoning.[2] These algorithms could be used, for example, to detect disease outbreaks or migration patterns. They could use other natural data aggregation units besides the interviewer. In a large survey, for example, high-level managers could use the algorithms to find anomalous *supervisor-level* distributions. With this information, they might be able to find systematic data quality problems resulting from supervisor misunderstanding. Or, a ministry of health could use the algorithms to find anomalous *hospital-level* distributions in disease prevalence data. With this information, they could gain insight into geographical health patterns.

[2] In fact, SVA has already been shown to be useful in a very different context. As mentioned in Section 3.3.2, this algorithm is similar to the *Tariff Method*, which was used to predict the cause of death from verbal autopsy data [63].

7.2.3 Implementing and exploring more sophisticated user-trace logging

In Chapter 6, I showed that features extracted from user-trace logs could be used by curbstoning algorithms to improve their accuracy. There are many possible ways to explore this direction further. First, the potential of more sophisticated features can be explored. GPS data is a natural candidate. I could not evaluate it for my study because data fabrication always occurred in the same physical location. However, if larger evaluations are performed in more realistic survey environments, GPS could prove to be an important signal for data quality. Other possibilities for advanced user-trace data include audio signals (perhaps compressed into summary measures, like average volume level), video signals, or accelerometer data. Note that the use of any of these signals comes with privacy concerns that should be considered carefully.

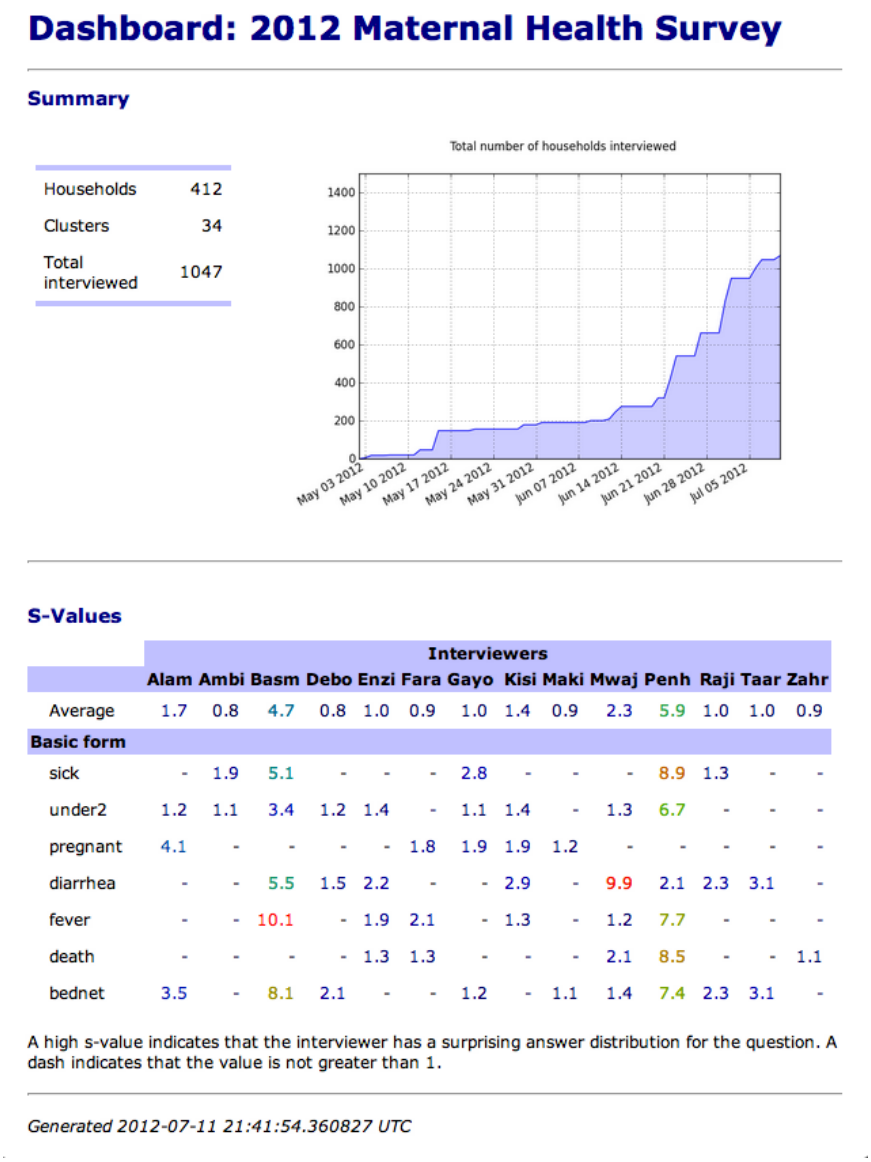
A version of the user-trace logging instrumentation that I implemented will be available in version 1.2 of ODK Collect [86]. There are other applications of these logs that merit further exploration. As an example of how these logs can be used for applications outside of curbstoning detection, Hartung used similar logging instrumentation (which inspired my own) to learn important information about a research study that he was running in India [52]. While collecting data for the study, he observed several unexpected patterns in the log data. His observations inspired him to contact supervisors who were more closely involved with the data collection. After discussing the patterns with these supervisors, he learned that the study was not being conducted as he intended. He would not have been able to learn this important information without the logs.

There are other ways user-trace logs may prove to be useful. They could, for example, be used to investigate usability issues in the design of data-collection software. Researchers may be able to use them to increase data entry efficiency or to reduce data entry error.

7.2.4 Building effective interfaces

As I mentioned in the introduction, my work leaves open how to report the output of curbstoning detection algorithms most effectively. In this realm, there are a number of interesting survey methodology and data visualization questions. What should the threshold be to report suspicious data to a supervisor? How should this data be presented initially? How can reporting redundant information be avoided? What kind of tools should be built to allow the supervisor to investigate why data is being flagged as suspicious? These questions all deserve in-depth exploration.

As a preliminary idea of how to report the output of the algorithms, Figure 7.1 shows a mockup of a dashboard that displays the results of unsupervised algorithms like MMA



Dashboard: 2012 Maternal Health Survey

Summary

Households	412
Clusters	34
Total interviewed	1047

S-Values

	Interviewers													
	Alam	Ambi	Basm	Debo	Enzi	Fara	Gayo	Kisi	Maki	Mwaj	Penh	Raji	Taar	Zahr
Average	1.7	0.8	4.7	0.8	1.0	0.9	1.0	1.4	0.9	2.3	5.9	1.0	1.0	0.9
Basic form														
sick	-	1.9	5.1	-	-	-	2.8	-	-	-	8.9	1.3	-	-
under2	1.2	1.1	3.4	1.2	1.4	-	1.1	1.4	-	1.3	6.7	-	-	-
pregnant	4.1	-	-	-	-	1.8	1.9	1.9	1.2	-	-	-	-	-
diarrhea	-	-	5.5	1.5	2.2	-	-	2.9	-	9.9	2.1	2.3	3.1	-
fever	-	-	10.1	-	1.9	2.1	-	1.3	-	1.2	7.7	-	-	-
death	-	-	-	-	1.3	1.3	-	-	-	2.1	8.5	-	-	1.1
bednet	3.5	-	8.1	2.1	-	-	1.2	-	1.1	1.4	7.4	2.3	3.1	-

A high s-value indicates that the interviewer has a surprising answer distribution for the question. A dash indicates that the value is not greater than 1.

Generated 2012-07-11 21:41:54.360827 UTC

Figure 7.1: Mockup of survey dashboard using unsupervised algorithm.

or SVA. This dashboard displays the m_j^i values reported by the MMA algorithm for each interviewer–question pair. These values are color coded to indicate severity to highlight the most anomalous data. One could imagine that these numbers are clickable; once a supervisor clicks on one, she could see response distribution graphs, such as the one shown in Figure 7.2, that would help her understand how exactly the data is anomalous.

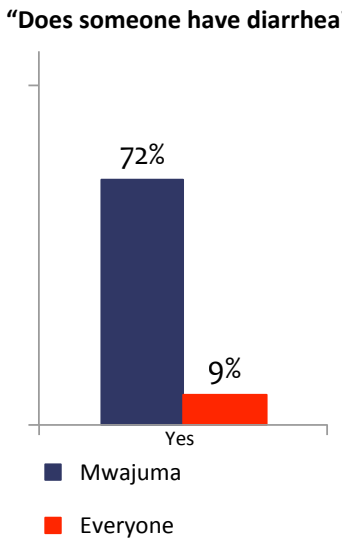


Figure 7.2: Distribution details from dashbaord mockup.

One could imagine similar dashboards for the form-level supervised and unsupervised algorithms. Predictions of fabrication or other data quality issues could be integrated into a system such as ODK Aggregate [53] or CommCareHQ [34], that allows supervisors to browse form-level data.

7.2.5 Investigating supervisor tuning

A supervisor may have important knowledge about specific data quality issues in her survey. For example, she might have prior beliefs about response distributions or know that certain questions are especially likely to be fabricated. An interesting research direction is to investigate how to integrate this knowledge into software systems that use curbstoning detection algorithms. There are two important subquestions: first, how can the *algorithms* be modified to take into account supervisor feedback; and second, how can *interfaces* be developed that support this in a natural way. On the algorithms side, one could imagine, for example, that the model selection algorithms described in Section 3.2.3 could be modified to include supervisor input. On the interfaces side, one could imagine that a dashboard like the one shown in Figure 7.1 could be modified so that a supervisor can click on specific questions to add priors for the distributions.

One could also imagine incorporating supervisor tuning as a way to combine supervised and unsupervised algorithms. To start, a curbstoning detection system could use unsu-

pervised methods to alert supervisors to suspicious patterns. As she follows up on these alerts, she would provide feedback to the system regarding the true cause of the suspicious data. The system could then use this feedback as labels to train a supervised classification algorithm. Indeed, one could frame this problem as an instance of *active learning* [100] in which labels can be obtained, but at a cost associated with the follow ups.

7.3 Final remarks

In this dissertation, I have developed and validated general algorithmic techniques to detect interviewer data fabrication. This issue is one of many that can negatively affect the quality of data coming from surveys in developing countries—surveys that may be the only way to obtain critical health and economic data. The algorithmic and computing technology available to detect these types of problems is growing more prevalent every day. It is imperative that we, as researchers and developers, continue to explore how it can be used most effectively.

Acknowledgments

All of the work presented here was a collaborative effort. I thank my co-authors—Gaetano Borriello, Brian DeRenzi, Abraham Flaxman, Anna Karlin, and Neal Lesh—for their critical contributions.

I thank the members of my doctoral committee—Gaetano Borriello, Anna Karlin, Abie Flaxman, Richard Anderson, Neal Lesh, and Ali Mokdad. A special thanks goes to Anna for six years of enthusiasm, intelligence, and patience—despite the winding and unpredictable path through graduate school that I have chosen; to Gaetano, for generously agreeing to provide me with such expert guidance through my final year; to Abie, for dispensing such brilliant technical advice; and to Neal, for inspiring my transition back to graduate school, suggesting this problem, and helping me stay grounded during difficult times.

I thank Gilbert Agaba, Nick Amland, Ken Bayona, Amelia Sagoff, and Paul Ssengooba for their help in obtaining the data sets in Chapter 4. It has been a pleasure to work with your organizations: the Grameen Foundation's AppLab, Dimagi, and ITIDO. I thank Crystal Eney and the other CSE advisors for their help in designing the survey in Chapter 5. I thank Carl Hartung for giving me a model on which to base my user-trace logging code.

I thank Lindsay Michimoto, Julie Svendsen, and Alicen Smith for their help throughout graduate school. I thank Dan Weld and Luke Zettlemoyer for sharing their expertise in machine learning.

I thank Adrienne Andrew, Michael Buettner, Kayur Patel, and the Change group—Yaw Anokwa, Nathan Breit, Waylon Brunette, Rohit Chaudhri, Nicki Dell, Brian DeRenzi, Mayank Goel, Carl Hartung, amd Mitch Sundt—for their valuable advice and support. I thank Kendall Turner for her unexpected but momentous suggestion to finish my degree.

My dissertation work has been supported by a National Science Foundation Graduate Research Fellowship under Grant No. (DGE-0718124), and grants from Google and Yahoo!

Finally, I thank Jenny Klein and my family—Janet, Stan, Sharon, Sarah, Marc, and Richa—for their immense support over the years. Words cannot express how grateful I am.

Bibliography

[1] Abdul Latif Jameel Poverty Action Lab, http://www.povertyactionlab.org/.

[2] A. Agresti. *Categorical Data Analysis*. Wiley-Interscience, 1990.

[3] American Association for Public Opinion Research. Interviewer falsification in survey research: Current best methods for prevention, detection and repair of its effects, 2003. http://www.amstat.org/sections/srms/falsification.pdf.

[4] N. Amland. Active data management and effective performance monitoring, comments, 2011. http://groups.google.com/group/ict4chw/browse_thread/thread/a59d01d6b64fe9af/0e0028d07c8f4101?lnk=gst&q=alvin#0e0028d07c8f4101.

[5] Y. Anokwa. *Improving Clinical Decision Support in Low-Income Regions*. PhD thesis, University of Washington, 2012.

[6] Y. Anokwa, C. Hartung, W. Brunette, G. Borriello, and A. Lerer. Open source data collection in the developing world. *Computer*, 42:97–99, 2009.

[7] M. Banko and E. Brill. Scaling to very very large corpora for natural language disambiguation. In *ACL '01*, pages 26–33, 2001.

[8] K. Banks and E. Hersman. FrontlineSMS and Ushahidi—a demo. In *ICTD '09*, pages 484–484, 2009.

[9] F. Benford. The law of anomalous numbers. *Proceedings of the American Philosophical Society*, 78(4):551–572, 1938.

[10] A. Bennett. Survey on problems of interviewer cheating: Observations on the so-called cheater problem among field interviewers. *International Journal of Opinion and Attitude Research*, 2:89–96, 1948.

[11] A. Bennett. Toward a solution of the "cheater problem" among part-time research investigators. *Journal of Marketing*, 2:470–474, 1948.

[12] M. Berg, J. Wariero, and V. Modi. Every child counts—the use of SMS in Kenya to support the community based management of acute malnutrition and malaria in children under five, 2009. http://www.mobileactive.org/files/file_uploads/ChildCount_Kenya_SMS.pdf.

[13] P. P. Biemer and L. E. Lyberg. *Introduction to Survey Quality*. Wiley-Interscience, 2003.

[14] B. Birnbaum, B. DeRenzi, A. D. Flaxman, and N. Lesh. Automated quality control for mobile data collection. In *DEV '12*, pages 1:1–1:10, 2012.

[15] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[16] J. Blaya and H. Fraser. Development, implementation and preliminary study of a PDA-based bacteriology collection system. In *American Medical Informatics Association Annual Symposium*, 2006.

[17] J. A. Blaya, H. Fraser, and B. Holt. E-health technologies show promise in developing countries. *Health Aff. (Millwood)*, 29(2):244–51, 2010.

[18] S. Boriah, V. Chandola, and V. Kumar. Similarity measures for categorical data: A comparative evaluation. In *ICDM '08*, pages 243–254, 2008.

[19] L. Bottou. Online algorithms and stochastic approximations. In D. Saad, editor, *Online Learning and Neural Networks*. Cambridge University Press, Cambridge, UK, 1998.

[20] H. Boyd and R. Westfall. Interviewers as a source of error in surveys. *Journal of Marketing*, 19:311–324, 1955.

[21] S. Bredl, P. Winker, and K. Kötschau. A statistical approach to detect cheating interviewers. Technical Report 39, University Giessen, Center for International Development and Environmental Research (ZEU), 2008.

[22] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[23] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[24] J. M. Bushery, J. W. Reichert, K. A. Albright, and J. C. Rossiter. Using date and time stamps to detect interviewer falsification. *Proceedings of the American Statistical Association (Survey Research Methods Section)*, pages 316–320, 1999.

[25] S. Byers and A. E. Raftery. Nearest-neighbor clutter removal for estimating features in spatial point processes. *Journal of the American Statistical Association*, 93(442):577–584, 1998.

[26] R. Caruana, N. Karampatziakis, and A. Yessenalina. An empirical evaluation of supervised learning in high dimensions. In *ICML '08*, 2008.

[27] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):15:1–15:58, 2009.

[28] K. Chen, H. Chen, N. Conway, J. M. Hellerstein, and T. S. Parikh. USHER: Improving data quality with dynamic forms. *IEEE Transactions on Knowledge and Data Engineering*, 23(8):1138–1153, 2010.

[29] K. Chen, J. M. Hellerstein, and T. S. Parikh. Designing adaptive feedback for improving data entry accuracy. In *UIST '10*, pages 239–248, 2010.

[30] K. Chen, J. M. Hellerstein, and T. S. Parikh. Data in the first mile. In *CIDR '11*, 2011.

[31] K. Chen, A. Kannan, Y. Yano, J. M. Hellerstein, and T. S. Parikh. Shreddr: pipelined paper digitization for low-resource organizations. In *DEV '12*, 2012.

[32] E. K. Chiteri, F. Odhiambo, G. Orwa, and K. Laserson. Quality control on HDSS automated application systems. `http://groups.google.com/group/ict4chw/browse_thread/thread/688fe36f845d56a5/b940831e85722424?lnk=gst&q=kenya#b940831e85722424`.

[33] M. J. Cho, J. L. Eltinge, and D. Swanson. Inferential methods to identify possible interviewer fraud using leading digit preference patterns and design effect matrices. *Proceedings of the American Statistical Association (Survey Research Methods Section)*, pages 936–941, 2003.

[34] CommCareHQ, `http://www.commcarehq.org/`.

[35] G. V. Cormack. Email spam filtering: A systematic review. *Found. Trends Inf. Retr.*, 1(4):335–455, 2008.

[36] L. P. Crespi. The cheater problem in polling. *Public Opinion Quarterly*, 9(4):431–445, 1945.

[37] *CyberTracker*. `http://cybertracker.co.za/`.

[38] M. H. DeGroot and M. J. Schervish. *Probability and Statistics*. Addison Wesley, 4th edition, 2011.

[39] N. Dell, N. Breit, T. Chaluco, J. Crawford, and G. Borriello. Digitizing paper forms with mobile imaging technologies. In *DEV '12*, pages 2:1–2:10, 2012.

[40] B. DeRenzi. *Technology for Workforce Performance Improvement of Community Health Programs*. PhD thesis, University of Washington, 2011.

[41] B. DeRenzi, G. Borriello, J. Jackson, V. S. Kumar, T. S. Parikh, P. Virk, and N. Lesh. Mobile phone tools for field-based health care workers in low-income countries. *The Mount Sinai Journal of Medicine, New York*, 78(3):406–418, 2011.

[42] B. DeRenzi, L. Findlater, J. Payne, B. Birnbaum, J. Mangilima, T. Parikh, G. Borriello, and N. Lesh. Improving community health worker performance through automated SMS. In *ICTD '12*, pages 25–34, 2012.

[43] L. Diero, J. K. Rotich, J. Bii, B. W. Mamlin, R. M. Einterz, I. Z. Kalamai, and W. M. Tierney. A computer-based medical record system and personal digital assistants to assess and follow patients with respiratory tract infections visiting a rural Kenyan health centre. *BMC Med. Inform. Decis. Mak.*, 6(21), 2006.

[44] eMocha, `http://emocha.org/`.

[45] EpiSurveyor, `http://www.episurveyor.org/`.

[46] F. B. Evans. On interviewer cheating. *Public Opinion Quarterly*, 25:126–127, 1961.

[47] T. Fawcett. An introduction to ROC analysis. *Pattern Recogn. Lett.*, 27:861–874, June 2006.

[48] FrontlineSMS, `http://www.frontlinesms.com/`.

[49] Global Pulse. Twitter and perceptions of crisis related stress, 2011. `http://www.unglobalpulse.org/projects/twitter-and-perceptions-crisis-related-stress`.

[50] R. M. Groves. *Survey Errors and Survey Costs*. Wiley, 1989.

[51] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: An update. *SIGKDD Explorations*, 11(1), 2009.

[52] C. Hartung. *Open Data Kit: Technologies for Mobile Data Collection and Deployment Experiences in Developing Regions*. PhD thesis, University of Washington, 2012.

[53] C. Hartung, Y. Anokwa, W. Brunette, A. Lerer, C. Tseng, and G. Borriello. Open Data Kit: Tools to build information services for developing regions. In *ICTD '10*, 2010.

[54] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001.

[55] J. M. Hellerstein. Quantitative data cleaning for large databases, 2008.

[56] T. P. Hill. The difficulty of faking data. *Chance Magazine*, 12(3):31–37, 1999.

[57] V. J. Hodge and J. Austin. A survey of outlier detection methodoligies. *Artificial Intelligence Review*, 22(2):85–126, 2004.

[58] H. S. Hong, I. K. Kim, S. H. Lee, and H. S. Kim. Adoption of a PDA-based home hospice care system for cancer patients. *Comput. Inform. Nurs.*, 27(6):365–71, 2009.

[59] C. C. Hood and J. M. Bushery. Getting more bang from the reinterview buck: Identifying "at risk" interviewers. *Proceedings of the American Statistical Association (Section on Survey Research Methods)*, pages 820–824, 1997.

[60] P. J. Huber. *Robust Statistics*. Wiley, 1981.

[61] J. A. Inciardi. Fictitious data in drug abuse research. *The International Journal of Addictions*, 16:377–380, 1981.

[62] Institute for Health Metrics and Evaluation, `http://www.healthmetricsandevaluation.org/`.

[63] S. L. James, A. D. Flaxman, and C. J. Murray. Performance of the Tariff Method: validation of a simple additive algorithm for analysis of verbal autopsies. *Population Health Metrics*, 9(1):31, 2011.

[64] I. T. Jolliffe. *Principal Component Analysis*. Springer, 2nd edition, 2002.

[65] G. Judge and L. Schechter. Detecting problems in survey data using Benford's Law. *Journal of Human Resources*, 44(1):1–24, 2009.

[66] A. Kapoor, N. Eagle, and E. Horvitz. People, quakes, and communications: Inferences from call dynamics about a seismic event and its influences on a population. In *AAAI Symposium on Artificial Intelligence for Development*, 2010.

[67] R. Khan. ClickDiagnostics: Experiences from Bangladesh, 2010. `http://groups.google.com/group/ict4chw/browse_thread/thread/96dd344a921f124e`.

[68] P. Kiecker and J. E. Nelson. Do interviewers follow telephone survey instructions? *Journal of the Market Research Society*, 38:161–176, 1996.

[69] E. A. Krejsa, M. C. Davis, and J. M. Hill. Evaluation of the quality assurance falsification interview used in the Census 2000 dress rehearsal. *Proceedings of the American Statistical Association (Section on Survey Research Methods)*, pages 635–640, 1999.

[70] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek. LoOP : Local outlier probabilities. In *CIKM '09*, pages 1649–1652, 2009.

[71] S. O. Lal, F. W. Smith, J. P. Davis, H. Y. Castro, D. W. Smith, D. L. Chinkes, and R. E. Barrow. Palm computer demonstrates a fast and accurate means of burn data collection. *J. Burn Care Rehabil.*, 21(6):559–61, 2000.

[72] N. Landwehr, M. Hall, and E. Frank. Logistic model trees. *Machine Learning*, 59(1-2):161–205, 2005.

[73] S. J. Lane, N. M. Heddle, E. Arnold, and I. Walker. A review of randomized controlled trials comparing the effectiveness of hand held computers with paper methods for data collection. *BMC Medical Informatics and Decision Making*, 6:23, 2006.

[74] S. le Cessie and J. C. van Houwelingen. Ridge estimators in logistic regression. *Applied Statistics*, 41(1):191–201, 1997.

[75] U. Lehmann and D. Sanders. Community health workers: What do we know about them? Technical report, World Health Organization, 2007.

[76] J. Li, J. M. Brick, B. Tran, and P. Singer. Using statistical models for sample design of a reinterview program. *Proceedings of the American Statistical Association (Survey Research Methods Section)*, pages 4681–4695, 2009.

[77] R. Likert. A technique for the measurement of attitudes. *Archives of Psychology*, 140:1–55, 1932.

[78] C. D. Mathers, D. M. Fat, M. Inoue, C. Rao, and A. D. Lopez. Counting the dead and what they died from: an assessment of the global status of cause of death data. *Bulletin of the World Health Organization*, 83(3):171–177, 2005.

[79] G. Mhila, C. Mushi, M. Steele, D. Roos, J. Jackson, B. DeRenzi, T. Wakabi, P. Dhadialla, C. Sims, and N. Lesh. Using mobile applications for community-based social support for chronic patients. In *HELINA '09*, 2009.

[80] T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[81] T. M. Mitchell. The discipline of machine learning. Technical report, School of Computer Science, Carnegie Mellon University, 2006.

[82] J. Murphy, R. Baxter, J. Eyerman, D. Cunningham, and J. Kennet. A system for detecting interview falsification. In *American Association for Public Opinion Research 59th Annual Conference*, 2004.

[83] C. J. Murray et al. Population health metrics research consortium gold standard verbal autopsy validation study: design, implementation, and development of analysis datasets. *Population Health Metrics*, 9(1):27, 2011.

[84] C. J. Murray and J. Frenk. Health metrics and evaluation: strengthening the science. *Lancet*, 371(9619):1191–9, 2008.

[85] M. Nigrini. A taxpayer compliance application of Benford's Law. *Journal of the American Taxation Association*, 18:72–91, 1996.

[86] Open Data Kit, `http://opendatakit.org/`.

[87] S. Papadimitriou, H. Kitagawa, P. Gibbons, and C. Faloutsos. LOCI: fast outlier detection using the local correlation integral. In *ICDE '03*, pages 315–326, 2003.

[88] T. Parikh, K. Ghosh, and A. Chavan. Design studies for a financial management system for micro-credit groups in rural india. In *CUU '03*, pages 15–22, 2003.

[89] T. S. Parikh. Engineering rural development. *Communications of the ACM*, 52(1):54–63, 2009.

[90] T. S. Parikh, P. Javid, S. K., K. Ghosh, and K. Toyama. Mobile phones and paper documents: Evaluating a new approach for capturing microfinance data in rural india. In *CHI '06*, pages 551–560, 2006.

[91] A. Patcha and J.-M. Park. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, 51(12):3448–3470, 2007.

[92] S. Patnaik, E. Brunskill, and W. Thies. Evaluating the accuracy of data collection on mobile phones: A study of forms, SMS, and voice. In *ICTD '09*, pages 74–84, 2009.

[93] Pendragon Forms, `http://pendragonsoftware.com/`.

[94] M. I. Petrovskiy. Outlier detection algorithms in data mining. *Programming and Computer Software*, 29(4):228–237, 2003.

[95] J. Porras and N. English. Data-driven approaches to identifying interviewer data falsification: The case of health surveys. *Proceedings of the American Statistical Association (Section on Survey Research Methods)*, pages 4223–4228, 2004.

[96] RapidSMS, `http://www.rapidsms.org/`.

[97] A. L. Ratan, S. Chakraborty, P. V. Chitnis, K. Toyama, K. S. Ooi, M. Phiong, and M. Koenig. Managing microfinance with paper, pen and digital slate. In *ICTD '10*, 2010.

[98] C. J. Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, 1979.

[99] I. Schreiner, K. Pennie, and J. Newbrough. Interviewer falsification in census bureau surveys. *Proceedings of the American Statistical Association (Survey Research Methods Section)*, pages 491–496, 1988.

[100] B. Settles. Active learning literature survey. Comp. Sci. Tech. Report 1648, University of Wisconsin–Madison, 2009.

[101] C. Shäfer, J.-P. Schräpler, K.-R. Müller, and G. G. Wagner. Automatic identification of faked and fraudulent interviews in surveys by two different methods. *Proceedings of the American Statistical Association (Survey Research Methods Section)*, pages 4318–4325, 2004.

[102] P. B. Sheatsley. An analysis of interviewer characteristics and their relationship to performance. *International Journal of Opinion and Attitude Research*, 5:79–94, 1951.

[103] J. Sherwani, S. Palijo, S. Mirza, T. Ahmed, N. Ali, and R. Rosenfeld. Speech vs. touch-tone: telephony interfaces for information access by low literate users. In *ICTD '09*, 2009.

[104] K. Shirima, O. Mukasa, J. A. Schellenberg, F. Manzi, D. John, A. Mushi, M. Mrisho, M. Tanner, H. Mshinda, and D. Schellenberg. The use of personal digital assistants for data entry at the point of collection in a large household survey in southern tanzania. *Emerging Themes in Epidemiology*, 4(2), 2007.

[105] N. Soleman, D. Chandramohan, and K. Shibuya. Verbal autopsy: current practices and challenges. *Bulletin of the World Health Organization*, 84(3):239–245, 2006.

[106] C. Stanfill and D. Waltz. Toward memory-based reasoning. *Communications of the ACM*, 29(12):1213–1228, 1986.

[107] M. Sumner, F. Eibe, and M. Hall. Speeding up logistic model tree induction. In *9th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 675–683, 2005.

[108] D. Swanson, M. J. Cho, and J. Eltinge. Detecting possibly fraudulent or error-prone survey data using Benford's Law. *Proceedings of the American Statistical Association (Survey Research Methods Section)*, pages 4172–4177, 2003.

[109] C. F. Turner, J. N. Gribble, A. A. Al-tayyib, and J. R. Chromy. Falsification in epidemiologic surveys: Detection and remediation. Technical Report 53, Research Triangle Institute, 2002.

[110] United Nations Department of Economic and Social Affairs, Population Division. *World Urbanization Prospects*, 2011.

[111] United Nations Department of Economic and Social Affairs, Statistics Division. *Household Sample Surveys in Developing and Transition Countries*, 2005.

[112] United Nations Development Programme. *The Human Development Report*, 2011.

[113] Ushahidi, `http://ushahidi.com/`.

[114] J. Van den Broeck, S. A. Cunningham, R. Eeckels, and K. Herbst. Data cleaning: detecting, diagnosing, and editing data abnormalities. *PLoS medicine*, 2(10):e267, 2005.

[115] D. R. Wilson and T. R. Martinez. Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research*, 6, 1997.

[116] I. H. Witten, E. Frank, and M. A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 3rd edition, 2011.

[117] World Bank. *Monitoring and Evaluation: Some Tools, Methods and Approaches*, 2004.

[118] World Economic Forum. *Big Data, Big Impact: New Possibilities for International Development*, 2012.

[119] P. Yu, M. de Courten, E. Pan, G. Galea, and J. Pryor. The development and evaluation of a PDA-based method for public health surveillance data collection in developing countries. *International Journal of Medical Informatics*, 78(8):532–42, 2009.

[120] K. Zhang, M. Hutter, and H. Jin. A new local distance-based outlier detection approach for scattered real-world data. In *PAKDD '09*, 2009.

[121] M. Zimic, J. Coronel, R. H. Gilman, C. G. Luna, W. H. Curioso, and D. a. J. Moore. Can the power of mobile phones be used to improve tuberculosis diagnosis in developing countries? *Transactions of the Royal Society of Tropical Medicine and Hygiene*, 103(6):638–40, 2009.

Appendix A
THE STUDY HABITS SURVEY

This appendix contains the entire contents of the Study Habits survey described in Chapter 5.

Table A.1: The Study Habits Survey.

Question	Response	Skip Logic
1. `interviewer-id` To the interviewer: please enter your interviewer ID.	*numeric*	
2. `consent-note` Read the following to the respondent: "As part of a research project from the Computer Science Department at the University of Washington, I am interviewing university students about study habits and academic attitudes. This survey should take between five and ten minutes. Participation is purely voluntary. You may decide to stop at any point without penalty or prejudice to you or the interviewer on the part of the research team. The interviewer will have a consent form that explains this study and lists contact information for the members of the research team. If you decide to participate in the survey, please read and sign the provided consent form."		
3. `consent` To the interviewer: did the participant sign the consent form?	{Yes, No}	If no, go to `no-consent`. Else, go to `taken-before`.
4. `no-consent` You must sign the consent form to take the survey.		Go to `thanks`.
5. `taken-before`		

Continued on next page

Table A.1 – *Continued from previous page*

Question	Response	Skip Logic
Have you taken this survey before?	{Yes, No}	If yes, go to `cant-take-before`. Else, go to `sex`.
6. `cant-take-before` You cannot take this survey more than once.		Go to `thanks`.
7. `sex` What is your sex?	{Male, Female}	
8. `age-years` What is your age in years?	*numeric*	If between 18 and 25 (inclusive), go to `university-student`. Else, go to `invalid-age`.
9. `invalid-age` You must be between the ages of 18 and 25 to take this survey.		Go to `thanks`.
10. `university-student` Are you a university student?	{Yes, No}	If yes, go to `uw-student`. Else, go to `not-university-student`.
11. `not-university-student` You must be a university student to take this survey.		Go to `thanks`.
12. `uw-student` Are you a student at the University of Washington?	{Yes, No}	If yes, go to `uw-college`. Else, go to `decided-major`.
13. `uw-college` In which college at the University of Washington are you enrolled?	{Arts and Sciences, Business, Education, Engineering, Law, Nursing, Public Health, Social Work, I don't know, Other}	If other, go to `other-college`. Else, go to `decided-major`.
14. `other-college` Enter the name of the college in which you are enrolled.	*free text*	
15. `decided-major` Have you decided on a major yet?	{Yes, No}	If yes, go to `cse-major`. Else, go to `hardest-course`.
16. `cse-major` Are you a computer science and engineering major?	{Yes, No}	If yes, go to `intended-major`. Else, go to `major`.
17. `major` What is your major?	*free text*	

Continued on next page

Table A.1 – *Continued from previous page*

Question	Response	Skip Logic
18. `intended-major` Is this your declared major, intended major, or both?	{Intended, Declared, Both}	
19. `hardest-course` Which course did you spend the most time studying for in the last year?	*free text*	
20. `when-hardest-course` When did you take it?	{Winter 2012, Fall 2011, Summer 2011, Spring 2011, Winter 2011}	
21. `enjoy-hardest-course` Did you enjoy it?	{Yes, No}	
22. `average-work` On average, how many hours do you spend studying?	{One hour for every hour spent in class, Two hours for every hour spent in class, Three hours for every hour spent in class, Four or more hours for every hour spent in class}	
23. `list-obligations` Read the following to the respondent: "I'm going to list some obligations that could take up time outside of studying. For each, I'll ask if it takes up more than one hour per week of your time. If it does, I'll ask you how many hours per week you spend on it."		
24. `paid-work` Do you spend more than an hour per week doing paid work?	{Yes, No}	If yes, go to `paid-work-hours`. Else, go to `volunteer-work`.
25. `paid-work-hours` How many hours per week do you spend on paid work?	*numeric*	
26. `volunteer-work` Do you spend more than an hour per week doing volunteer work?	{Yes, No}	If yes, go to `volunteer-work-hours`. Else, go to `research`.

Continued on next page

Table A.1 – *Continued from previous page*

Question	Response	Skip Logic
27. `volunteer-work-hours` How many hours per week do you spend on volunteer work?	*numeric*	
28. `research` Do you spend more than an hour per week doing research?	{Yes, No}	If yes, go to `research-hours`. Else, go to `family`.
29. `research-hours` How many hours per week do you spend on research?	*numeric*	
30. `family` Do you spend more than an hour per week on family obligations?	{Yes, No}	If yes, go to `family-hours`. Else, go to `other-obligations`.
31. `family-hours` How many hours per week do you spend on family obligations?	*numeric*	
32. `other-obligation` Apart from the obligations already asked about, is there another obligation that takes up more than one hour per week of your time?	{Yes, No}	If yes, go to `other-obligation-name`. Else, go to `more-costly`.
33. `other-obligation-name` What is that obligation?	*free text*	
34. `other-obligation-hours` How many hours per week do you spend on it?	*numeric*	
35. `more-costly` If tuition was more expensive for certain majors, would that stop you from pursuing a more costly major?	{Yes, definitely. No, additional expenses would not make a difference in my choice of major. Maybe, depending on how expensive it was.}	If maybe, go to `more-costly-how-much`. Else, go to `faculty`
36. `more-costly-how-much`		

Continued on next page

Table A.1 – *Continued from previous page*

Question	Response	Skip Logic
What is the lowest additional expense that would stop you from pursuing a more costly major? Please think about this question carefully before responding.	{One thousand dollars per year, Two thousand dollars per year, Three thousand dollars per year, It would have to more than three thousand dollars per year in additional expenses to deter me from pursuing a more costly major, I'm not sure how much money it would have to be}	
37. `faculty` How often in a quarter do you typically seek help from faculty for your course work?	{Zero, One - two, Three - six, Seven or more}	If zero, go to `facult-why-not`. Else, go to `tas`
38. `facult-why-not` Why do you not typically seek help from faculty?	*free text*	
39. `tas` How often in a quarter do you typically seek help from a TA for your course work?	{Zero, One - two, Three - six, Seven or more}	If zero, go to `tas-why-not`. Else, go to `tutors`
40. `tas-why-not` Why do you not typically seek help from TAs?	*free text*	
41. `tutors` How often in a quarter do you typically seek help from a tutor for your course work?	{Zero, One - two, Three - six, Seven or more}	If zero, go to `tutors-why-not`. Else, go to `advisors`
42. `tutors-why-not` Why do you not typically seek help from tutors?	*free text*	
43. `advisors` How often in a quarter do you typically seek help from an academic advisor for your course work?	{Zero, One - two, Three - six, Seven or more}	If zero, go to `advisors-why-not`. Else, go to `study-more`
44. `advisors-why-not` Why do you not typically seek help from advisors?	*free text*	
45. `study-more`		

Continued on next page

Table A.1 – *Continued from previous page*

Question	Response	Skip Logic
Do you think that you study more than the average student?	{Yes, No}	
46. `procrastinate` Do you think that you procrastinate more than the average student?	{Yes, No}	
47. `more-obligations` Do you think that you have more obligations outside of school than the average student?	{Yes, No}	
48. `name-phone-number` Read the following to the respondent: "For the purpose of verifying the accuracy of the data in this survey, the researchers conducting the study may perform checks to ensure certain interviews actually happened. If you consent to leave your phone number, one of the members of the research team may give you a call during the next two weeks to ensure that this interview took place. If you consent, your phone number will not be used for any other purpose besides this verification, and it will be discarded no later than three weeks after we receive this data. There are no consequences to you or the interviewer if you opt not to provide this information."		
49. `name-phone-consent` Do you consent to have the interviewer record your phone number?	{Yes, No}	If yes, go to `phone`. Else, go to `thanks`
50. `phone` What is your phone number?	*free text*	
51. `name` Whom should we ask for if we call to verify the interview took place?	*free text*	
52. `thanks` To the interviewer: please thank the respondent for his or her time.		

Appendix B
FOLLOW UP SESSION DETAILS

In this appendix, I provide additional technical details on the follow up sessions, including the algorithm used to perform the classification, how the fabrication scores were calculated, and the scripts that were read to participants. One complication is that these protocols changed slightly during an initial pilot period consisting of two follow up sessions in which six interviewers participated. Ultimately, the changes were minor enough that I decided to mix the data from the pilot period with the data from the main study period. For completeness, I describe here exactly how these protocols changed during this period.

First, in Section B.1, I describe the algorithm that was used for classification and how the output of that algorithm was used to generate the fabrication scores reported to participants. Second, in Section B.2, I list the various scripts that were used during the pilot and normal study period. A summary of how the protocols changed throughout the pilot is shown in Table B.1.

B.1 Algorithm and fabrication score

As stated in Section 5.3.3, for both the first and second rounds of each follow up session, a supervised classification algorithms was run on the real and fabricated data from the participants in the session. The features used for training and prediction were the feedback-eligible features listed in Table 5.2. Based on the output of 10-fold cross-validation, a nu-

Session	# Int.	R1 Script	R2 Script	Score
Pilot Session 1	4	V1	V1	V1
Pilot Session 2	2	V2	V1	V1
Rest of Sessions	22	V3	V2	V2

Table B.1: Fabrication score generation method and scripts used in follow up sessions.

Here's some more information about the purpose of the study. We are designing algorithms to automatically detect fabricated survey data. To do so, we are gathering survey data, some of which we know to be real, and some of which we know to be fabricated. Then we are testing how well our algorithms predict which data is fabricated.

Last week in the lab, you fabricated some survey data, but I did not explain why you were doing that. In this follow up session—now that you have collected real data and know the purpose of the study—I will have you fabricate some more data.

For the next **40 minutes**, please fabricate between **4 and 10 more surveys**. To make things interesting, I will be giving you feedback at the end of the 40 minutes on how well you fooled the algorithm. Each submission that you give me will be given a score by the algorithm, where a high score indicates that it could not tell your data was fabricated—or, in other words, that you did a good job fabricating data.

Think of this as a friendly challenge to see who can do the best job fooling the algorithm. The person who gets the highest average score over all of his or her submissions will get an **extra $10 gift certificate**. So please try to fabricate the data in a realistic way. You may think back your data collection over the last week, but you must make up new data. When you have finished fabricating forms, please put your phone down and let me know.

Figure B.1: Follow up session Round 1 script, version 1.

meric score, called the *fabrication score*, was given to each participant that indicated the ability of the classifier to identify that participant's fabricated data.

The classification algorithm used was LogitBoost with simple regression functions as base learners as implemented by the Weka framework [72, 107]. I chose this algorithm because it was efficient and performed well in some initial unpublished investigations on the data from Chapter 4. Ultimately, I did not choose this algorithm as one of the supervised classification techniques that I investigated in detail in Chapters 4 and 6. However, its performance was similar to the algorithms that I did end up choosing. This level of performance was more than sufficient to generate reasonable fabrication scores during the follow up sessions.

To generate fabrication scores, a separate classifier was trained for each participant i in the follow up session. To do so, a subset $\mathcal{H}_i$ of the training data was constructed that consisted of the real data from *all* of the interviewers in the follow up session[1] and the fake data from

[1]The reason for including the real data from all of the interviewers was to increase the size of the data

Here's some more information about the purpose of the study. We are designing algorithms to automatically detect fabricated survey data. To do so, we are gathering survey data, some of which we know to be real, and some of which we know to be fabricated. Then we are testing how well our algorithms predict which data is fabricated.

Last week in the lab, you fabricated some survey data, but I did not explain why you were doing that. In this follow up session—now that you have collected real data and know the purpose of the study—I will have you fabricate some more data.

For the next **40 minutes**, please fabricate between **4 and 10 more surveys**. To make things interesting, I will be giving you feedback at the end of the 40 minutes on how well you fooled the algorithm. Each submission that you give me will be given a score by the algorithm, where a low score indicates that it could not tell your data was fabricated—or, in other words, that you did a good job fabricating data.

Think of this as a friendly challenge to see who can do the best job fooling the algorithm. The person who gets the **lowest average score** over all of his or her submissions will get an **extra $10 gift certificate**. Since we are looking at the average score, there is not necessarily an advantage to creating a lot of fake data. You may—but do not have to—use the entire 40 minutes.

Please try to fabricate the data in a realistic way. You may think back to your data collection over the last week, but you must make up new data. When you have finished fabricating forms, please put your phone down and let me know. If you are still working after 35 minutes, I will let you know and ask you to finish the survey that you are currently working on.

Figure B.2: Follow up session Round 1 script, version 3. The only difference between this and Version 2 is that the word "highest" was changed to "lowest."

only interviewer i. Let $\mathcal{F}_i \subseteq \mathcal{H}_i$ denote the fabricated forms from interviewer i. Once the data set $\mathcal{H}_i$ was constructed, 10-fold cross-validation was used to get a probability p_x from LogitBoost for each form x from interviewer i. This probability was the classifier's belief that form x was fabricated.

The set of probabilities p_x for interviewer i were aggregated to get the fabrication score for interviewer i. There were two versions of how this was done, which changed between the pilot period and the rest of the study:

- **Version 1**: The fabrication score for participant i was the mean of $(1 - p_x)$ over all

---

set and protect against the possibility that interviewer i did not collect any real data during the interview period.

Here's some more information about the purpose of the study. We are designing algorithms to automatically detect fabricated survey data. To do so, we are gathering survey data, some of which we know to be real, and some of which we know to be fabricated. Then we are testing how well our algorithms predict which data is fabricated.

Last week in the lab, you fabricated some survey data, but I did not explain why you were doing that. In this follow up session—now that you have collected real data and know the purpose of the study—I will have you fabricate some more data.

For the next **40 minutes**, please fabricate **between 4 and 10 more surveys**. To make things interesting, I will be giving you feedback at the end of the 40 minutes on how well you fooled the algorithm. Each submission that you give me will be given a score by the algorithm that is between 0 and 1, where a low score indicates that it could not tell your data was fabricated—or, in other words, that you did a good job fabricating data.

If you get an average score of less than 0.75 over all of your submissions, you will get an **extra $10 gift certificate**. (About half of the study participants so far have been able to do this well.) Since we are looking at the average score, there is not necessarily an advantage to creating a lot of fake data. You may—but do not have touse the entire 40 minutes.

Please try to fabricate the data in a realistic way. You may think back to your data collection over the last week, but you must make up new data. When you have finished fabricating forms, please put your phone down and let me know. If you are still working after 35 minutes, I will let you know and ask you to finish the survey that you are currently working on.

Figure B.3: Follow up session Round 1 script, alternate.

forms $x \in \mathcal{F}_i$. This is the mean probability of being real that the classifier assigned to i's fabricated forms, and hence the *higher* the score, the better interviewer i is at fabricating data.

- **Version 2**: The fabrication score for participant is the difference between two values: 1) the mean of p_x for all forms $x \in \mathcal{F}_i$ and 2) the mean of p_x for all forms $x \in \mathcal{H}_i \setminus \mathcal{F}_i$. Here, the *lower* the score, the better interviewer i is at fabricating data.

The reason I changed from Version 1 to Version 2 is that the scores generated with Version 1 were unstable to changes in the size of the data set.

Now I'm going to give each of you some personalized feedback on how you did. The algorithm looks at various measures of how you entered the data for each form. High or low values of these measures may indicate that your data is likely to be fabricated.

For each of you, I'm going to send you an email that gives the top three measures that were used to predict which of your forms were fabricated. You can think of these as the clues that the algorithm used to figure out which forms you were fabricating.

The specific measures used may vary from person to person. Please do not share them with each other.

After I send you this report, I will ask you to spend another **40 minutes** fabricating another **4-10 surveys**. When you fabricate these surveys, keep in mind the report you get. This may help you to better fool the algorithm. As before, the person who gets the lowest average score will get another **$10 gift certificate**.

Figure B.4: Follow up session Round 2 script, version 2. The only difference between this and Version 1 is the word "highest" was changed to "lowest."

Now I'm going to give each of you some personalized feedback on how you did. The algorithm looks at various measures of how you entered the data for each form. High or low values of these measures may indicate that your data is likely to be fabricated.

I'm going to send you an email that gives the top three measures that were used to predict which of your forms were fabricated. You can think of these as the clues that the algorithm used to figure out which forms you were fabricating.

After I send you this report, I will ask you to spend another **40 minutes** fabricating another **4-10 surveys**. When you fabricate these surveys, keep in mind the report you get. This may help you to better fool the algorithm. Again, if you do well, you will get another **$10 gift certificate**. Specifically you must get an average score that is **no more than half** of the score that you got in the first round to get the $10.

Figure B.5: Follow up session Round 2 script, alternate.

B.2 Scripts

There were two scripts that I read to the participants in the follow up session, one before Round 1 and one before Round 2. For both of these scripts, there was an *alternate* version

that I read if a follow up session consisted of only one participant.

There were three versions of the Round 1 (non-alternate) script. The first version is shown in Figure B.1. I used Version 1 in the first follow up session, which had four participants. I changed this version after the first session to emphasize the fact that there is not necessarily an advantage to creating a lot of fabricated data and that the participants did not have to use the entire 40 minutes available to them. I used Version 2 in the second follow up session, which had two participants. Finally, to match the change from Version 1 to Version 2 of the fabrication score generation algorithm, I changed this version to indicate that a *lower* score was better, instead of a *higher* score. Version 3, which was the same as Version 2 except for this change, is shown in Figure B.2.

There were two versions of the Round 2 (non-alternate) script. The only change between the two versions was that the word "highest" was changed to "lowest" in the second version to match the change in the fabrication score generation algorithm. Version 2 is shown in Figure B.4.

Because I did not use the alternate scripts until after the pilot period, there is only one version of the Round 1 and Round 2 alternate scripts. The Round 1 script is shown in Figure B.3, and the Round 2 script is shown in Figure B.5.

Appendix C
FEATURES CHOSEN BY LOGISTIC REGRESSION

The tables in this appendix list all of the features chosen by the bottom-up greedy feature selection algorithm outlined Section 3.2.3 and used in Chapter 6. The features are listed in the order that they were chosen.

diff-total-answer-time
diff-average-work-ord
sqdiff-age-years-response
faculty-num-contiguous-edits
diff-faculty-time

Table C.1: Features chosen for logistic regression classifiers for $\mathcal{D}_0$.

diff-total-delay-to-first-edit
diff-average-work-time
total-time
other-college-num-contiguous-edits

Table C.2: Features chosen for logistic regression classifiers for $\mathcal{D}_1$.

diff-total-delay-to-first-edit
total-scrolled
diff-tutors-why-not-delay-to-first-edit
diff-research-time
sqdiff-num-swipes

Table C.3: Features chosen for logistic regression classifiers for $\mathcal{D}_2$.

uw-student-response
advisors-response
uw-college-response
age-years-response

Table C.4: Features chosen for logistic regression classifiers on response data for $\mathcal{D}_0$.

uw-college-response
decided-major-response
more-costly-response
sqdiff-hardest-course-hours-response

Table C.5: Features chosen for logistic regression classifiers on response data for $\mathcal{D}_1$.

uw-student-response
tas-response
tutors-response

Table C.6: Features chosen for logistic regression classifiers on response data for $\mathcal{D}_2$.

num-conditional
diff-total-scrolled
diff-mean-string-length
more-obligations-response
sqdiff-tutors-num-non-contiguous-edits
sqdiff-facult-why-not-num-non-contiguous-edits

Table C.7: Features chosen for logistic regression classifiers with no timing-based features for $\mathcal{D}_0$.

diff-mean-select-one-contiguous-edits
diff-intended-major-num-contiguous-edits
diff-taken-before-num-non-contiguous-edits
sqdiff-intended-major-num-non-contiguous-edits
diff-volunteer-work-hours-num-contiguous-edits
family-num-contiguous-edits
num-conditional
diff-study-more-num-contiguous-edits
diff-more-costly-num-non-contiguous-edits
sqdiff-total-hours-on-activities

Table C.8: Features chosen for logistic regression classifiers with no timing-based features for $\mathcal{D}_1$.

diff-cse-major-num-non-contiguous-edits
tas-why-not-num-non-contiguous-edits
intended-major-num-contiguous-edits
uw-college-num-non-contiguous-edits
sqdiff-advisors-ord
max-select-one-non-contiguous-edits
diff-average-work-num-contiguous-edits
diff-more-costly-how-much-num-non-contiguous-edits

Table C.9: Features chosen for logistic regression classifiers with no timing-based features for $\mathcal{D}_2$.

median-delay-to-first-edit
when-hardest-course-delay-to-first-edit
faculty-time

Table C.10: Features chosen for logistic regression classifiers with no interviewer-normalized features for $\mathcal{D}_0$.

more-costly-delay-to-first-edit
total-time
average-work-delay-to-first-edit
intended-major-num-contiguous-edits
tutors-delay-to-first-edit

Table C.11: Features chosen for logistic regression classifiers with no interviewer-normalized features for $\mathcal{D}_1$.

total-delay-to-first-edit
median-answer-time
more-costly-response
tas-response
decided-major-response

Table C.12: Features chosen for logistic regression classifiers with no interviewer-normalized features for $\mathcal{D}_2$.