Modern Survey Estimation with Social Media and Auxiliary Data

by

Robyn A. Ferg

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Statistics)
in The University of Michigan
2020

Doctoral Committee:

Assistant Professor Johann A Gagnon-Bartsch, Chair
Professor Frederick Conrad
Professor Kerby Shedden
Associate Professor Gongjun Xu

Robyn A. Ferg

fergr@umich.edu

ORCID iD: 0000-0001-6498-7551

Table of Contents

List of Figures

List of Tables

List of Appendices

Abstract

Traditional survey methods have been successful for nearly a century, but recently response rates have been declining and costs have been increasing, making the future of survey science uncertain. At the same time, new media sources are generating new forms of data, population data is increasingly readily available, and sophisticated machine learning algorithms are being created. This dissertation uses modern data sources and tools to improve survey estimates and advance the field of survey science.

We begin by exploring the challenges of using data from new media, demonstrating how relationships between social media data and survey responses can appear deceptively strong. We examine a previously observed relationship between sentiment of "jobs" tweets and consumer confidence, performing a sensitivity analysis on how sentiment of tweets is calculated and sorting "jobs" tweets into categories based on their content, concluding that the original observed relationship was merely a chance occurrence. Next we track the relationship between sentiment of "Trump" tweets and presidential approval. We develop a framework to interpret the strength of this observed relationship by implementing placebo analyses, in which we perform the same analysis but with tweets assumed to be unrelated to presidential approval, concluding that our observed relationship is not strong. Failing to find a meaningful signal, we next propose following a set of users over time. For a set of politically active users, we are able to find evidence of a political signal in terms of frequency and sentiment of their tweets around the 2016 presidential election.

In a given corpus of tweets, there are likely to be several topics present, which has the potential to introduce bias when using the corpus to track survey responses. To help discover and sort tweets into these topics, we create a clustering-based topic modeling algorithm. Using the entire corpus, we create distances between words based on how often they appear together in the same tweet, create distances between tweets based on the distance between words in the tweets, and perform clustering on the resulting distances. We show that this method is effective using a validation set of tweets and apply it to the corpus of tweets from politically active users and "jobs" tweets.

Finally, we use population auxiliary data and machine learning algorithms to improve survey estimates. We develop an imputation-based estimation method that produces an unbiased estimate of the mean response of a finite population from a simple random sample when population auxiliary data are available. Our method allows for any prediction function or machine learning algorithm to be used to predict the response for out-of-sample observations, and is therefore able to accommodate a high dimensional setting and all covariate types.

Exact unbiasedness is guaranteed by estimating the bias of the prediction function using subsamples of the original simple random sample. Importantly, the unbiasedness property does not depend on the accuracy of the imputation method. We apply this estimation method to simulated data, college tuition data, and the American Community Survey.

Chapter 1

Introduction

Surveys are crucial for social research and used in a wide variety of application, including official statistics, political polling, and market reserach. Results from these surveys are important for setting public policy and understanding the public's reaction to events. Traditional survey methodology has its roots in the early 1900s, when random sampling, as opposed to purposeful sampling, was thought to be an important component to survey sampling (Bowley 1906). It wasn't until Neyman (1934) that probability sampling became an essential component of survey sampling, applying similar methods to social surveys as Fisher had applied to agricultural experiments. Over the subsequent decades, classical theory on survey sampling had nearly been completed (Hansen and Hurwitz 1943; Horvitz and Thompson 1952).

While traditional survey sampling methods have been successfully implemented for many years, recently they are becoming increasingly difficult to perform, with people being more reluctant to respond and growing costs to implementation. Despite many methods being proposed to overcome this problem, such as methods for analyzing data from non-simple random samples, the future of survey science remains unclear (Keeter 2012; Massey and Tourangeau 2013).

In the modern era, there are many forms of data that were not available in the past. This is due to multiple reasons, including new forms of media being developed, data being generated at an unprecedented rate, and data being more widely available than ever before. This 'new data' ushers in a new era of survey science, with novel methods of utilizing this new data being developed and implemented in practice. At the same time, new predictive machine learning methods are being developed. These new algorithms can be applied to a wide variety of data types and capture subtle relationships between covariates and a given response. By taking advantage of data from new sources and developments in machine learning, both separately and together, precision and accuracy of estimates from surveys can be improved upon.

One proposal in public opinion research is using data extracted from social media to supplement or replace traditional surveys (Murphy et al. 2014). On its face this might seems like a very fruitful method of tracking public opinion, as social media has several advantages over traditional surveys: it is inexpensive

to acquire data (e.g. social media posts are sometimes free to obtain using an API), there is no burden on the respondent (i.e. user), a variety of topics are discussed over social media, and much of the general public engages with social media.

Twitter has perhaps been the most widely used social media platform for statistical analysis due its global popularity and public availablity of its data. There have been many studies establishing that signals of interest can be extracted from Twitter data. For example, tweets have been shows to predict the results of elections in Europe (Tumasjan et al. 2010; Ceron et al. 2014). This phenomenom is also seen in the US, with Wikipedia page views helping to predict elections (Smith and Gustafson 2017). Golder and Macy (2011) track the mood of Twitter users around the globe by their tweets, finding daily, weekly, and seasonal patterns. Antenucci et al. (2014) predict unemployment claims based on the number of tweets mentioning phrases related to 'laid off'.

Early results also suggest that public opinion can be captured using social media data. O'Connor et al. (2010) were one of the earliest and most influential analyses connecting social media data to public opinion polling. They found correlations between sentiment of tweets containing the word "jobs" and consumer confidence in 2008-2009 ($r = 0.731$ as measured by Gallup, and $r = 0.635$ as measured by the Index of Consumer Sentiment), between sentiment of "Obama" tweets and presidential approval in 2009 ($r = 0.725$), and between frequency of "Obama" and "McCain" tweets and presidential election polling in 2008 ($r = 0.79$ for "Obama" tweets, $r = 0.74$ for "McCain" tweets). Twitter was in its infancy during this time frame, but relationships nonetheless remained strong over time as Twitter gained popularity. Cody et al. (2016), for example, found similar correlations between sentiment of "Obama" tweets and presidential approval. This phenomenon was not only present in tweets from the United States, but with other social media platforms in other countries as well; Daas and Puts (2014) found strong relationships between sentiment of Dutch social media posts and consumer confidence in the Netherlands. These early results helped to spark optimism in the idea of replacing traditional public opinion surveys with social media data.

Despite the initial positive results, there are many criticisms and inconsistencies of social media analyses, some of which were not fully realized until years later. For example, O'Connor et al. (2010) failed to find a relationship between "job" (as opposed to "jobs") or "economy" tweets and consumer confidence, raising concerns about the robustness of the findings. Further confusing the issue, Cody et al. (2016) did find a relationship between "job" tweets and consumer confidence, resulting in a set of subtly contradictory findings. Daas and Puts (2014) found correlations between Dutch consumer sentiment and various subsets of Dutch social media messages (such as messages containing pronouns, messages containing the most frequent spoken and written words in Dutch, and messages containing the Dutch equivalents of "the" and "a/an") that were just as strong as messages containing words about the economy, raising red flags for whether the economic tweets were truly capturing consumer confidence. Furthermore, there are many ways in which social media data differs from traditional survey data, such as with population and topic coverage (Schober et al. 2016). New forms of bias and measurement error are introduced and constantly changing due to the nature of

social media use. Social media data is not the only form of data from new media that has "believers" and "skeptics" in terms of aiding survey estimation. Another form of new data that followed a similar story is Google Trends (i.e. Google searches over time and location) (Jun, Yoo, and Choi 2018), which has famously been used to track the number of flu cases each year (Yang, Santillana, and Kou 2015; Dugas et al. 2013). However, this model needs to updated each year and failed to catch certain spikes in flu cases (Lazer et al. 2014). As another example, Choi and Varian (2009) predict unemployment claims using categories of Google Trends, and go on to predict, among others, travel and automobile sales (Choi and Varian 2012).

We add to the field of analyzing Twitter data in a number of ways, the first being in developing methods to evaluate relationships observed between tweets containing some keyword and survey responses. The first relationship we consider is between sentiment of "jobs" tweets and consumer confidence, as was found in O'Connor et al. (2010). If there is truly an underlying relationship between sentiment of tweets and survey responses, the observed relationship should be robust to changes in sentiment calculation. We perform such a sensitivity analysis, finding that these seemingly small changes can drastically change the resulting relationship. By taking a closer look at the individual "jobs" tweets, we notice that while the intention was to purposefully select tweets related to the economy, that is not the case. We create an algorithm to sort the "jobs" tweets into five different categories. However, this did not restore the relationship. We conclude that the relationship between "jobs" tweets and consumer confidence was likely spurious. Much of this work can also be found in Conrad et al. (2019).

In our analysis of "jobs" tweets, we found it relatively easy to adjust analysis parameters such that a seemingly strong relationship is found. Sensitivity methods used in the "jobs" analysis can help to determine whether a relationship between tweets and survey responses is spurious, but given the optimizing over parameters that is usually performed in observing such a relationship, it does not answer the question of how strong and meaningful the relationship itself is. We develop a framework to do just that, and apply the framework in the context of presidential approval. Using methods similar to O'Connor et al. (2010) and Cody et al. (2016), we find the correlation between sentiment of "Trump" tweets and presidential approval from 2017 through mid-2019. In doing so, we optimize smoothing and lag parameters, and therefore traditional correlation significance tests fail. To determine how strong the observed relationship is, we use the idea of placebo analyses: we perform the exact same analysis, but with tweets assumed to be unrelated to presidential approval. Comparing our observed correlation with "Trump" tweets to the distribution of correlations from the placebo analysis, we conclude that our observed correlation is not strong.

Failing to find evidence that previously found relationships are meaningful, we propose a new method of collecting tweets: instead of following tweets containing a given word over time, we follow users over time. We create a set of politically active users and classify them as Democratic or Republican based on the accounts they follow. We find convincing evidence of a political signal in terms of frequency and sentiment of tweets from our set of Democrats and Republicans around the 2016 presidential election.

In all of the previously described analyses, the corpora of tweets were carefully chosen to capture a

chosen signal of interest. However, despite these intentions, the resulting corpora contained tweets relating to multiple topics. For example, as was evident from our analysis of "jobs" tweets, the corpus that was intended to be about jobs in an economic sense contained tweets about many topics (e.g. our corpus of "jobs" tweets contains tweets about job losses and Steve Jobs). Tweets from unwanted topics have the potential to do more than add pure noise; they could introduce bias. Thus, we want a method to filter and sort tweets to better understand what is being disucssed in a given corpus. Topic modeling is one method of doing this. Many topic modeling methods are based off of Latent Dirichlet Allocation (LDA) (Blei, Ng, and Jordan 2003), which assumes the topic of each word in a document is drawn from a multinomial distribution, and the word itself is drawn from another multinomial distribution according to the topic. While LDA works well with larger documents, it often fails with shorter documents such as tweets. Aggregation methods have been proposed, where tweets are aggregated together by some criteria and LDA is applied to the aggregated tweets (Hong and Davison 2010; Mehrotra et al. 2013; Alvarez-Melis and Saveski 2016; Quan et al. 2015). Given the nature of tweeting and character limit of tweets, a more realistic assumption may be to assign each tweet to a single topic. Zhao et al. (2011), for example, provide a variant of the LDA model where they make this assumption.

To better understand what is being discussed in a corpus of tweets and filter unwanted tweets, we introduce a clustering-based topic modeling algorithm. This algorithm creates a distance between each word in the corpus's vocabulary, and uses those distances to find the distance between each tweet in the corpus. A distance-based clustering algorithm is applied to the resulting distance matrix to estimate the latent topic for each tweet. We use this algorithm on a validation set of tweets from Twitter users who tweet about very different topics to confirm its effectiveness. We apply this algorithm to our set of politically active users and our corpus of "jobs" tweets.

Tweets often contain auxiliary data, such as the user information, that may be useful in analyses, especially when this information is known for the entire population of interest. While our topic modeling algorithm by design does not take advantage of those auxiliary variables, we demonstrate how they can be beneficial. Auxiliary population is not limited to just social media data, but is available in a wide variety of applications.

Another modern method of improving survey estimates is incorporating auxiliary population information and machine learning prediction models. Simple random samples resemble the population from which the sample was drawn in expectation, but deviations are present in individual samples. Population auxiliary information can be used to minimize the effect of these deviations and decrease the standard error of resulting estimates. While population data has been collected for centuries (Bethlehem 2009), it has not always been easily accessible. Population auxiliary data has long been used to improve estimates from samples; the use of ratio and regression estimators go back decades (Hartley and Ross 1954; Williams 1961), although these methods typically only assume knowledge of a population covariate mean, as estimates with more information would be computationally intensive before computers were widely available.

Many current methods for incorporating population auxiliary data with sample data rely on predictive modeling, where models trained on sample data are used to predict responses for out-of-sample observations. More sophisticated predictive models (e.g. Hastie, Tibshirani, and Friedman (2009)) are helpful in this regard: better predictive models lead to better estimated responses. However, with more sophisticated predictive modeling techniques it can be easier to overfit a model to a sample, especially for smaller sample sizes or when the feature dimension is high. Model assumptions also might not be exactly met by the given population. These can lead to overly optimistic estimates of standard error and biased estimates for response predictions, and ultimately biased estimates for functions of the population response.

We introduce a new unbiased estimation method for predicting a population mean response from a simple random sample given available population auxiliary information. For any chosen prediction function, we estimate the bias of that prediction function by using leave-one-out subsamples of the original sample. This method does not result in a large loss of precision over standard adjustment techniques using predictive modeling. Since our estimation method works with any arbitrary prediction function, we are able to add to the literature of incorporating modern machine learning models into survey estimation. Importantly, the unbiasedness property of our method does not depend on the accuracy of the imputation method used. We apply this method to simulated data, college tuition data, and the American Community Survey.

This dissertation is organized as follows. In chapter 2 we present two case studies in tracking public opinion survey responses with data extracted from Twitter. The first is a sensitivity analysis on the relationship between "jobs" tweets and consumer confidence. In the second we search for a political signal in Twitter data, developing a method to interpret the strength of the observed correlation between the sentiment of "Trump" tweets and presidential approval. We then search for a political signal when following politically active users over time. In chapter 3 we develop a clustering-based topic modeling algorithm for tweets. In chapter 4 we develop a new unbiased estimation method for a population mean response with presence of population auxiliary variables using leave-one-out predictive modeling. Chapter 5 concludes.

Chapter 2

Signals in Twitter Data: Case Studies in Consumer Confidence and Politics

Relationships found between public opinion polls and data extracted from social media have led to optimism about supplementing traditional surveys with these new sources of data. However, many initial findings have not been met with usual levels of scrutiny and skepticism. Our goal is to introduce a higher level of scrutiny to these types of analyses. In doing so, we demonstrate challenges of using social media data to track survey responses: seemingly small researcher decisions can have a large effect on observed relationships, relationship are inconsistent across time, and relationships are not much larger than we would observe by chance.

We first consider the relationship between sentiment of "jobs" tweets and consumer confidence, a relationship that has been observed to weaken over time. In hopes of restoring the relationship, we classify "jobs" tweets into categories based on their content and calculate sentiment of tweets using a variety of methods. None of these approaches improved the relationship in the original or more recent data. We find no evidence that the original relationship in these data was more than a chance occurrence.

We then focus on political signals in Twitter data, which we believe might be some of the strongest signals on social media, providing an illuminating test case. Our first contribution is to develop a framework to interpret the strength of relationships found between public opinion poll surveys and tweets containing a given keyword. Following methods that exist in the literature, we measure the association between survey based measures of presidential approval and tweets containing the word "Trump". We then implement placebo analyses, in which we perform the same analysis as with the "Trump" tweets but with tweets unrelated to presidential approval, concluding that the relationship between "Trump" tweets and survey responses is not strong. As our second contribution, we suggest following social media users longitudinally. For a set of politically active Twitter users, we classify users as a Democrat or Republican and find evidence of a political signal in terms of frequency and sentiment of their tweets around the 2016 presidential election. However, even in this best-case scenario of focusing exclusively on politics and following users who are politically

engaged, the signal found is relatively weak. For the goal of supplementing traditional surveys with data extracted from social media, these results are encouraging, but cautionary.

2.1 Introduction

Surveys are critical for understanding public opinion and setting public policy. While asking survey questions to samples designed to represent the entire population has been very successful for many years, generally producing quite accurate results, surveys are becoming increasingly costly to perform and people are increasingly reluctant to respond (De Heer and De Leeuw 2002). It is unclear whether the traditional method of gathering survey responses will remain viable in the future. One proposed alternative, as laid out by the AAPOR task force on big data (Murphy et al. 2014), is to use data gathered from social media to supplement or in some cases replace traditional surveys (Hsieh and Murphy 2017). While there are many open problems, early analyses have been promising, suggesting there may be an underlying relationship between some social media data and some public opinion surveys.

As one of the world's largest and most popular social media platforms, Twitter has been used for many studies in the social sciences (e.g. Golder and Macy 2011). Due to its popularity, a wide variety of topics are discussed, making Twitter data potentially applicable to many fields of study. Moreover, recent posts and user profile information are publicly available through Twitter's API, unlike many other social media platforms.

Early analyses were promising, finidng high correlations when tracking public opinion surveys with tweets containing a given keyword. For example, O'Connor et al. (2010) calculate correlations between sentiment of tweets containing a given word and consumer confidence, presidential approval, and election polling. They find a high positive correlation between sentiment of tweets from 2009 containing the word "Obama" and President Obama's 2009 presidential approval rating. They also find a high correlation between Obama's standing in 2008 presidential election polls and the frequency - but not sentiment - of "Obama" tweets. Surprisingly, they also find a positive correlation between the frequency of tweets that contain the word "McCain" (Obama's opponent in the 2008 presidential election) and Obama's standing in election polls. As demonstrated with these "Obama" and "McCain" tweets, sometimes a relationship is found with only sentiment of tweets, other times with only frequency of tweets, and not always in the direction one might expect.

Cody et al. (2016) find similar correlations with more recent tweets through 2015. Among others, correlations are found between Obama's quarterly presidential approval and average quarterly sentiment of tweets containing the word "Obama" from 2008 through 2015. They find a correlation of 0.56 between sentiment of "Obama" tweets and quarterly presidential approval with no lag, which increases to 0.76 when predicting presidential approval one quarter out. The lag is interpreted as Twitter data having the potential to predict future presidential approval.

Not only is this phenonomon found in tweets from the US, but from other countries and other social media platforms as well. Daas and Puts (2014) compare the sentiment of social media messages from multiple social media platforms in the Netherlands from 2009 through 2014 to consumer confidence in the Netherlands, finding very high correlations. All of these findings suggest there may be an underlying relationship between data extracted from social media and public opinion surveys.

However, inconsistencies in these initial analyses warrant skepticism in underlying relationships between social media data and survey responses. In O'Connor et al. (2010), sometimes a high correlation is observed between survey responses and sentiment of tweets, and other times between survey responses and frequency of tweets. O'Connor et al. (2010) did not find a relationship between "job" (as opposed to "jobs") or "economy" tweets and consumer confidence. This is contrast to Cody et al. (2016), who did find a relationship between "job" tweets and consumer confidence. Daas and Puts (2014) find correlations between consumer confidence and sentiment of twenty subsets of Dutch language social media messages. Some of these subsets contain words related to consumer confidence (e.g. economy, job, jobs, etc.), whereas other subsets of social media messages include messages containing pronouns, messages containing the most frequent spoken and written words in Dutch, and messages containing the words "the" and "a/an". Nearly all of the twenty subsets of social media messages were found to have very high correlations with consumer confidence. Surprisingly, the correlation between consumer confidence and sentiment of tweets containing a word related to the economy is no stronger than the correlation between consumer confidence and any other subset of tweets. This result is interpreted positively by Daas and Puts: changes in an underlying mood of the Dutch population affect both the population responding to consumer confidence surveys and the population posting social media messages. However, we interpret the result with more skepticism. We would expect tweets with keywords related to the economy to be more closely related to consumer confidence than all tweets or tweets subsetted by, say, personal pronouns. That this is not the case raises questions about how strong these relationships between Twitter and various public opinion polls really are.

There are two main contributions in this chapter. Our first contribution is methodological. If social media is to be reliably used to track public opinion, there needs to be a method of evaluating the strength of association between social media data and public opinion surveys. While inconsistencies cast doubt on the credibility of previously observed relationships between Twitter sentiment and public opinion surveys, there remains a need for a systematic framework to interpret the strength of such relationships. Taking the previously observed relationship between "jobs" tweet and consumer confidence, we perform a sensitivity analysis on how sentiment of tweets is calculated, showing that it can be relatively easy to observe spurious relationships that are deceptively strong. This demonstrates the need for a systematic way of interpreting the strength of such an observed relationship. To address this issue we propose the use of placebo analyses. The idea behind a placebo analysis is to replicate the primary analysis but using variables that are known to have no true relationship with the response. As an example of a placebo analysis, DiNardo and Pischke (1996) revisited a previous study that claimed wage differentials were due to computer use in the workplace.

When replacing the variable for computer use with pen/pencil use, the estimated effect of pencil use on wage differentials was similar to the estimated effect of computer use. This casts doubt on the original claim that computers in the workplace were causing the wage differential since the true effect for the placebo variable (pencil use) should be zero. The implication of an estimated non-zero effect is that the original analysis was not credible, see Athey and Imbens (2017) for further details. We develop a framework to evaluate and interpret the strength of observed correlations between social media sentiment and public opinion surveys by essentially performing multiple placebo tests. In the context of presidential approval, we first calculate the correlation between survey-based measures of presidential approval and the sentiment of tweets that contain the word "Trump". In doing so, however, we adjust smoothing and lag parameters to obtain the best possible correlation, as is typically done in similar analyses (O'Connor et al. 2010; Cody et al. 2016). Because we optimize over these parameters, it is difficult to interpret the strength of the resulting correlation. We therefore compare our observed correlation to other correlations that are calculated in a similar way, but which are assumed to be spurious. Using this framework, we conclude that while there may be a signal when tracking sentiment of tweets containing the word "Trump" with presidential approval, it is small and not obviously useful. These results cast doubt as to whether Twitter data can reliably be used as a replacement for traditional surveys.

Our second contribution deals with the method in which social media data are obtained. As an alternative to the commonly used method of simply collecting tweets that contain a given keyword (e.g. "Trump") irrespective of who is posting them, we propose following a set of politically active Twitter users over time. This method of collecting tweets is similar to Golder and Macy (2011), who tracked mood using up to 400 tweets for each of millions of users. By collecting tweets in this manner we observe change among a set of users. We classify politically active Twitter users as a Democrat or Republican and find evidence of a political signal when tracking the frequency and sentiment of these users' tweets around the 2016 U.S. presidential election.

This chapter is organized as follows. In section 2.2 we examine the previously observed relationship between sentiment of "jobs" tweets and consumer confidence, concluding that the original relationship was likely spurious. In section 2.3 we develop a framework for interpreting the strength of a relationship between messages containing a given word and survey responses. In section 2.4 we follow a set of politically active users over time, finding a political signal in the frequency and sentiment of their tweets over time. Section 2.5 concludes.

2.2 Consumer Confidence

In this section we further explore the relationship between sentiment of tweets containing the word "jobs" and consumer confidence, as originally reported in O'Connor et al. (2010). We first discuss the reseracher decisions made in finding this relationship, and then show how each of these decision affects the resulting

relationship. Much of this section can be found in Conrad et al. (2019).

2.2.1 Methods

Data Sources

A dataset of tweets from January 2007 through June 2014 containing the word "jobs" is provided by a third-party service Topsy, which provides the tweet text, user, and date and time the tweet was created. This is the same keyword as in O'Connor et al. (2010). Topsy has since bought out by Apple in 2013 and no longer provides this service. Topsy removed spam tweets, but the exact method of doing so is not publicly available. We omit tweets from 2007 since there were so few tweets in that year as Twitter was only starting to emerge as a social media presence. To reduce computational burden, we use a random sample of 500 tweets per day (or all of the tweets from a day if there are less than 500 "jobs" tweets).

Consumer confidence data is taken from the University of Michigan Survey of Consumers survey (SCA). The SCA conducts about 500 (mostly) telephone interviews for a national sample of U.S. adults. The monthly Index of Consumer Sentiment (ICS) is calculated using responses to five questions. These five questions are given in appendix B. In O'Connor et al. (2010) and Cody et al. (2016), Twitter sentiment was only compared to the overall monthly ICS. The five questions making up ICS range from personal finances to expectation of the U.S. economy; we look at responses to these individual questions. In particular, we are interested in the two questions: "Now looking ahead–do you think that a year from now you (and your family living there) will be better off financially, or worse off, or just about the same as now?" (which we refer to as the *self* question) and "Now turning to business conditions in the country as a whole–do you think that during the next twelve months we'll have good times financially, or bad times, or what?" (which we refer to as the *collective* question). While this consumer confidence survey data is publicly available on a monthly basis, through our affiliation with the University of Michigan we are able to access individual responses, which are gathered nearly daily.

Somewhat surprisingly, perceptions of society differ from aggregated individual circumstances since trends have not yet impacted individuals. As an example, asking individuals who they think will win an election often outperforms polls asking individuals who they will vote for (Graefe 2014). With this idea in mind, we hypothesize that sentiment from social media will match up closer to survey data measuring individuals' perceptions about society rather than measuring personal circumstances. That is, we expect that Twitter sentiment will be more accurate in predicting the *collective* survey question than the *self* survey question.

Classifying Tweets

Consider the number of "jobs" tweets per day (before downsampling to 500 "jobs" tweets per day) in Figure 2.1. The eight days with the most "jobs" tweets, in decreasing order, are: October 6, 2011; October 7, 2011; August 25, 2011; July 20, 2011; October 5, 2012; July 22, 2011; September 7, 2012; and October 17, 2012.

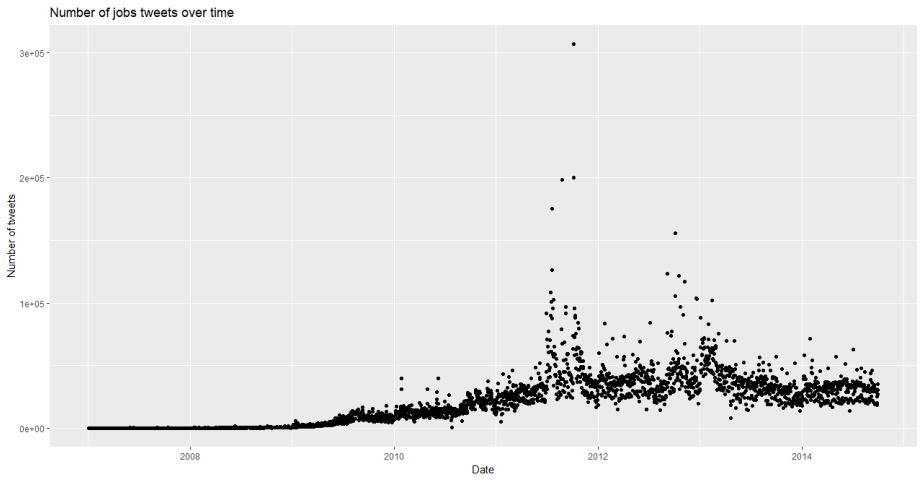

Figure 2.1: Number of "jobs" tweets from 2007 through mid-2014.

By looking at individual tweets from each of those day, it becomes obvious what causes the spike in number of tweets for each of those days.

Steve Jobs died October 5, 2011. The first two days with the largest number of "jobs" tweets are the two days immediately following his death. The fifth day with the largest number of tweets is the one year anniversary of Steve Jobs's death. Steve Jobs resigned as CEO of Apple on August 25, 2011, the day with the third highest frequency. These events should not have an impact on consumer confidence, but with the high volume of tweets in response to these events, it is likely to introduce unwanted bias to the sentiment of "jobs" tweets. For the purpose of finding some underlying relationship between Twitter sentiment and consumer confidence, these tweets are irrelevant and should not be taken into account.

On the fourth and sixth highest days, July 20 and 22, 2011, Twitter was spammed with many variants of 'Unemployed single mom makes $$$ from home. Check it out << *link* >>.' We consider these tweets to be junk tweets since they should be unrelated to the public's opinion on the economy.

The last two dates, September 7 and October 17, 2012, both have large amounts of tweets related to politics. Around September 7, a jobs report was released and many tweets were discussing job creation as a factor in the upcoming 2012 presidential elections. The second Obama-Romney presidential debate was held on October 16, during which a question about job creation was asked to each candidate. Many tweets on the 17th were in response to this. Unlike the previous examples with Steve Jobs and spam, "jobs" mentions on these days might actually be useful in determining the public's opinion on the direction of the economy.

The context of "jobs" mentions varies greatly. Some contexts we would expect to be related to various aspects of consumer confidence, while others we expect to be completely unrelated. The wide variety of "jobs" content may cause unwanted variation. Through inspection, we classify "jobs" tweets into five broad categories: *news/politics, personal, advertisement, junk* and *other*. Each of these categories is described below. In appendix C we provide example tweets that fall into each of these five categories.

1. **News and Politics**: This category of tweet generally refers to either current events on the national level or political opinions. Many of these tweets have to do with the U.S. economy as a whole.

2. **Personal**: Tweets in this category refer to one's individual job, many times commenting on job satisfaction or change in employment status.

3. **Advertisements**: Tweets in this category display jobs available in various fields and various cities. Many of these are through a 'Tweet My Jobs' third party service. Despite referring to actual jobs, we don't expect these tweets to have much relationship with consumer confidence since they do not provide any opinion.

4. **Other**: Tweets in the *Other* category are usually articles or lists, unrelated to current economic events, but typically having to do with employment in some way. For example, more articles may be written about recession-proof jobs during a recession.

5. **Junk**: The *jobs* mentioned in junk tweets refer to something other than employment. The most common include Steve Jobs, the TV show Dirty Jobs, and jobs of a sexual nature. Junk tweets should be independent of economic conditions and consumer confidence.

These five categories are not necessarily distinct; many tweets could easily fit into two or more categories. For simplicity, we assign each tweets to only one category.

In the appendix C we provide the detailed classification algorithm used to sort "jobs" tweets into one of the five categories mentioned above and provide evidence that it works as intended by comparing classifications as given by the algorithm to classifications given by manually.

Sentiment Calculation

Whereas survey responses directly measure sentiment, tweets do not. We score the sentiment of tweets using two broad methods: *aggregate scoring* and *individual scoring*.

Aggregate sentiment scores are only meaningful when examining many tweets across a given time period; these sentiment scores are not very meaningful at an individual tweet level. Aggregate methods are often dictionary-based, meaning they use a dictionary of words, each word being labeled positive or negative (or sometimes neutral). Because tweets contain relatively few words, many tweets contain no words in a given dictionary. Even if a tweet contains one or two words found in a dictionary, it is difficult to assign such tweets a meaningful continuous sentiment score. Instead, we use aggregate methods to assign individual tweets to sentiment categories, such as a positive tweet or negative tweet depending on the number of positive and negative words the tweet contains. We can also count the number of positive and negative words found in all tweets from a given time period. The overall sentiment for a given day is then calculated by using either the number of positive and negative tweets from that day or the number of positive and negative words found in all tweets from that day.

Dictionary-based method are the most straightforward method of sentiment analysis. While there are numerous limitations (such as a single word being both negative and positive in different contexts, inability to detect sarcasm), they are relatively easy to implement. Once we have the total number of positives or negatives (counted as either tweets or words) for a single day, we calculate overall sentiment for that day in one of three ways: (1) $\frac{positives}{negatives}$, (2) $\frac{positives-negatives}{total}$, or (3) $\frac{positives}{positives+negatives}$. Considering these multiple metrics using dictionary-based methods allows us to both reproduce previous results and check whether these seemingly small changes affect the resulting relationships.

We consider three dictionaries:

Lexicoder: Lexicoder was developed to measure tone in news content (Young and Soroka 2012). Lexicoder was trained on and shown to perform well on measuring the sentiment of political newspaper articles. The dictionary consists of 1,700 positive and 2,857 negative words. One difference between Lexicoder and the other two dictionaries in the inclusion of negation. That is, for each positive and negative word in the dictionary (e.g. "happy", "sad"), there is an associated negated term (e.g. "not happy", "not sad"). The final positive word count is calculated as the number of positive words + number of negated negative words - number of negative positive words. The final negative word count is calculated as number of negative words + number of negated positive words - number of negated negative words. We implement Lexicoder using the quanteda package in R (Benoit et al. 2018).

Liu-Hu. The second dictionary we consider is Liu-Hu (Hu and Liu 2004; Liu, Hu, and Cheng 2005). This dictionary was created using online product reviews, where customers often mention both positive and negative aspects of a product. Because this data comes from consumer reviews instead of professionally published words, it contains some common misspellings, and therefore may be better suited for analyzing Twitter sentiment. The dictionary contains 4,783 negative words and 2,006 positive words. We implement Liu-Hu through the SentimentAnalysis package in R (Feuerriegel and Proellochs 2018).

OpinionFinder. The final dictionary we consider is OpinionFinder, which was developed to evaluate a theory of polarity in lexical semantics (Wilson, Wiebe, and Hoffmann 2005). The OpinionFinder word lists consist of 1,600 positive and 1,200 negative words. Similar to Lexicoder, this dictionary does not contain slang or misspellings. This was the same dictionary as used in O'Connor et al. (2010).

Individual scoring methods, on the other hand, assign a continuous sentiment score to each individual tweet and typically make use of machine learning algorithms. These scores are meaningful and continuous on the tweet level; they indicate not just whether a tweet is positive or negative, but also to what degree. Scores are assigned using rule-based models based on lexical feature of the tweet. They often take into account not just words (such as content words, negated words, intensifier words), but nonword entries such as punctuation, capitalization, and emojis as well. We include two machine learning-based methods in our study:

Vader. Vader was trained on and shows to perform very well at evaluating the sentiment of individual short messages (Hutto and Gilbert 2014). It takes into account text features commonly found in tweets, such

as words, slang, negations, intensifies, punctuation, emoticons, and emojis. Vader assigns each individual tweets a sentiment score between -1 and 1.

TextBlob. The second machine learning-based method we use is TextBlob, a Naive Bayes classifier trained on the Stanford NLTK data set of movie reviews. TextBlob outputs a sentiment score of -1 to 1 for each individual tweets. Similar to Vader, it incorporates negation and intensifier words when calculating sentiment of tweets.

Overall sentiment for a single day, when using machine learning-based sentiment methods, is calculated as the mean sentiment of all tweets from that day.

Measures of Association

We measure the relationship between sentiment of tweets and consumer confidence using two methods: (1) correlation and (2) comovement, a measure of how often two time series move in the same direction from one time period to the next.

Person's correlation is the most commonly used method for assessing relationships between survey responses and Twitter sentiment. Before computing the correlation between Twitter sentiment and survey responses, we first smooth both the Twitter sentiment and survey responses. For each day in our time frame, we have an ICS score (based on the telephone interviews from that day) and Twitter sentiment (calculated using one of the methods mentioned above). Both of these time series are very noisy day-to-day. We introduce a smoothing parameter k and calculate the smoothed sentiment for a given day as the mean sentiment of that day and the previous $k-1$ days (therefore averaging k days total). We use the same k value for both time series. We also include a shift term L, which tells us whether Twitter sentiment leads or lags consumer confidence by L days. We then find the correlation between the smoothed Twitter sentiment and the smoothed and lagged survey responses.

As a technical note, the number of available data points used to find the correlation between smoothed Twitter sentiment and lagged survey responses decreases when smoothing and lag are added in this way. For example, if $k = 30$ we lose the first 29 days in the data since we cannot do 30-day smoothing unless we have tweets from the previous 29 days. If we shift survey responses up 30 days, we lose 30 additional days from the data. Overall, we lose $k+l-1$ data points. This is relevant because decreasing the number of data points used to calculate correlation can result in artificially inflated correlations, especially if we do not have many data points to begin with.

The second measure of association we use is comovement, which measures how often two time series move in the same direction from one time period to the next. While correlation takes into account the actual value of the time series on a given day, comovement only uses the direction of the differences. If we have T time units and two time series $x_1, x_2, \ldots, x_T$ and $y_1, y_2, \ldots, y_T$, we calculate the comovement as

$$comovement(x, y) = \frac{1}{T-1} \sum_{t=2}^{T} \mathbb{1}(sgn(x_t - x_{t-1}) == sgn(y_t - y_{t-1}))$$

where $\mathbb{1}(sgn(x_t - x_{t-1}) == sgn(y_t - y_{t-1}))$ is 1 if x and y move in the same direction from time period $t-1$ to t, 0 otherwise.

After shifting survey responses by L days, we calculate comovement on various timescales: daily, weekly, or monthly. Since comovement measures the percent of time two time series move in the same direction from one time period to the next, each time series can only have one value per time period. So depending on the timescale chosen, for each time series we only have one value per day, one value per week, or one value per month. Aggregating by time period with comovement is analogous to the k-day smoothing done with correlation.

We have some freedom in how exactly to calculate weekly and monthly comovement by choosing which day of the week or month to start on. For example, when calculating weekly sentiment, we can start the week on Sunday, meaning sentiment for a week is the average sentiment from Sunday through the following Saturday, or we can choose to start the week on Monday, meaning sentiment for a week in the average sentiment from Monday through the following Sunday, and so on. As a robustness check, we can start comovement on various days and compare the results. For example, if we start comovement on the 1st of the month compared to starting comovement on the 2nd, we don't expect a very large change in comovement if there is an underlying relationship between the two time series.

2.2.2 Results

Replicating Previous Results

We begin by replicating the analysis from O'Connor et al. (2010). Recall that in this paper, sentiment of tweets from 2008-2009 containing the word "jobs" was compared to consumer confidence as measured by the ICS. We attempt to replicate their results by using the same settings and time frame. Specifically, we calculate daily sentiment as the ratio of positive to negative tweets as measured using the OpinionFinder dictionary. A tweet is considered positive if it contains at least on positive word, similarly for negative tweets. Under this method of calculating tweet sentiment, a single tweet can be positive, negative, both positive and negative, or neither. We use the same smoothing parameter of $k = 30$ days and shift of $L = -50$ days. There are two main differences between our analysis and O'Connor et al. (2010): (1) a different corpus of "jobs" tweets (our corpus was provided by Topsy, their corpus was obtained using the Twitter API) and (2) ICS is computed daily in our study and monthly in theirs. O'Connor et al. (2010) found a correlation of 0.64; we find a correlation of 0.65. See top left cell of Table 2.1. Our replication succeeded.

Sorting by Tweet Category

We calculate the sentiment of the tweets sorted into each of the five content categories and, using the same settings as above, find the correlation between each of these categories and ICS. Results can be seen in the first column of Table 2.1. We expected the sentiment of tweets from the *news/politics* category to have

Category of Tweets	$\frac{positive\ tweets}{negative\ tweets}$	$\frac{positive\ tweets - negative\ tweets}{total\ tweets}$	$\frac{positive\ tweets}{positive\ tweets + negative\ tweets}$
All tweets	0.65	0.00	0.48
News/politics	0.17	0.30	0.19
Personal	-0.23	-0.30	-0.26
Advertisements	0.71	-0.24	0.32
Junk	0.42	0.16	0.32
Other	0.19	0.43	0.52

Table 2.1: Correlations between sentiment of tweet categories and ICS by how daily sentiment is calculated (using the OpinionFinder dictionary), based on settings in O'Connor et al. (2010).

the highest correlation with survey responses and *junk* and *advertisements* to have the lowest. This is self-explanatory for *junk* tweets: these tweets are by definition unrelated to the economy/jobs in the economic sense. *Advertisement* tweets are sent by accounts that specialize in posting job openings; they do not give any opinion about the state of the economy or job losses/gains. However, we find the opposite than what we expected. Correlation with *advertisements* (0.71) and *junk* (0.42) are much higher than with *news/politics* (0.17). The correlations with *personal* (-0.23) and *other* (0.19) are also not particularly strong. We assumed *advertisements* and *junk* tweets would be unrelated to employment, so these two high correlations may very well be spurious.

Robustness of Results

With many researcher decisions (e.g. choice of dictionary, the determination to count words vs. tweets, the particular smoothing interval chosen) contributing to the resulting correlation of 0.65 above, we are interested in how these decisions affect the observed correlation. To assess this, we adjust these parameters and compare the resulting correlations.

We begin by adjusting the method used to calculate sentiment. We use the three formulas as given in the "Sentiment Calculation" Section. Results can be seen in Table 2.1. We see that the choice of formula can drastically change the results. The most striking change in correlation with with all tweets, dropping from 0.65 to 0.00 when calculating sentiment as $\frac{positive\ tweets - negative\ tweets}{total\ tweets}$. Had O'Connor et al. (2010) used this scoring formula, they would have reached a drastically different conclusion: no relationship instead of a fairly strong relationship. The correlation with *advertisements* changes even more: from a strong positive relationship to moderately negative to moderately positive (0.71 to -0.24 to 0.32). Correlations with *news/politics* and *personal* tweets remained relatively constant and small. Clearly, the formula used to calculate sentiment of tweets can make a large difference.

We next explore the choice of dictionary and the difference between counting tweets versus counting words. Results can be seen in Table 2.2. We see that these decisions do not have a dramatic effect on the correlation with all tweets, with all of the correlations hovering around 0.6. Similar to what we saw in Table 2.1, the most dramatic effect of dictionary choice was with the *advertisement* tweets, ranging from 0.71 using OpinionFinder to 0.19 when using Lexicoder and counting words. In general, counting words versus tweets

Category	OpinionFinder		Lexicoder		Liu-Hu	
	$\frac{pos\ tweets}{neg\ tweets}$	$\frac{pos\ words}{neg\ words}$	$\frac{pos\ tweets}{neg\ tweets}$	$\frac{pos\ words}{neg\ words}$	$\frac{pos\ tweets}{neg\ tweets}$	$\frac{pos\ words}{neg\ words}$
All tweets	0.65	0.64	0.56	0.56	0.66	0.61
News/politics	0.17	0.10	0.30	0.39	0.15	0.18
Personal	-0.23	-0.25	-0.07	0.02	0.07	0.11
Advertisements	0.71	0.67	0.29	0.19	0.57	0.53
Junk	0.19	0.19	0.51	0.48	0.28	0.30
Other	0.42	0.33	0.56	0.36	0.49	0.43

Table 2.2: Correlations with ICS and tweets from 2008-2009 using various dictionaries.

Category	Vader	TextBlob
All tweets	0.54	0.18
News/politics	0.51	0.10
Personal	-0.05	0.22
Advertisements	-0.25	-.39
Junk	0.45	0.05
Other	0.64	0.60

Table 2.3: Correlations between sentiment of tweets using Vader and TextBlob and ICS from 2008-2009.

did not make a large difference in the resulting correlations. This is most likely due to the fact that tweets often only contain one word in a given dictionary due to the limitation on number of characters in a tweet.

As an alternative to the dictionary-based methods, we next look at the resulting correlations when using the machine learning-based sentiment methods in Table 2.3. Correlation between all tweets and ICS is 0.54 using Vader and 0.18 using TextBlob; correlation with *news/politics* is 0.51 using Vader and 0.10 using TextBlob; correlation with *irrelevant* tweets is 0.45 with Vader and 0.05 with TextBlob. The differences in size of these correlations presumably reflects the differences in the actual sentiment scores assigned to each tweet. This is evident in the modest correlation between Vader's and TexxtBlob's sentiment scores, $r = 0.54$.

Table 2.4 gives the correlations between many different sentiment methods. Many of these correlations are modest or low, suggesting that sentiment scoring tools–at least the five we consider in the table–are not interchangeable.

Lastly we compare results with different levels of smoothing and lag. Using the original settings as in O'Connor et al. (2010), we see how the correlation between all "jobs" tweets and survey responses changes as we adjust smoothing from $k = 1, \ldots, 100$ and the shift from $L = -100, -99, \ldots, 99, 100$ days. The resulting

	OpinionFinder	Lexicoder	Liu-Hu	Vader	TextBlob
OpinionFinder	1	0.564	0.471	0.424	0.264
Lexicoder		1	0.805	0.714	0.475
Liu-Hu			1	0.670	0.563
Vader				1	0.537
TextBlob					1

Table 2.4: Correlations between (unsmoothed) average daily sentiment of "jobs" tweets from 2008-2009.

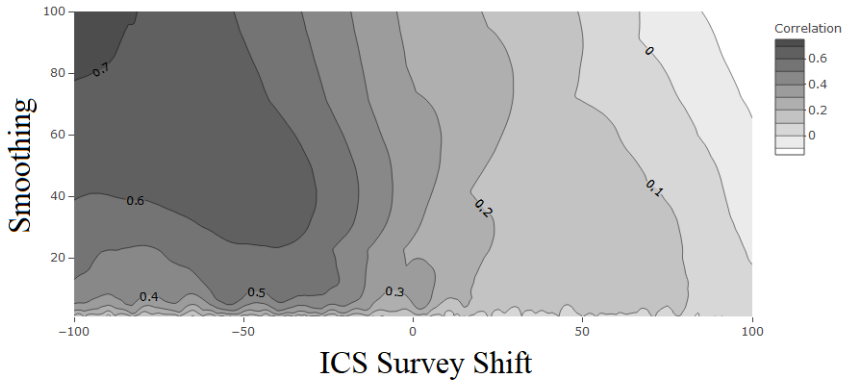

Figure 2.2: Effect of smoothing and lag parameters on correlation between ICS and all tweets from 2008-2009 using settings from O'Connor et al. (2010)

contour map can be seen in Figure 2.2. In general, under these settings, correlation increases as L changes from positive to negative, becoming largest (corresponding to the darkest region) when the lag is most negative, corresponding to social media preceding survey data in 100 days. However, we have no theoretical justification why such a large lag of over 3 months corresponds to a stronger correlation under the given settings. In particular, we would expect daily Twitter sentiment and daily ICS to be more or less aligned (i.e. a lag of around 0). While it may be the case that Twitter users form opinions at a somewhat different rate than the general population, we would expect this difference to be on the order of days, not months. The fact that we see the largest correlations with a shift on the order of months suggests that one should not read too much into these correlations as they may well be spurious.

We also note that while correlation tends to increase as smoothing increases, high levels of smoothing can artificially inflate correlation between two time series.

Comovement

We compute comovement using the same settings as O'Connor et al. (2010). Results are in Table 2.5. Using comovement there are no strong relationship between any Twitter categories and ICS survey responses at the daily or weekly levels; all hover around 0.5, what we would expect by chance. When calculating comovement on the monthly timescale using two years of data, there are only 23 monthly differences. With so few data points, it is easier to obtain more extreme comovements due to chance. At the monthly level, comovement varies depending on which day of the month we start on. If there is a true underlying relationship between survey responses and Twitter sentiment, we would not only expect comovement to be large, but also robust to starting date. Since that is not the case, it does not appear that there is a significantly strong relationship between survey response and Twitter sentiment.

Lastly, we note that for the two categories that we expected to be related to survey responses, *news/politics* and *personal*, the comovement is not particularly large for any timescale or start date.

Category	Daily	Weekly	Monthly 1st	Monthly 2nd	Monthly 4th
All	0.47	0.52	0.70	0.61	0.65
News/politics	0.46	0.53	0.39	0.35	0.35
Personal	0.46	0.47	0.65	0.65	0.61
Advertisements	0.43	0.40	0.48	0.52	0.57
Junk	0.52	0.54	0.57	0.70	0.57
Other	0.50	0.46	0.39	0.43	0.70

Table 2.5: Comovement between sentiment of tweets and ICS from 2008-2009 calculated daily, weekly, and monthly starting on the 1st, 2nd, and 4th day of the month.

Extension in Time

When we began this study, our motivation was to understand why the relationship between Twitter sentiment and survey responses deteriorated over time, raising the question of whether the relationship actually does weakened over time or simply started and remained volatile. To examine this, we compute the correlation for each year from 2008 through mid-2014 under the settings originally used by O'Connor et al. (2010) for 2008 through 2009. These correlations are displayed in Table 2.6. In some years, there is a very high correlation, which disappears or moves in the opposite direction the following year. Furtermore, there is no discernible pattern throughout. In particular, correlations do not slowly deteriorate over the years. If this had been the case, it could have suggested some systematic change in the Twitter data. Instead, the evidence suggests there never was a relationship to begin with.

Personal versus Collective Hypothesis

Conrad et al. (2015) hypothesized that a stronger relationship would be present when comparing Twitter sentiment to individual ICS questions as opposed the the overall ICS. Specifically, if people are writing tweets for others to read, like, and retweet, the content of the tweet might be more similar to questions about the national economy (collective) than about one's personal financial circumstances (self). Indeed, they observed a higher correlation with a collective question ("Now turning to business conditions in the country as a whole–do you think that during the next twelve months we'll have good times financially, or bad times, or what?"), $r = 0.84$, than with a self question ("Now looking ahead–do you think that a year from now you (and your family living there) will be better off financially, or worse off, or just about the same as now?"), $r = 0.39$. These correlation were measured over the years 2008-2011 in Conrad et al. (2015).

We revisit this main collective-vs-self hypothesis over a larger time period (2008-2014) and correlate survey responses with each category of tweet whose content may be relevant to the two survey questions examined. We expect the collective question to have a stronger relationship with *news/politics* tweets and the personal question to have a stronger relationship with *personal* tweets. We calculate sentiment as $\frac{\#\ positive\ words}{\#\ negative\ words}$ using Lexicoder (same as Conrad et al. (2015)) and use 30-day smoothing and 50-day lag (same as O'Connor et al. (2010)) Table 2.7 gives the correlations between each category of tweets and the two questions. There is no clear pattern throughout. Tweets about the U.S. economy (*news/politics*) are

Settings as in O'Connor et al. (2010):

Category	2008	2009	2010	2011	2012	2013	2014
All tweets	0.21	0.66	-0.03	0.54	0.02	0.28	0.41
News/politics	-0.05	0.18	0.22	0.37	-0.02	0.02	-0.61
Personal	-0.10	0.36	0.08	0.23	-0.07	0.09	-0.24
Advertisements	-0.02	0.64	0.01	0.59	-0.17	0.29	0.84
Junk	0.06	0.29	-0.21	-0.21	-0.16	-0.16	0.16
Other	-0.38	0.46	-0.57	0.67	0.02	0.53	-0.25

Vader:

Category	2008	2009	2010	2011	2012	2013	2014
All tweets	-0.18	0.71	0.21	0.62	-0.04	-0.10	0.35
News/politics	0.21	0.45	0.75	0.47	0.20	-0.15	0.48
Personal	-0.32	0.11	0.49	0.44	0.09	-0.21	0.34
Advertisements	-0.14	-0.69	0.28	-0.07	-0.37	0.20	-0.74
Junk	0.16	0.51	0.52	-0.31	-0.32	-0.16	-0.71
Other	-0.05	0.78	-0.49	0.76	0.03	0.17	-0.33

TextBlob:

Category	2008	2009	2010	2011	2012	2013	2014
All tweets	-0.45	0.39	-0.44	0.37	0.23	-0.21	0.07
News/politics	-0.10	0.28	0.40	0.16	0.33	-0.41	0.40
Personal	-0.24	0.18	0.05	0.28	0.02	-0.20	0.27
Advertisements	-0.25	-0.41	-0.23	-0.21	-0.06	0.13	-0.46
Junk	-0.13	0.01	-0.24	-0.51	0.18	-0.10	-0.39
Other	0.41	0.45	-0.43	0.59	0.27	0.07	0.61

Table 2.6: Correlations between ICS and Twitter categories for years 2008-2014 under settings used in O'Connor et al. (2010) (top), Vader (middle), and TextBlob (bottom).

Collective:							
Category	2008	2009	2010	2011	2012	2013	2014
All tweets	0.18	0.84	0.24	0.44	0.18	0.07	0.70
News/politics	0.16	0.25	0.68	0.50	0.21	-0.41	0.80
Personal	0.44	-0.14	0.60	0.33	0.35	-0.21	-0.19
Advertisements	0.06	0.65	0.39	0.20	0.15	0.12	-0.66
Junk	0.08	0.62	0.34	0.03	-0.38	0.14	0.16
Other	-0.30	0.72	-0.53	0.60	0.30	0.14	0.23
Self:							
Category	2008	2009	2010	2011	2012	2013	2014
All tweets	0.19	0.52	0.27	0.02	0.03	0.13	-0.36
News/politics	0.17	-0.16	0.59	0.28	0.14	-0.14	-0.53
Personal	0.15	0.20	0.37	-0.03	0.08	-0.17	-0.15
Advertisements	0.01	0.52	0.42	-0.02	-0.08	0.01	0.35
Junk	-0.08	0.41	0.40	0.32	-0.31	0.14	0.37
Other	-0.61	0.29	0.33	0.07	0.29	0.21	-0.04

Table 2.7: Correlation between Twitter categories and ICS questions by year. Collective on top; self below.

no more related to survey responses about the direction of the national economy than any other category of tweets, and tweets about one's own job (*personal*) were no more related to survey responses about one's own personal finances than any other category of tweets.

Concluding Results

When we began our investigation, we hoped to discover why previously observed relationships between Twitter sentiment and survey responses fall apart over time, and possibly find the proper settings to restore the relationship. Despite our optimism that filtering out irrelevant tweets, or using the correct sentiment method, or using a more robust measure of association would restore the relationship, we ultimately cast doubt on whether the originally observed relationship was present to begin with.

2.3 Presidential Approval

In the previous section we concluded that the observed seemingly strong relationship between sentiment of "jobs" tweets and consumer confidence is likely spurious. If data from social media are ever going to be used to serve as an adequate substitute for traditional surveys, there needs to be a method of interpreting the strength of the correlation between sentiment of social media sentiment and survey responses. It can be relatively easy to find highly spurious correlations with time series data, as demonstrated by Daas and Puts (2014) and Vigen (2014). Additionally, as we demonstrated in the previous section, there are several researcher decisions that contribute to the final observed relationship; optimizing over these decisions can lead to artificially strong relationship, and standard techniques for assessing relationship strength fail.

With the benefit of hindsight, it is perhaps not surprising that public opinion for select topics, such as the economy, can be difficult to obtain from social media. For example, even if a user's "jobs" tweet is

about the economy (as opposed to, for example, Steve Jobs), the user's opinion about the economy is not always clear from the tweet. Tweets about politics, on the other hand, are often quite clear with regard to who or what a users supports or opposes. Furthermore, there is evidence that non-probability online survey panels produce plausible estimates of Americans' political ideology (Kennedy et al. 2016). Therefore, if there is a strong, reliable signal present in Twitter data that might be used to supplement traditional surveys, we might reasonably expect to find it in the political realm. In this section, we focus our attention on tracking presidential approval, which we regard as the "best-case scenario" for the goal of using social media to supplement traditional surveys.

The contribution of this section is developing a method to interpret the strength of observed relationships between social media sentiment and survey responses. We calculate the correlation between presidential approval and sentiment of tweets from January 2017 through August 2019 that contain the word "Trump". Similar to analyses described earlier (O'Connor et al. 2010; Cody et al. 2016; Conrad et al. 2019), we adjust smoothing and lag parameters to obtain this correlation. To interpret the strength of such relationships, we do not want to compare our observed correlation to zero. Rather, we want to compare our observed correlation to spurious correlations between Twitter sentiment and presidential approval. We then ask ourselves how large our observed correlation is compared to the spurious correlations. That is, we implement a placebo analysis, most commonly used in econometrics, to assess the strength of the observed correlation. The idea behind placebo analysis is to replicate the primary analysis using variables that are unrelated to the outcome. As an example, DiNardo and Pischke (1996) revisit an analysis that claimed wage differentials were due to computer use in the workplace. DiNardo and Pischke (1996) found that when replacing the variable for computer use with pen/pencil use, the effect of pencil use on wage differentials was similar to the effect of computer use. This casts doubt on the original claim that computers in the workplace were causing the wage differential. The true effect for the placebo variable (pencil use) should be zero. The implication of a non-zero effect is that the original analysis was not credible, see Athey and Imbens (2017) for further details. We develop a framework to evaluate and interpret the strength of observed correlations between social media sentiment and public opinion surveys by essentially performing multiple placebo tests. Using this framework, we conclude that while there may be a signal when tracking sentiment of tweets containing the word "Trump" with presidential approval, it is small and not obviously useful. This result casts doubt on whether Twitter data can reliably be used as a replacement for traditional surveys.

2.3.1 Methods

Daily presidential approval is taken from the website FiveThirtyEight, a data journalism website which publishes articles on politics, public opinion, and sports-based statistical analyses. FiveThirtyEight fits presidential approval and disapproval trend lines based on multiple polls, weighting each poll by sample size and pollster ratings (based on a poll's historical accuracy in predicting elections and methodological standards) (Silver 2017). The trend lines are fit using a local polynomial regression model. Three quadratic regression

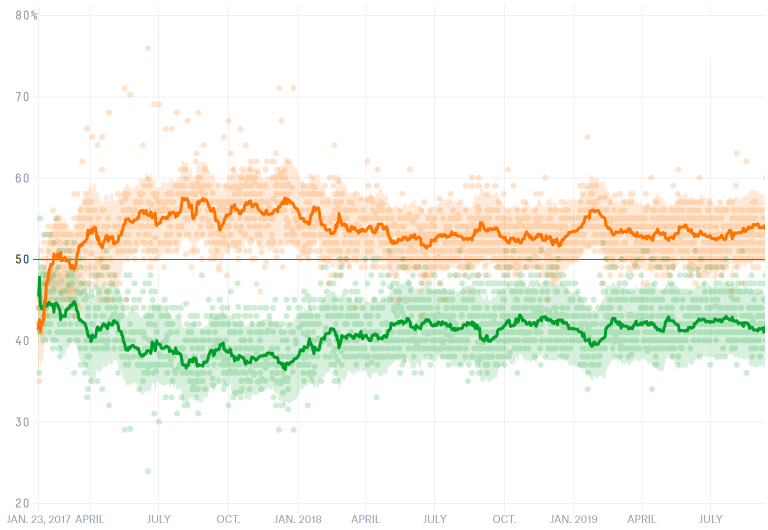

Figure 2.3: Daily presidential approval (green, bottom) and disapproval (orange, top) as given by FiveThirtyEight from January 2017 through August 2019.

models are fit with 10, 20, and 30 day bandwidth, giving three different estimates for presidential approval. These three estimates are averaged together to give the final estimate for a given day. FiveThirtyEight calculates two different trend lines, one for all adults and one for likely voters. The fitted approval and disapproval trend lines and all poll data used to calculate the trend lines are publicly available for download[1]. In our analysis we consider only the trend lines for all adults since Twitter users consist of all adults and not just those who are registered to vote. The fitted approval and disapproval trend lines for all adults, along with their respective confidence bands, can be seen in Figure 2.3.

During the time period from January 20, 2017 through August 25, 2019, we scraped 1000 tweets per day containing the word "Trump" using the Twitter API. This particular interval started with the first day of the Trump administration and covered the following 31 months.

Sentiment of tweets is calculated using Vader (Hutto and Gilbert 2014). Vader calculates sentiment of short social media messages through a rule-based model using lexical features of the message. These features include words, emoticons, acronyms, slang, punctuation, capitalization, degree modifiers, and contrast words. Each of these features has either a corresponding polarity and intensity of sentiment between -1 and 1 or an associated rule (e.g. negation words reversing the polarity of the following word). Vader combines these scores to give an overall sentiment score for each individual tweet between -1 and 1. We choose Vader as a sentiment tool as it was both trained on and shown to perform well in calculating the sentiment of individual tweets.

We calculate unsmoothed Twitter sentiment for a given day as the mean sentiment of all "Trump" tweets from that day. Unsmoothed sentiment is noisy day-to-day. This is not due to the fact that we have downsampled tweets, but rather a property of the population of tweets. Confidence intervals for sentiment

[1] `https://projects.fivethirtyeight.com/trump-approval-ratings/`

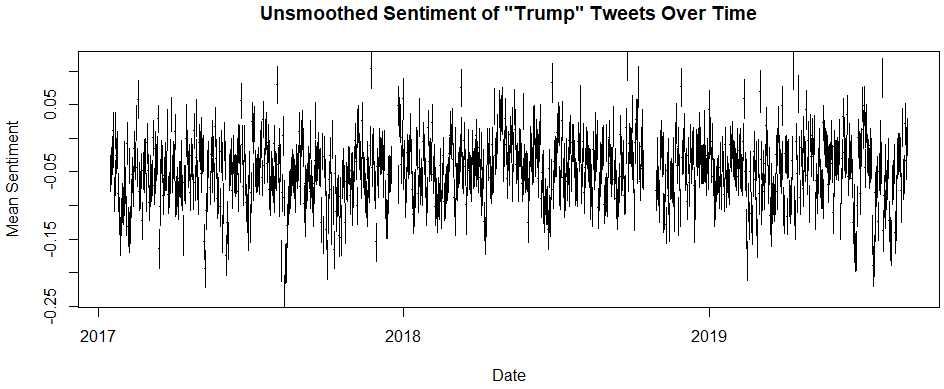

Figure 2.4: 95% confidence intervals for average daily unsmoothed sentiment of "Trump" tweets.

of "Trump" tweets often do not intersect from one day to the next, see Figure 2.4. We calculate smoothed daily Twitter sentiment for a given day by taking the average sentiment of that day and previous $k-1$ days. We introduce a lag term l, shifting survey responses ahead or behind by l days. This tells whether Twitter sentiment leads or lags presidential approval. We choose k to be in $\{1, 2, \ldots, 45\}$ and l to be in $\{-30, -29, \ldots, 29, 30\}$. We choose k and l such that we obtain the highest correlation between sentiment of "Trump" tweets and presidential approval. We choose k and l in this manner for three reasons: (1) it is not clear a priori whether social media lags survey responses or vice versa and it is not clear what the optimal smoothing might be, (2) we want to give the political signal the best chance of emerging, and (3) similar methods were performed in previous analyses (e.g. O'Connor et al. (2010) and Cody et al. (2016)).

After determining the smoothing and lag parameters that maximize the correlation, we then want to interpret the strength of the observed correlation. Autocorrelation and trends in time series data effectively reduces the sample size, making spurious correlations more common in time series data. To interpret the strength of the correlation, we want to take into account the relationship between sentiment of various subsets of tweets and presidential approval. If many of the subsets of tweets were more correlated with presidential approval than sentiment of "Trump" tweets, then we conclude the correlation we observed is spurious. Although this is similar to what was observed in Daas and Puts (2014) when looking at the relationship between sentiment of tweets containing the words "economy", "job", or "jobs" and consumer confidence, the conclusion is the opposite. When we compare against overall Twitter sentiment, it is a check for spuriousness.

We create a reference distribution using tweets containing everyday words. To define a set of everyday words, we use a random sample of 5000 tweets per day from the same time period and find words and symbols (such as emojis) that appear at least once every day. After removing stop words (e.g. "the", "an"), we are left with 495 such words. We call these placebo words, as the only relationship we expect to find between these words and presidential approval are assumed to be spurious. There are some "Trump" tweets in our random sample of all tweets, but they constitute a small percentage of our random sample. For each of these

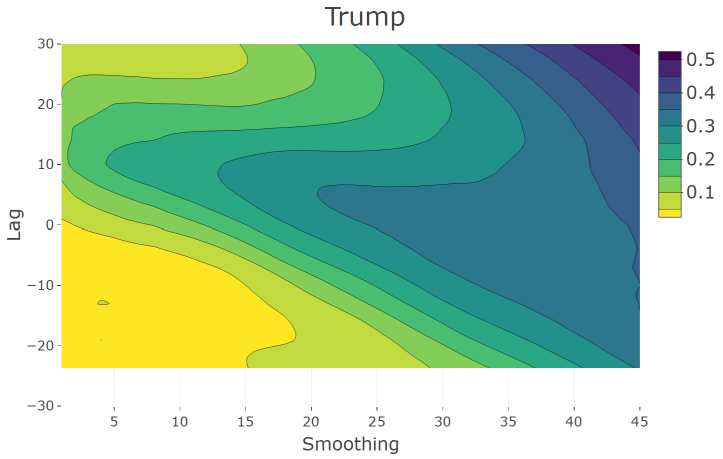

Figure 2.5: Correlation between sentiment of "Trump" tweets and presidential approval for various smoothing and lag values.

words, we repeat the same analysis as we did with the "Trump" tweets: using tweets that contain a given placebo word, adjust smoothing and lag such that we obtain the maximum absolute correlation between sentiment of tweets containing the placebo word and presidential approval.

Note that this framework can be used to evaluate the strength of any measure of association and any pre-processing of sentiment between messages containing some keyword and survey responses, not just correlation when adjusting for smoothing and lag in the context of presidential approval.

2.3.2 Results

Figure 2.5 shows how the correlation between sentiment of "Trump" tweets and presidential approval changes with various smoothing and lag values. With an optimal smoothing of 45 days and optimal lag of 30 days (meaning that Twitter sentiment lags presidential approval by 30 days), we obtain a maximum correlation of 0.516 between sentiment of "Trump" tweets and presidential approval. While this is not as high as was previously found using "Obama" tweets and presidential approval (as in O'Connor et al. (2010) and Cody et al. (2016)), the correlation of 0.516 nonetheless seems to suggest there is a relationship between "Trump" tweets and presidential approval from 2017 through mid-2019.

To demonstrate the placebo analysis, in Figure 2.6 we take a random sample of six placebo words and show how changing smoothing and lag parameters affect the correlation between those six words and presidential approval. Similar to "Trump" tweets, a higher level of smoothing often leads to higher correlations with these six random words. This is consistent with O'Connor et al. (2010), in which it was found that higher levels of smoothing of sentiment of "jobs" tweets leads to higher correlations with consumer confidence, as measured both by Gallup and the University of Michigan Index of Consumer Sentiment. Indeed, the optimal smoothing values often occur at the highest level of smoothing allowed in the windows, which is 45 in this case. Figure 2.7 shows where the optimal smoothing and lag parameters fall for each of the 495 placebo

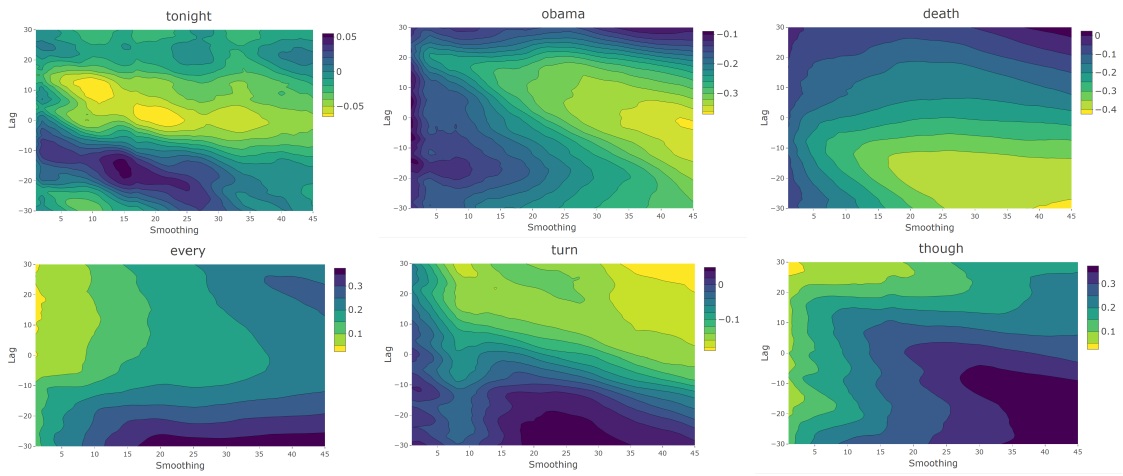

Figure 2.6: Correlation between sentiment of a random sample of six words and presidential approval for various smoothing and lag values.

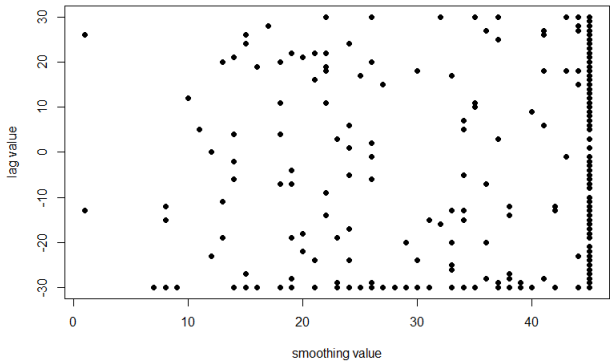

Figure 2.7: Locations of optimal smoothing and lag parameters between the 495 words and presidential approval. Each point represents where the maximum correlation occurs for one of the 495 words appearing in the Twitter corpus every day.

words. This is a cautionary message: too much smoothing can lead to artificially inflated correlations. This is the case with "Trump" tweets as well, and demonstrates why we cannot simply compare the observed relationship to 0.

For each of the 495 placebo words we find the most extreme correlation, i.e. the correlation with the maximum absolute value, within the given smoothing and lag values. The set of these correlations creates what we call the reference distribution. To assess the strength of the relationship between "Trump" tweets and presidential approval, we compare the observed correlation in relation to the reference distribution. If there truly is a relationship between sentiment of "Trump" tweets and presidential approval, the observed correlation should be much larger than nearly all of the placebo correlations. The reference distribution can be seen in Figure 2.8. The reference distribution is bimodal. This is because we manipulated the smoothing and lag parameters to find the optimal correlation between sentiment of tweets containing each

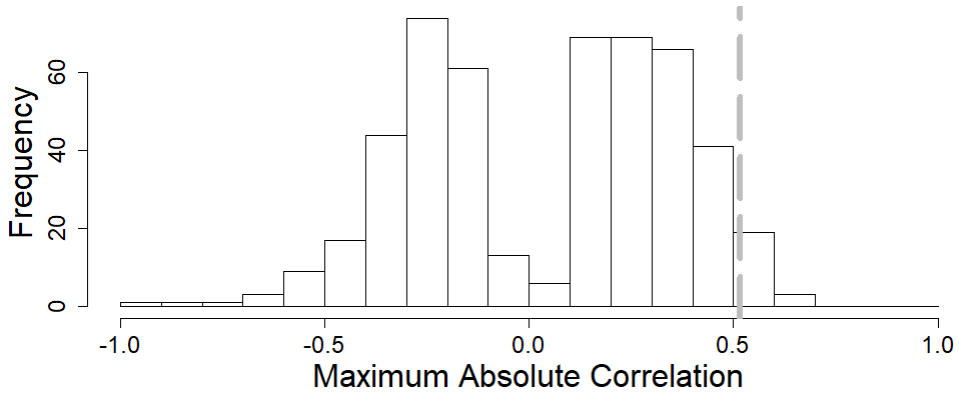

Figure 2.8: Reference distribution of maximum absolute correlations between presidential approval and sentiment of 495 placebo words with $k \in \{1, \ldots, 45\}$ and $l \in \{-30, -29, \ldots, 30\}$. Maximum correlation between sentiment of "Trump" tweets and presidential approval, 0.516, is denoted by the dashed vertical line.

of the placebo words and presidential approval. The vertical dashed line in Figure 2.8 is the correlation between the sentiment of "Trump" tweets and presidential approval, 0.516. This correlation is larger than many of the placebo correlations, but not considerably so. About 5.3% of the placebo correlations are larger in absolute value than the correlation between presidential approval and "Trump" tweets. However, none of the placebo words with maximum absolute correlations greater than 0.516 are meaningfully related to presidential approval, e.g. "giveaway", "17", "enough", "city", and "name" are the five words with the highest maximum absolute correlation with presidential approval. While there appeared to potentially be a signal, if anything it is a very small signal, a signal that cannot by itself predict public opinion.

Robustness over time

Throughout the time period of performing the analysis, we re-ran the analyses several times as newer data became available. Results often depend on the last data point available in the analysis, especially through early 2018. Consider finding the optimal correlation between sentiment of "Trump" tweets and presidential approval when the last data point available ranges from May 2017 to August 2019. For each of those end dates we find the smoothing and lag parameter that leads to the maximum absolute correlation. Figure 2.9 shows the maximum absolute correlation (thick line) and the correlation with 45 day smoothing and 30 day lag (dashed line) change over time. Figure 2.10 shows the optimal smoothing (top) and lag (bottom) values that produce the maximum absolute correlation as the end date of the data changes.

The reference distribution also changes as end date changes. Figure 2.11 shows how the proportion of placebo correlations that are more extreme that the correlation between sentiment of "Trump" tweets and presidential approval as end date changes. Around mid-2018, this proportion stabilizes to between 0.05 and

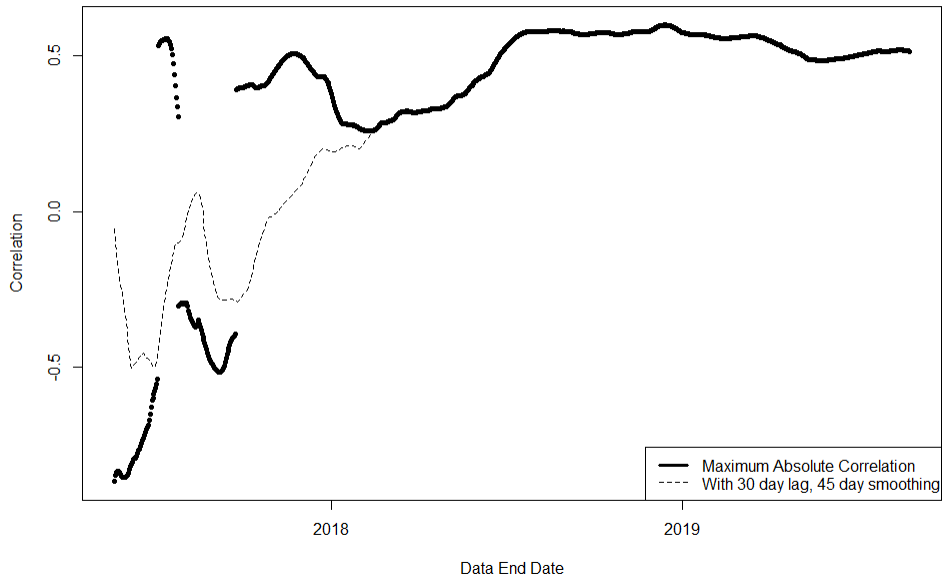

Figure 2.9: Maximum absolute correlation (bold) and correlation using 45 day smoothing and 30 day lag (dashed) as end date of data changes.

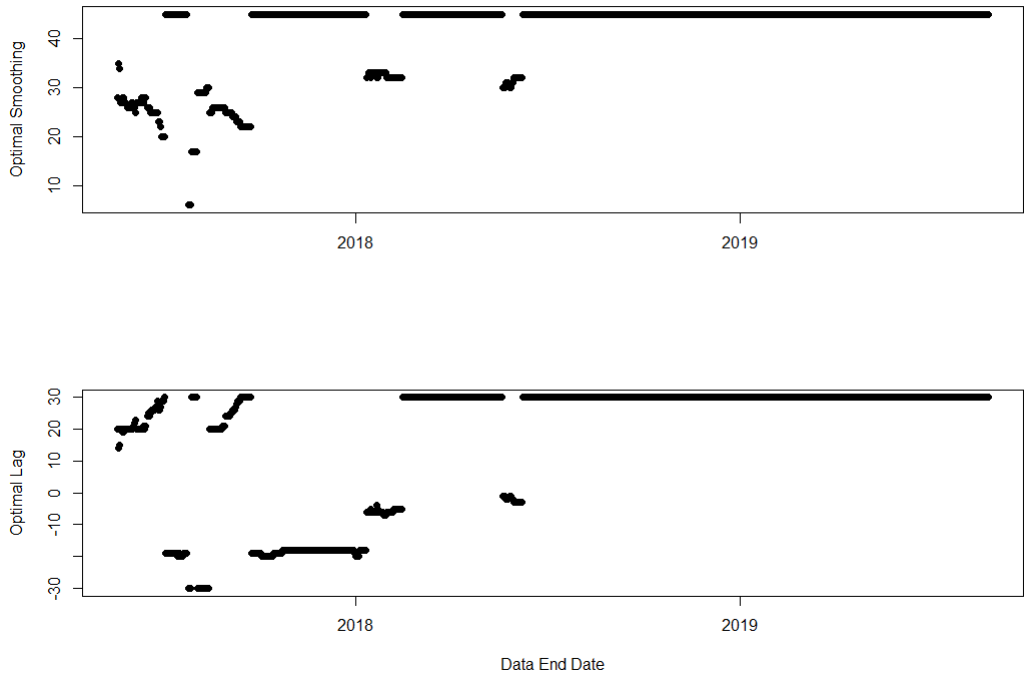

Figure 2.10: Optimal smoothing (top) and lag (bottom) parameters as end date of data changes.

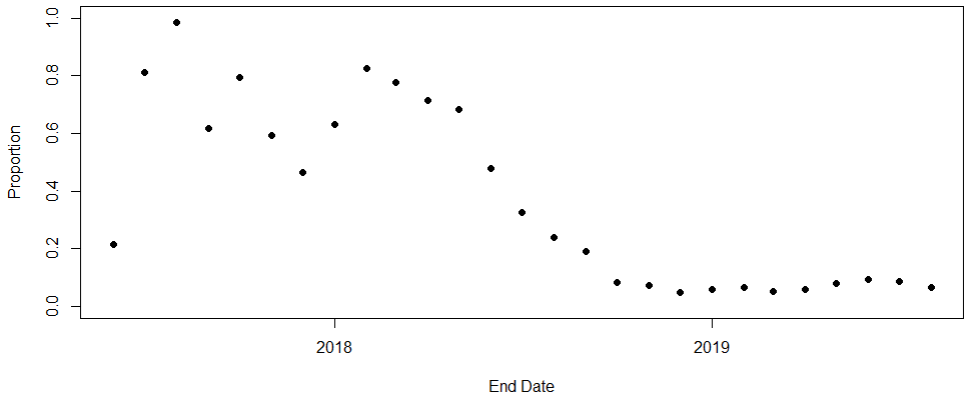

Figure 2.11: Proportion of placebo correlations more extreme than observed correlation between "Trump" tweets and presidential approval as end date changes.

0.10.

2.4 Longitudional Analysis of Politically Active Users

The results of the previous section raise concern on the utility of social media as a method of gauging public opinion. Our goal in this section is to determine whether we can convincingly detect any credible, non-spurious political signal in data extracted from Twitter. As an alternative to following the sentiment of tweets that contain the word "Trump" in a repeated cross-sectional design, we follow a group of politically active users over time. This idea is similar to Golder and Macy (2011), who tracked mood using tweets from a set of users. A longitudinal study performed in this manner may have several advantages. As noted in the introduction, when following the word "Trump" over time, we cannot be sure as to what extent the demographics of users tweeting about Trump are changing over time. We do not have this issue when following the same set of users longitudinally over time.

In this section we gather a set of politically active users and classify each of them into a political party based on profile information. We search for what we assume to be one of the largest signals on Twitter for this set of users: the outcome of the 2016 presidential election.

2.4.1 Methods

Identifying Politically Active Users

Using a corpus of tweets[2] provided by Sysomos, we classify tweets from 2016 as political or not political based on the words within each tweet. If a tweet contains at least one of the words "Obama", "Clinton",

[2]All tweets in the provided corpus contained the word "jobs" and was related to the data used in section 2.2.

"Trump", "Ryan", "McConnell", "potus", "teaparty", "democrat", "republican", "trade", "taxes", "senate", or "president", the tweet was classified as political. We created this list of political words by hand based on viewing the content of many tweets in our corpus. We then took a random sample of size 15,000 of the users whose tweet was classified as political and retrieved their 2016 tweet history through the Twitter API. We then identified 'politically active users' as follows. For each of these 15,000 users, we checked if the user produced at least 20 original tweets (non-retweets) in 2016, 10 of which contain at least one of the political words listed earlier. If so, we consider that user a politically active user. Note that under this definition of a political tweets, we surely have not identified all political tweets, but the tweets we identify as political are very likely to be political. We obtain 4189 politically active users using this definition.

Identifying Political Beliefs

Since our end goal is to find a political signal in the tweets belonging to our set of politically active users, we would ideally like to know each user's political party affiliation. We begin this process by creating a training set of users with known political affiliation from which we train a classifier. We identify a list of political words commonly found in users' self-provided profile description on Twitter ("conservative", "Trump", "MAGA", "NRA", "Constitution", "Republican", "Libertarian", "Democrat", "liberal", "Hillary", "Clinton", "Obama", "progress*", "Bern*", "resist*", "president"). If a user's self-provided provided profile description contained one of these words, we hand-classify the user as belonging to one of the two major political parties in the US: Democratic or Republican. These users were explicitly clear in their profile description about their political beliefs or about which candidate they did or did not support in the 2016 presidential election. We classify self-described libertarians as Republicans, and classify self-described socialists as Democrats. We classify Never-Trump Republicans as Republicans, and classify Never-Hillary Democrats as Democrats. This creates our training set of 170 Democrats and 393 Republicans.

The classifier for predicting political party is built using the list of Twitter accounts that the users with known political party follow. As predictor variables we use Twitter accounts that are followed by at least 30 of the users with known political party. There are 3040 such accounts, meaning we have 3040 binary variables (following or not) that are used to predict political party. A random forest is used as a classifier. Table 2.8 gives the classification error rates for the random forest. Out of the 170 users hand classified as Democrats, 160 were correctly identified as Democrats, and 388 out of the 393 users hand classified as Republicans were correctly identified as being Republicans. Overall, only 2.66% of users with known political party were incorrectly sorted by the random forest. Investigating the known Republican users who were misclassified revealed they were either self-described libertarians or outspoken anti-Trump Republicans. This is because there were relatively few of these users and they tended to follow both liberal and conservative accounts. Figure 2.12 gives the variable importance of the Twitter accounts used to classify. The most important accounts for classification are either politicians (e.g. BarackObama, realDonaldTrump, SenWarren, HillaryClinton, newtgingrich), political commentators (e.g. seanhannity, IngrahamAngle, maddow,

		Predicted Democrat	Republican	classification error
Actual	Democrat	160	10	0.090
	Republican	5	388	0.0085

Table 2.8: Random forest confusion matrix. Actual party affiliation corresponding to the hand classification; predicted party affiliation corresponding to the random forest out-of-bag prediction.

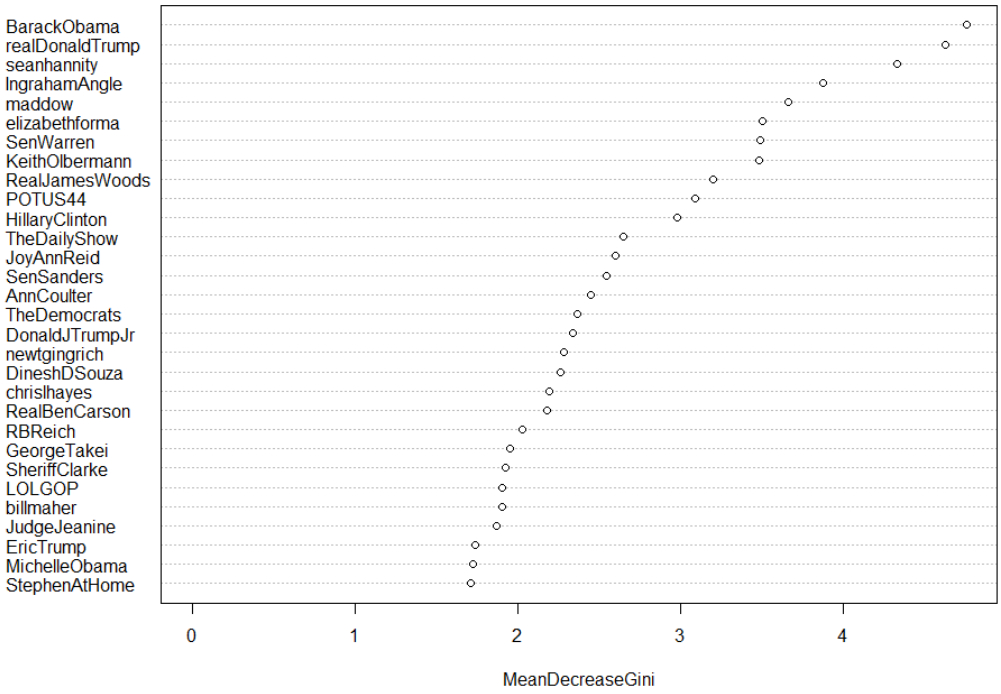

Figure 2.12: Variable importance of following accounts used in classifying users as Democrat or Republican.

TheDailyShow), or family members of politicians (e.g. DonaldJTrumpJr, EricTrump, MichelleObama). The random forest appears to find a true political signal within the 3040 accounts used to classify.

The trained random forest is used to predict political party for the remaining users with unknown political party. Since the users in the training set make their political opinions explicitly known in their self-provided description, they may have stronger political opinions or their political opinions may be more closely tied with their personal identity than the users with unknown political party. Therefore, it is possible that the users with known political affiliation are fundamentally different than the users with unknown political affiliation. When using the random forest to predict political affiliation of the remaining users, we want to both be fairly confident that users predicted to be in a certain political party are actually members of that party and have enough users in each party to detect a potentially small political signal. We choose an 80% cutoff rate to accomplish both goals. That is, a user is considered to be a Democrat if at least 80% of the trees predict the user the be a Democrat; similarly for Republican. This gives 489 total Democrats and 996 total Republicans that we use going forward.

Bots

The set of politically active users was created in mid-2017. Twitter has since deleted many bot accounts that had the goal of influencing other users' political opinions. We want to ensure that we have not gathered multiple bot accounts in our set of politically active users. We want the opinions of real people.

Out of the 1485 politically active users identified in mid-2017, 99 accounts were unable to be scraped again in May 2018. These are split fairly evenly across Democrats and Republicans: 7% of Republicans' and 5% of Democrats' tweets were not able to be gathered using the Twitter API in May 2018. However, this does not mean the account was a bot; users can choose to delete their account at any time, can make their account private, or have their account suspended by Twitter, all of which would result in the account being inaccessible using the Twitter API.

NBC published a list of 453 bot users and tweets from those bots (Popken 2018). None of these known bots were included in our list of Democrats and Republicans.

Metrics

The politically active users identified above do not tweet exclusively about politics. Some tweets are about their personal life and other interests (sports, entertainment, etc.). We found it difficult to hand classify these users' tweets as political or not political, much less create an algorithm to do so, since we do not know the intention of the user or the context in which the tweet was sent. Additionally, when a user retweets, we do not know if they are retweeting because they agree with the sentiment of the original tweet or are making fun of the original tweet/retweeting sarcastically. It has been found that users either retweet users who share very similar or very antagonistic views (Guerra et al. 2017). Thus, only original tweets are considered, and retweets ignored.

We consider two metrics to demonstrate that a political signal exists in tweets from 2016: frequency of tweets and sentiment of tweets. Frequency indicates whether or not our set of useres tweet about political events, and sentiment tells us their reaction to those events. For frequency we will look at the number of original tweets sent per user per day. For sentiment we continue to use Vader to calculate sentiment of original tweets (Hutto and Gilbert 2014).

2.4.2 Results

Figure 2.13 shows the frequency of original tweets for Democrats and Republicans from 2016 through mid-2017. The vertical lines on these plots represent election day (November 8, 2016) and inauguration day (January 20, 2017). The top four days with the highest frequency of tweets for Democrats, in order of frequency, are November 9, 2016; October 10, 2016; October 20, 2016; and September 27, 2016. These days correspond to the day after the election and the days after the three presidential debates between Hillary Clinton and Donald Trump. The top four days for Republicans are November 9, 2016; October 20, 2016;

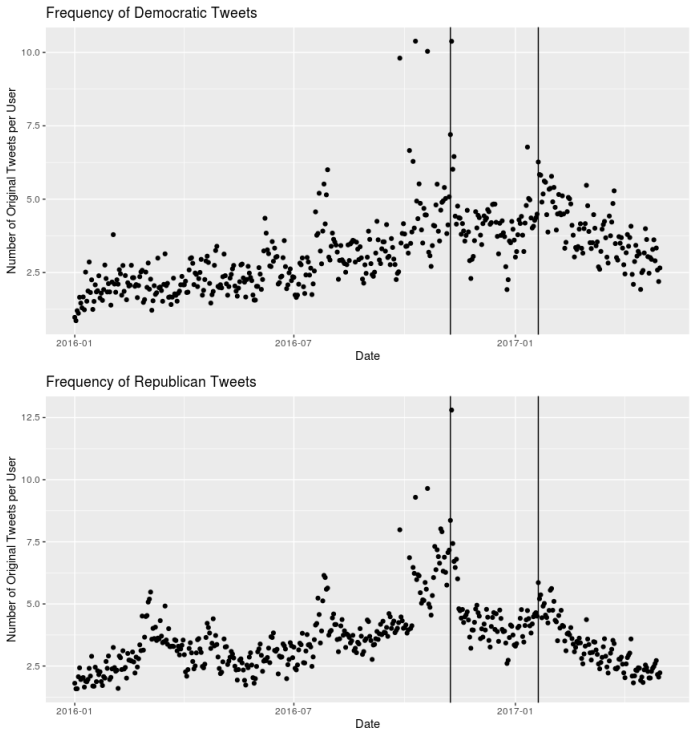

Figure 2.13: Average number of original tweets per day per Democrat (top) and Republican (bottom) from 2016 through mid 2017. Vertical lines represent election day (November 8, 2016) and inauguration day (January 20, 2017).

November 8, 2016; and October 10, 2016. These days correspond to the day after the election, election day, and days after the third and second debates. The frequency of tweets is clearly politically driven for both Democrats and Republicans.

We next consider sentiment of original tweets. We find that while frequency of tweets is mainly driven by political events, sentiment for both Democrats and Republicans is also driven by events outside of politics. Large daily spikes, i.e. increases, in average sentiment are due to holidays, such as Christmas and Thanksgiving (in late November), and a large daily drop is in response to a mass shooting, as can be seen in Figure 2.14. Many events that affect sentiment occur outside of the political realm. Therefore, with the idea that Democrats and Republicans react to holidays and tragedies with similar sentiment, we are also interested in the difference in sentiment between Democrats and Republicans. Figure 2.15 shows the daily difference in the mean sentiment of Democratic and Republican tweets from two months before the election through two months after the election. There is a clear drop the day after the election, and a general a change after the election, with Democrats generally happier before and Republicans happier after. Presumably because the election results were a surprise for many, there was a notable change in sentiment from the days leading up to the election compared to the days after as opposed to a gradual change.

To get a more detailed understanding of what is driving the change in difference in sentiment, we next look at how the positive and negative sentiments change over time. Figure 2.16 shows the difference in

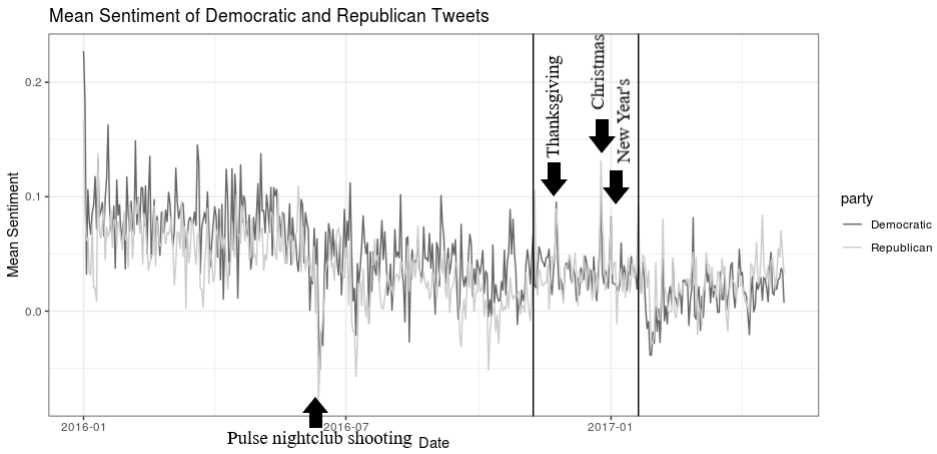

Figure 2.14: Average daily sentiment for Democrats (dark grey line) and Republicans (light grey line) from May 2016 through May 2017.

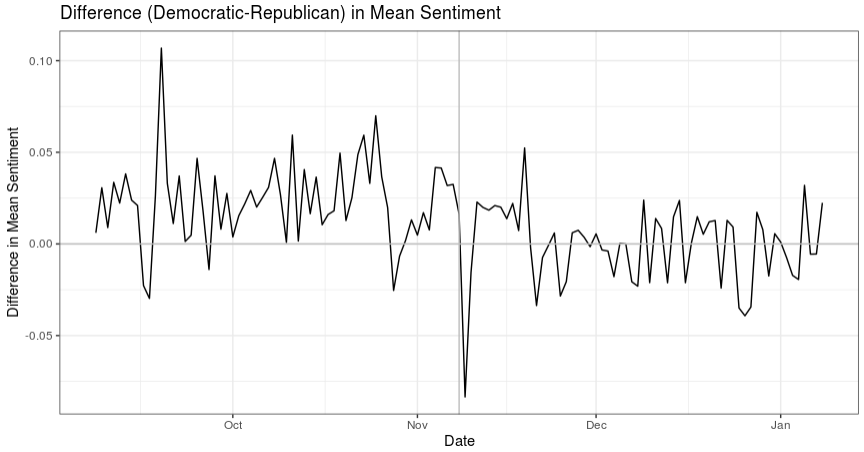

Figure 2.15: Difference in average sentiment between Democrats and Republicans (Democrat minus Republican) from two months before the election (September 8, 2016) to two months after the election (January 8, 2017). The vertical line is election day (November 8, 2016). The difference in sentiment is almost always positive before the election and often negative after the election.

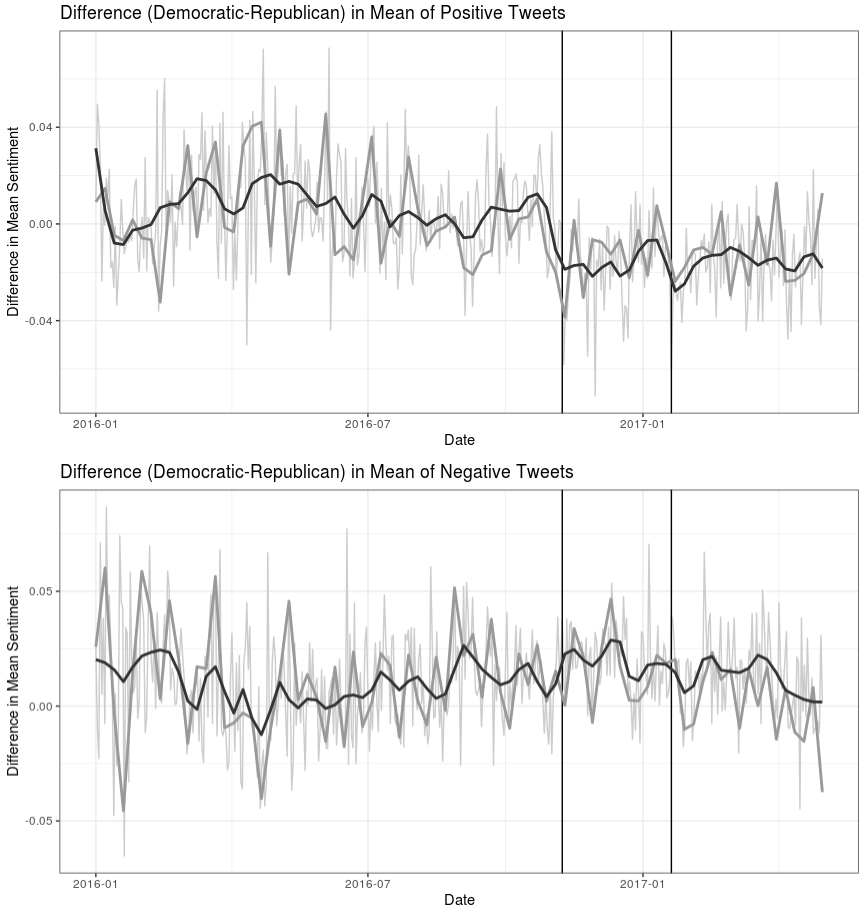

Figure 2.16: Difference in means of positive tweets (top) and negative tweets (bottom) for Democrats minus Republicans. The vertical lines are election day (November 8, 2016) and inauguration (January 20, 2017). The different shaded lines are for different smoothing levels to more easily see how sentiment changes over time.

means of the positive tweets and difference in means of the negative tweets for various smoothing levels. We see a clear drop in difference in positive means immediately following the election, and a small drop around inauguration. However, we do not see a similar change in negative means. The overall change in difference in sentiment was driven by Republicans' positive tweets becoming more positive post-election.

To summarize, analysis of both frequency and sentiment of politically active users' tweets suggest that tweets from the politically active users do indeed contain a political signal. This raises the possibility that following the users longitudinally may be a viable alternative for tracking public opinion as opposed to following the sentiment of tweets containing a given word under a repeated cross-sectional design.

2.5 Discussion

Summary of Results

If social media data is ever to be used to supplement or replace survey tracking public opinion, there must be sufficient evidence that the social media data is indeed a valid way of measuring public opinion. This includes evidence that we are indeed tracking the signal of interest, a high signal to noise ratio, and stability of the relationship over time. We address these issues in accomplishing our main goals: assessing the stability of a previously observed relationship, developing a framework to interpret the strength of an observed relationship between tweets containing some word and survey responses, and finding evidence of a political signal when following Twitter users longitudinally.

We had initially believed that the relationship between sentiment of "jobs" tweets and consumer confidence could be restored with the correct analytic techniques, such as using the correct sentiment analysis method or filtering out tweets irrelevant to the economy. To test this idea, we attempted to list the decisions that were made along the way. We classified "jobs" tweets into five different categories based on the content: *news/politics, advertisements, personal, irrelevant*, and *other*. This did not restore the relationship. In fact, we found the highest correlations with *advertisements* and *irrelevant* tweets, which we have to reason to expect to have any meaningful relationship with consumer confidence.

We also hypothesized that sentiment methods tailored to tweets might help to restore the relationship, since the writing styles in tweets is less formal. However, using Twitter-specific machine learning methods for sentiment calculation did not restore the relationship. Furthermore, seemingly small changes in how sentiment is calculated using dictionary-based methods resulted in large changes in the resulting relationship.

Overall, in our analysis of "jobs" tweets, we find that minor changes can have major impacts on outcomes. We found it is not difficult to create relatively large correlations by arbitrarily adjusting parameters, leading to deceptively encouraging results.

With this conclusion in mind, our goal for the second analysis was to develop a framework for interpreting the strength of these observed relationships, which we explored in the context of presidential approval. We found the correlation between sentiment of "Trump" tweets and presidential approval, 0.516, by optimizing smoothing of sentiment and lag between survey responses and tweets. We developed a framework to interpret the strength of this observed correlation by comparing it to 495 placebo correlation obtained by performing the same analysis, but with tweets containing placebo words unrelated to presidential approval. The correlation of 0.516 was not especially strong in comparison with the reference distribution. This shows that there is a high level of noise in Twitter data; many of the placebo correlations, which should consist of nearly pure noise, were as high as the correlation between "Trump" tweets and presidential approval. The resulting relationships were also not consistent over time.

As an alternative method to tracking tweets that contain the word "Trump" over time, we proposed following politically active users longitudinally over time. We found evidence of a political signal when

classifying users as Democrat or Republican based on the accounts they follow. When tracking the frequency of their tweets over time, we found a clear political signal, with frequency of tweets spiking around political events. The difference in sentiment between Democrats' and Republicans' tweets also changed immediately following the 2016 election. Noticeable changes in the tweeting patterns of our set of users around political events confirms that we are indeed capturing our political signal of interest. This is consistent with previous results that found events in Twitter data, for example frequency of "Obama" and "Romney" tweets leading up the 2012 presidential election (Barberá and Rivero 2015) and sentiment of "Obama" tweets spiking on Obama's birthday (Pasek et al. 2019). However, given that the election was what we assumed to be one of the clearest signals on Twitter for this particular set of users, the change in sentiment is relatively small.

While we only considered social media data extracted from Twitter, similar methods are applicable to data extracted from other social media platforms. For example, we can interpret the relationship between Reddit posts containing the word "Trump" and presidential approval using our placebo analysis framework. Following social media users from other platforms over time is also a valid and fruitful method of data collection.

Challenges of Future Work

Creating a post on social media is in many ways different from responding to a survey (Schober et al. 2016), so in hindsight maybe the initial seemingly optimistic results should have been met with more skepticism. All of these differences have the potential to introduce bias, and completely removing this bias from social media data may be a nearly impossible task. Going forward with these types of analyses, these differences must be addressed.

One such challenge in using social media data to track survey responses is that the population of social media users does not fully represent the general population. One suggestion is to apply methods that were developed for nonprobability samples to social media data. This can be difficult since demographic information of users is often not known. Methods exist for infering users' demographics in some cases, but they do not cover every demographic characteristic and far from perfect. Some of these categories include location (Ajao, Hong, and Liu 2015; Jurgens et al. 2015; Schulz et al. 2013), political affiliation (as demonstrated in section 2.4.1), income (Preoţiuc-Pietro et al. 2015), age (Antenucci et al. 2014), and gender (Antenucci et al. 2014). Pasek et al. (2018) adjusted consumer confidence survey responses to match the demographic characteristics of the Twitter population, but that did not strengthen the relationship.

One possible avenue for further research includes improving current sentiment methods for tweets. While there are sentiment methods created specifically for short social media messages, they may fail to accurately capture intended sentiment for different writing styles and need to be continually updated as the social media language evolves (Shen et al. 2018).

Another suggested line of work is taking the content of the tweets more seriously instead of relying on purely the sentiment and frequency, as using just the sentiment might lose too much information. Many

standard text modeling methods are developed with longer texts in mind. Tweets, however, contain auxiliary information that longer texts do not, such as when the tweet was sent, information about the user, and information about the popularity of the tweet (likes, retweets). This auxiliary data may be able to be utilized in the creation of newer text modeling methods for social media messages.

While there is no evidence that tweets containing a given word reliably track public opinion, we still believe there is potential for social media data to be utilized. The results of our longitudinal analysis suggest that there is a real signal in Twitter data, and a future line of work could make use of that signal. This may not be in a way that replaces traditional public opinion surveys, but rather supplements surveys. Smith and Gustafson (2017) provide an example of supplementing election polls with Wikipedia page views of candidates to more accurately predict election results. Many challenges lie ahead, but with the right methods, there is potential for social media data to improve upon traditional methods of capturing public opinion.

Chapter 3

Clustering-Based Topic Modeling for Short Texts

In a given corpus of tweets, there are likely to be many topics present. Some of these topics may be unrelated to the signal of interest, introducing noise and potentially bias. By correctly sorting tweets by topic, we hope uncover signals of interest. In this chapter we introduce a new clustering-based topic modeling algorithm to sort tweets into categories based on their content. First, distances between words are created based on how often two words appear together in the entire corpus. Then, distances between tweets are created using the distances between the words in the two tweets. A distance-based clustering algorithm is applied to the resulting distance matrices to reveal the latent topic for each tweet. This algorithm does not take advantage of any auxiliary information typically available in social media posts, and is therefore able to applied to any corpus of short texts. We apply this algorithm to a validation set of Twitter users that are known to tweet about different topics, a corpus of "jobs" tweets, and tweets from a set of politically active users.

3.1 Introduction

As one of the world's most popular social media platforms, Twitter is a source of breaking news and public discussion on nearly every topic. Tweet and user information are also by default publicly available for download through the Twitter API (and can be disabled if a user specifies). This makes Twitter a valuable data source for the analysis of societal reactions and how information disseminates through social networks. Twitter data has been used in a variety of applications, from tracking public opinion polls (e.g. O'Connor et al. (2010)), to capturing change in an individual's mental state (e.g. Resnik et al. (2015)), to predicting election outcomes (e.g. Tumasjan et al. (2010)). However, tweets are characterized by having short length (limited to 280 [formerly 140] characters), informal language, and noise, which makes analysis of tweets challenging. In analyzing Twitter data, tweets are typically sampled in some strategic way to reflect the given study.

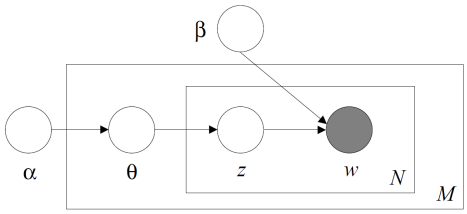

Figure 3.1: Plate notation for LDA, from Blei, Ng, and Jordan (2003).

For example, sampling tweets that contain a given word over time when tracking the relationship between tweets and survey responses (e.g. tweets containing the word "jobs" when tracking consumer confidence), or tweets from users that are politically active when searching for political signals in Twitter data. However, even the most strategic sampling of tweets will contain tweets on topics unrelated to the sought after topic. For example, "jobs" tweets contain tweets about Steve Jobs, and politically active users also tweet about non-political events. If we can accurately extract topics from tweets and filter unwanted topics, Twitter analyses can hopefully be improved, being better able to determine when users tweet about certain topics and what their reactions are to topical events.

Topic modeling is one of the most popular text mining methods and is used to extract common themes from bodies of texts, typically without human supervision. This allows one to more easily analyze and understand a large corpus of texts. Topic models assume some number of underlying topics, where a topic is often defined by the distribution of words found in that topic. Applying topic models to social media posts helps to better understand what users are discussing online and has been used in many fields, including physical health (Karami et al. 2018a) and mental health (Resnik et al. 2015).

The most common method for topic modeling of texts is Latent Dirichlet Allocation (LDA), a hierarchical Bayesian model introduced in Blei, Ng, and Jordan (2003). LDA assumes that each document in a corpus of texts is comprised of a distribution of topics, where a topic is defined as a probability distribution over words. More formally, let M denote the number of documents and N_i denote the number of words in document i. Document i has topic distribution θ_i, where $\theta_i \sim Dirichlet(\alpha)$. Each word w_{ij} in document i belongs to topic $z_{ij} \sim \theta_i$, and $w_{ij} \sim Dirichlet(\beta)$, where β is the prior on the per-topic word distribution. In this model, only the words w are known, the remaining variables are latent. Figure 3.1 gives the plate notation for this generating model. LDA typically works well with larger bodies of text, such as newspaper articles and books, but is less accurate with shorter bodies of texts due to the small number of word co-occurrences within each short text (Tang et al. 2014).

While standard LDA achieves mixed results when applied to tweets due to the small number of words in each document (tweet), there are several extensions of LDA that have been proposed to mitigate that problem. One such proposed method, and perhaps the most straightforward, is tweet aggregation. Tweet aggregation is performed as a processing step, with LDA (as well as other topic modeling methods) being

applied to the newly created documents consisting of multiple tweets. In the case of tweet aggregation by user, for example, all tweets sent by the same user are aggregated into a single new document, reducing the overall number of documents. Various methods of aggregation that have been proposed include by user (Hong and Davison 2010), hashtag (Mehrotra et al. 2013), conversation (Alvarez-Melis and Saveski 2016), and information retrieval information (Hajjem and Latiri 2017). While aggregating tweets removes the problem of applying topic models to small bodies text, aggregation methods are limited by the method in which the corpus was sampled and the auxiliary information available for each tweet. Additionally, these methods are not generalizable to short texts that are not from social media.

Zuo et al. (2016) use the idea of a pseudo-document, where each tweet is generated from a smaller number of larger latent documents; the aggregation method for this method is latent. Quan et al. (2015) suggest aggregating short texts into pseudo-documents using automatic clustering algorithms in combination with LDA, with each tweet assumed to be a text snippet sampled from the longer pseudo-document. These methods do not assume knowledge of auxiliary information for each text. Aggregating multiple tweets into new documents increases the word co-occurrence within each new document compared to the word co-occurrence in the original tweets.

Another method of mitigating the problem of few words per document in topic modeling is to artificially enrich the short documents using relevant texts to create longer documents. By adding more words to a text, within-document word co-occurrences increase, which increases the accuracy of standard topic models such as LDA. This is typically done using texts that exist outside of the corpus of texts being modeled. Bicalho et al. (2017) expand tweets (i.e. add words to tweets) using random draws of words weighted by how 'close' words in a dictionary are to words in a given tweet, using Wikipedia as an external data set to determine word relatedness. Jin et al. (2011) propose a Dual LDA model that performs LDA jointly on short texts and auxiliary related longer texts. However, it may not be clear a priori which external data set to use to enrich texts. In some cases, a proper external data set may not exist.

Standard LDA models assume that each document is a mixture of several topics, with the topic of each word in a document being drawn from some multinomial distribution. However, given the nature of tweeting and that tweets are short and contain few words, it may be more realistic to assume that each tweet belongs to only one topic. This is known as the unigram model (Nigam et al. 2000). There are variants of the LDA model that make this assumption, such as the Twitter-LDA model introduced by Zhao et al. (2011), which assumes the topic of each tweet is drawn from a distribution of topics unique to each author. However, this method assumes multiple tweets from the same author, which is not always the case in a given corpus of social media posts. Li et al. (2018) assume that each tweet contains only one functional topic, but also contain 'common topics', which refer to common and uninformative words (e.g. 'haha', 'rt' for retweet) that are spread across functional topics.

Another method for extracting topics from texts is Latent Semantic Analysis (LSA) (Landauer, Foltz, and Laham 1998; Deerwester et al. 1990). LSA creates a term-document matrix. This matrix is often quite

sparse, especially for shorter texts, so SVD is used to de-sparsify and reduce the dimension of the matrix. Documents are compared to one another by taking the dot product of the normalizations between vectors, with similar documents taking values close to 1 and dissimilar documents taking a value close to 0. However, LSA has been shows to perform worse at extracting topics from tweets than LDA (Qomariyah, Iriawan, and Fithriasari 2019).

We assume a unigram model, where each tweet is assumed to belong to only one of several latent topic. Our goal is to determine which topic each tweet belongs to. That is, we wish to discover the latent topic variable for each tweet. With that goal in mind, we approach topic modeling of tweets as a clustering problem. This is similar to Karami et al. (2018a), who use fuzzy clustering to determine topics of texts; and Quan et al. (2015), who use clustering within the context of LDA; and Xu, Liu, and Gong (2003), who cluster documents using matrix factorization. To generalize to multiple settings, we do not assume any auxiliary information is know for the tweets (similar to many others, such as Karami et al. (2018a), Yan et al. (2013), and Quan et al. (2015)). In this case, all we have to go off of are the words in the tweets themselves. Typical topic modeling methods, such as LDA, do not work well when applied to short texts since each individual document has sparse word co-occurrence (Wang and McCallum 2006; Boyd-Graber and Blei 2010). To overcome this, we use word co-occurrences found within the entire corpus, similar to Yan et al. (2013). Using corpus-level word co-occurrence, we create a measure of distance between words found in the corpus. We then find the distance between tweets based on the distance between each of the words found in the tweets. An unsupervised clustering algorithm is then applied to the distance matrix of tweets to uncover the latent topics for each tweet in the corpus. As far as the author is aware, calculating distances in this manner and the use of distance-based clustering is novel in the field of short text topic modeling.

This chapter is organized as follows. In section 3.2 we present the topic modeling algorithm. In section 3.3 we apply our method a validation set to confirm that our algorithm works as intended. We then apply the algorithm to a corpus of "jobs" tweets and tweets from politically active users. Section 3.4 concludes.

3.2 Method

Many topic modeling algorithms assume each document is a mixture of several topics (e.g. LDA (Blei, Ng, and Jordan 2003)). While this is a realistic assumption for longer bodies of text, we believe that the unigram model is a more realistic assumption for analysis of Twitter data, where each tweets belongs to only one topic (Nigam et al. 2000). With this idea in mind, we approach topic modeling of tweets as a clustering problem, where the goal is to unmask the latent clusters each tweet belongs to, similar to Karami et al. (2018b) and Xu, Liu, and Gong (2003).

Aggregation methods for tweets rely on tweets being sampled in a specific manner, such as having multiple tweets for each user, and do not work well when tweets are collected in a different manner. For example, tweet aggregation by user will not be effective when sampling tweets containing a given word, as many users

in the corpus will only have one tweet containing the given word. We want our topic modeling method to be applicable to any collection of tweets, or any collection of short texts, so we do not take advantage of auxiliary information of tweets, such as user and time sent. We use only the words found within each tweet. The intuition behind our method is that tweets belonging to the same topic will in some sense be 'close' to one another. Since tweets consist of words, if two tweets are from the same latent topic, their words will be in some way similar. So if we know the 'distance' between words, we can use those distances to find the 'distance' between tweets. This is similar to the ideas behind latent semantic analysis (Landauer, Foltz, and Laham 1998; Deerwester et al. 1990), where distances are found between texts based on the singular value decomposition of a term-document matrix.

In this section we describe our topic modeling algorithm. The general steps of our algorithm, which we describe in greater detail below, consist of pre-processing tweets, creating a document-term matrix, creating a word co-occurrence matrix, creating a word distance matrix, creating a tweet distance matrix, and clustering on the resulting distance matrix.

3.2.1 Pre-Processing

Similar to many text analysis techniques, the first step we take is to pre-process the tweets. This is not a necessary step for he rest of the algorithm to run, but make results more accurate and interpretable. These processing steps can be altered and chosen as one sees fit given the corpus at hand. We generally process tweets in the following ways:

- Convert every letter to lowercase. This is so a word is considered the same whether it starts a sentence (i.e. capitalized), in all uppercase for emphasis, or in all lowercase in the middle of a sentence.
- Removing certain words and symbols. We remove the '#' and '@' symbols, urls, and stopwords. We remove stop words (such as 'the', 'an') since they presumably show up fairly often in nearly every topic; two tweets should not be considered 'close' just because they both contain the word 'the'. We are interested in the words relating to the content of the tweet.
- Stemming. Stemming attempts to get at the base of the word, typically removing suffixes such as '-s' and '-ing'. For example, 'walked', 'walks', and 'walking' are all stemmed to the base word 'walk'. Stemming does not always work perfectly, but for our purposes it does well enough.

These steps also reduce the run time of the algorithm since multiple original words can be condensed into one word (e.g. through stemming) and certain words are deleted (e.g. stop words and urls).

3.2.2 Term-Document Matrix

Once tweets are pre-processed, we create a term-document, or word occurrence, matrix W. Each row of W represents a single tweet, and each column of W represents a word. w_{ij} is the number of times word j appears

in tweet i. Creating this term-document matrix is a very common step present in nearly all bag-of-words models.

Since tweets contain few words, W is very sparse. There are often many words that appear less than a handful number of times. This could be for a variety of reasons, such as uncommon words, misspellings, or words in a different language. We remove these words from the analysis. That is, we remove the columns whose sum is less than some specified number. We generally set this number to be small, so the only words that are removed do not carry much information. After removing these columns, we remove rows whose sum is 0. This updates the term-document matrix W, with n rows (i.e. tweets) and v columns (i.e. words and symbols). In doing this, some tweets will be removed from the analysis and not be assigned to a topic. Similar to Yang et al. (2014), we assume some tweets are 'pointless babble'. These tweets have no meaningful content, so by removing these tweets we are essentially removing noise. In practice, removing 'pointless babble' tweets results in the removal of only a small percentage of tweets.

The term-document matrix follows a bag-of-words model, where the ordering of words does not matter; the only thing that matters is whether or not a word appears somewhere in a tweet. Alternatively, word occurrence can be counted using a sliding window, where words are considered to co-occur only if they are within the window of each other (Zuo, Zhao, and Xu 2016). Transformations and reweightings can be applied to the term-document matrix. Some of these transformations give weaker weights to more common words or correct for differences in document length. These types of transformations are commonly referred to as *tf-idf*, or term frequency-inverse document frequency. Berry and Browne (2005) give some common transformations of the term-document matrix. Yan et al. (2012) introduce a *normalized cut* term weighting for clustering short texts based on if-idf.

3.2.3 Co-occurrence Matrix

Conventional topic models tend to fail with short texts because of sparse document-level word co-occurrence (Wang and McCallum 2006; Boyd-Graber and Blei 2010). For this reason, instead of focusing on document-level word co-occurrences, we focus on corpus-level word co-occurrences. Yan et al. (2013) are one of the few who also take this approach, modeling *biterms*, or unordered word pairings. Yan et al. (2013) directly model the biterms, whereas we use the biterms as the foundation for finding the distance between words found in the corpus's vocabulary. The reasoning for using co-occurrence of words is that if two words often appear together in the same tweet, they are likely from the same underlying topic.

Using the term-document matrix W, we create the $v \times v$ co-occurrence matrix C, where c_{ij} represents the number of times words i and j appear in the same tweet together. C is a symmetric matrix and c_{ii} is the number of tweets in the corpus that contain word i.

3.2.4 Word Distance Matrix

Next we create a measure of distance between words using C. The idea behind the distance between two words is fairly simple: if two words appear together frequently, they are likely to be from the same topic. We do recognize that it is possible for a word to belong to multiple topics. This is especially prevalent when a single word has multiple meanings. For example, the word 'play' can be in reference to a Broadway play or a sports player. Nonetheless, we create distance measures based on how often two words appear together in tweets. In our example, the word 'play' would end up closer to words in two different clusters: Broadway words and sports words.

We create a distance matrix $D_{v\times v}$, where d_{ij} is the distance between words i and j. This is similar to Pedrosa et al. (2016) and Bicalho et al. (2017). As D is a distance matrix, we require D to be symmetric. We create D from the corpus-level word co-occurrence matrix C. There are many possible distance measures; we give two of them below.

The first distance measure between word i and word j involves the conditional probability of observing word i in a tweet given that word j is in the tweet, and vice versa.

$$d_{ij} = 2 - [P(word\ i \in tweet | word\ j \in tweet) + P(word\ j \in tweet | word_i \in tweet)]$$

Using this conditional probability measure of distance between words, each entry of D ranges from 0 to 2, where $d_{ij} = 2$ if words i and j never appear together in the same tweet and $d_{ij} = 0$ if words i and j always appear together in the same tweet. We use this method exclusively for the analyses in the chapter.

Another method for calculating distance between tweets, using the Jaccard index as described in Pedrosa et al. (2016) and Bicalho et al. (2017), is

$$d_{ij} = 1 - \frac{(\#tweets\ with\ word\ i\ AND\ j)}{(\#tweets\ with\ word\ i\ OR\ j)}$$

The above distance measures are two of many different possible distance measures available. Transformations of the above distance measures are also a valid option for calculating distance between words. The main goal we try to achieve for the distance measure is that it gives a smaller value when two words are from the same topic, and a larger value when two words are from two separate topics.

Since the distance between words is determined by the co-occurrence matrix, the distance between words is dependent on the corpus itself. Words i and j can be close in one corpus, and distant in another. If co-occurrence or word distance was based on outside text documents, such as Wikipedia pages, the distance between words would not depend on the corpus, but on the outside texts chosen. It is not always clear a priori which outside text is most appropriate for a given corpus, and, in fact, an appropriate outside text may not exist for a given corpus. Choosing the wrong outside text may make the results in less accurate results. For these reasons we do not use outside texts in determining distance between words found in the

corpus.

3.2.5 Tweet Distance Matrix

Using the word distance matrix D we create distances between each tweet. Call this matrix $S_{n\times n}$, where s_{ij} is the distance from tweet i to tweet j. If two tweets are 'close' (i.e. belonging to the same topic), we would expect *all* the words in one tweet to be fairly related to *all* words in the other tweet To calculate the distance between tweets i and j, we restrict D such that the columns are words found in tweet i but not j, and rows corresponding to words found in tweet j but not i. We restrict d in this way so a word that appears in both texts but with different meaning does not make the two tweets artificially close. Then let s_{ij} be the mean of this restricted matrix. To give an example, for tweets t_1 and t_2 consisting of words $w_1, w_2, \ldots, w_5$, let $t_1 = w_1w_2w_3w_4$ and $t_2 = w_2w_4w_5$. Then

$$s_{12} = mean \begin{bmatrix} & w_5 \\ w_1 & \begin{pmatrix} d_{1,5} \\ d_{3,5} \end{pmatrix} \\ w_3 & \end{bmatrix} = \frac{1}{2}(d_{1,5} + d_{3,5})$$

If either dimension of the restricted matrix is 0, let $s_{ij} = 0$. This happens in the case where one tweet is a subset of another tweet and when two tweet are exactly the same, which happens in the case of retweets.

3.2.6 Clustering

Now that we have distances between all the words and tweets in our corpus, we find latent topics. Unlike other topic modeling methods that perform clustering, we do not have coordinate values for each tweet, but rather the distance between them. To uncover the latent topics we perform clustering techniques on the distance matrices. We use k-medoids (also known as Partitional Around Medoids, or PAM). K-medoids is similar to k-means clustering, but with observations being centers of the clusters instead of coordinate values (Kaufman and Rousseeuw 1987). This algorithm has fairy quick runtime (Schubert and Rousseeuw 2019).

Because we have distance matrices between both words and tweets, we are able to find latent topics in both words and tweets. However, a given cluster in words does not necessarily have an analogous topic in tweets. An alternative is to perform clustering on the tweets, and then find the word distribution across clusters. This outputs word distributions over topics that is comparable to other standard topic modeling methods such as LDA.

Since we have the option of performing clustering on both the words and tweets (i.e. rows and columns of the term-document matrix), connections can be drawn to biclustering. Biclustering is a technique in which clustering is performed simultaneously on rows and columns of a matrix (Hartigan 1972; Mirkin 2013). Since the term-document matrix W is very sparse, applying biclustering algorithms (e.g. Cheng and Church (2000),

Dhillon (2001), and Dhillon, Mallela, and Modha (2003)) directly on W might not yield accurate results due to low word co-occurrence within each short text. This is the same reason why standard LDA fails on short texts.

Clustering assigns each tweet (or word) to a single latent topic. The number of topics k is chosen by the user. There are several ways to determine the optimal number of clusters k. One such way is the elbow method, where the total within-cluster sum of squares in plotted for various values of k and we look for a bend in the curve. However, this method can be ambiguous, with the optimal number of clusters k not always clear. Other methods include using the gap statistic (Tibshirani, Walther, and Hastie 2001), which compares the within-cluster variation to what is expected under a null distribution, and the silhouette coefficient (Rousseeuw and Kaufman 1990).

The output of the k-medoids algorithm assigns each tweet to one single topic, matching our unigram assumption. However, it is possible to relax the unigram assumption and allow tweets to belong to multiple topics. Karami et al. (2018a) use fuzzy clustering to find the degree to which a tweet belongs to each topic. Karami et al. (2018a) do not use corpus-level co-occurrence, but SVD on a weighted word-document matrix as coordinate data points for each tweet.

3.3 Results

We apply our topic modeling algorithm to three corpora of tweets. With the first, we seek to validate the algorithm, demonstrating the accuracy of our algorithm with users we know tweet about very different topics. For the second, we apply the topic modeling to the corpus of "jobs" tweets from section 2.2. Lastly, we apply the algorithm to the corpus of tweets from politically active users from section 2.4.

3.3.1 Validation With Control Users

To assess the performance of our topic modeling algorithm, we apply the method to a set of tweets from a hand-selected set of users that we know tweet about very different topics. Specifically, we obtain tweets from four Twitter users: University of Michigan Football (UMichFootball), Vegan Cooking (vegancook101), Planned Parenthood (PPFA), and AccuWeather (breakingweather). These are tweets that a human could fairly easily classify as belonging to the correct user. While each of these users tweets about a single topic in general, these topics contain subtopics, and we assume there to be relatively little user overlap in the larger topics. We have nearly 3200 of the most recent tweets from each of the four users scraped from the Twitter API in R.

We begin with 12780 tweets from the four users, consisting of 12063 unique words after pre-processing. We keep words that appear at least 20 times, leaving 1349 unique words and symbols. There are 12443 tweets that have at least one word in this vocabulary. Therefore, we only discard 2.6% of tweets as pointless babble due to not having any common words that appear at least 20 times in the corpus. These pointless

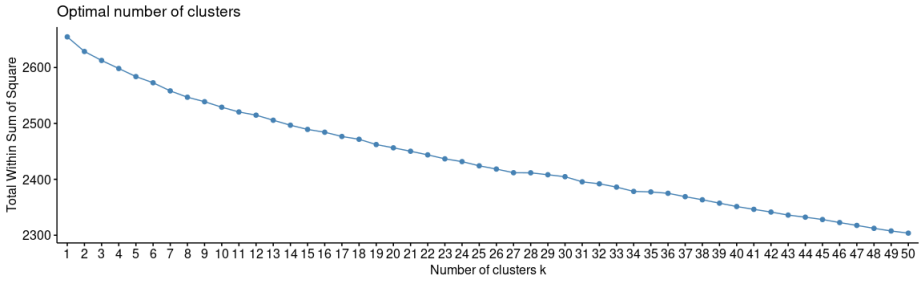

Figure 3.2: Weighted sum of squares by number of clusters for clustering on words in validation set.

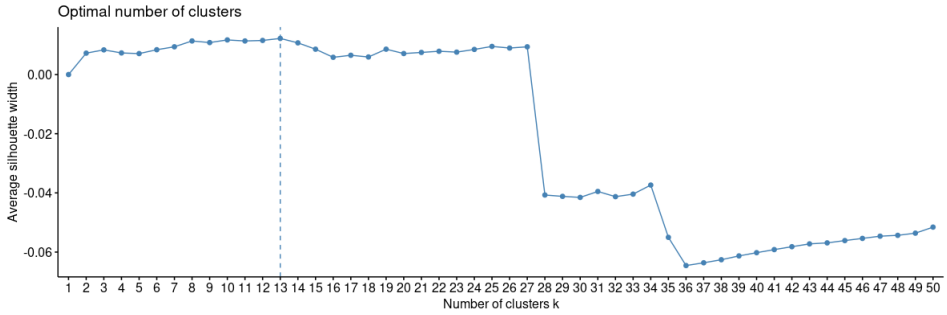

Figure 3.3: Silhouette by number of clusters for clustering on words in validation set.

babble tweets mainly consist of urls. We calculate the distance between words using conditional probability. Note that we also performed the analysis using Jaccard index, but the conditional probability appear to perform slightly better. We omit results using Jaccard index as the distance between words.

First we perform k-medoids clustering on the distance between words found in the corpus of tweets. To determine the optimal number of clusters k, we consider the elbow method using the within sum of squares and silhouette analysis. The plot of weighted sum of squares can be seen in figure 3.2. As mentioned earlier, the optimal number of cluster can be ambiguous using this method; this is one such example. We also consider a silhouette analysis, as can be seen in figure 3.3. Using this method, 13 is the optimal number of clusters. Thus, we choose 13 clusters.

Appendix D gives tables showing the words found in each cluster as given by our algorithm. The size of the clusters varies, but nearly every topic fairly clearly refers to only one user. For example, cluster 9 (‘earthquak’, ‘accord’, ‘feel’, ‘felt’, ‘magnitud’, ‘report’, ‘san’) refers to earthquakes and cluster 8 (‘vegan’, ‘ad’, ‘add’, ‘almond’, ‘amaz’, ‘appl’, ‘asparagu’, ‘avocado’, ‘bake’, ‘banana’, ‘bar’, ‘base’, ‘basil’, ‘bbq’,...) clearly refers to vegan cooking. Note that the first word in each cluster (also bold in the table) is the word chosen as the centroid for each cluster, and in most cases this word accurately describes the general theme of that particular cluster, e.g. ‘storm’, ‘health’, ‘ppfa’ (note that ppfa is the Twitter handle of the Planned Parenthood account), ‘goblu’, ‘earthquak’, ‘rain’, ‘tropic’, and ‘snow’.

We compare these results to using LDA, again using 13 topics. In appendix D we give the top 30 words with the highest frequency in each latent topic as determined by LDA. Most of these topics generally refer

	PPFA	UMichFootball	breakingweather	vegancook1010
1	265	2825	128	48
2	12	7	4	3047
3	2781	53	61	23
4	91	65	2998	35

Table 3.1: Users versus sorted tweet topic as given by our topic modeling algorithm: 4 clusters.

	PPFA	UMichFootball	breakingweather	vegancook1010
1	57	2366	54	61
2	13	88	3026	16
3	3064	388	106	70
4	15	108	5	3006

Table 3.2: Users versus sorted tweet topic as given by LDA: 4 clusters.

to only one topic, such as topic 2 referring to rainfall. According to the results on words, both methods appear to do fairly well at finding latent topics based on words.

Next we perform clustering on the tweets themselves. Since there are four users specifically chosen because they tweet about different topics, we first perform clustering using four topics. These results can be seen in table 3.1 for our algorithm, and table 3.2 for LDA. For LDA with k topics, each tweet has an associated k-dimensional probability distribution across topics, with an entry for each category. To make the results comparable with our method, we assign each tweet to a single category, essentially fitting the LDA to a unigram model. We assign each tweet to the topic containing the maximum entry for the k-dimension probability distribution across the k topics. Both algorithms seem to work fairly well at determining which tweets belong to which users.

We now choose there to be 13 latent topics, as we did with the words. Table 3.3 gives the users versus topic each tweet was sorted into. Our algorithm appears to work fairly well: most latent topics consist of tweets from mostly one single user. Furthermore, each user has one cluster that contains most of their tweets. Some topics appear to do particularly well, containing only tweets from a single user, such as cluster 2 having 222 of its 223 tweets from 'vegancook101'. Taking a closer look at one of the clusters that appeared to do the worst, nearly every tweet in cluster 5, a mix of 'PPFA' and 'vegancook101' tweets, contained the word 'black'; for PPFA tweets that was in reference to race, and with vegancook101 tweets this was in reference black beans, black olives, etc. Note that since clustering is done independently on word and tweet distances using our algorithm, a specific cluster in words does not necessarily have a corresponding cluster in tweets. With LDA, on the other hand, a given topic number for frequent words is associated with the same topic number in table 3.4.

We compare this tweet clustering result with LDA. Table 3.4 gives these results aggregated by user. Comparing these results to the results using our algorithm in table 3.3, our algorithm performs much better. This demonstrates how standard LDA can fail for short texts such as tweets. Figure 3.4 gives the log-likelihood $\pm$ one standard deviation for various numbers of clusters. According to this measure, LDA has

	PPFA	UMichFootball	breakingweather	vegancook101
1	237	2767	127	42
2	1	0	0	222
3	2348	44	42	17
4	1	12	0	0
5	85	2	2	104
6	335	39	29	15
7	1	12	0	0
8	80	63	2811	42
9	49	3	173	2
10	0	1	0	77
11	12	6	4	2525
12	0	1	0	105
13	0	0	3	2

Table 3.3: Users versus sorted tweet topic as given by our topic modeling algorithm.

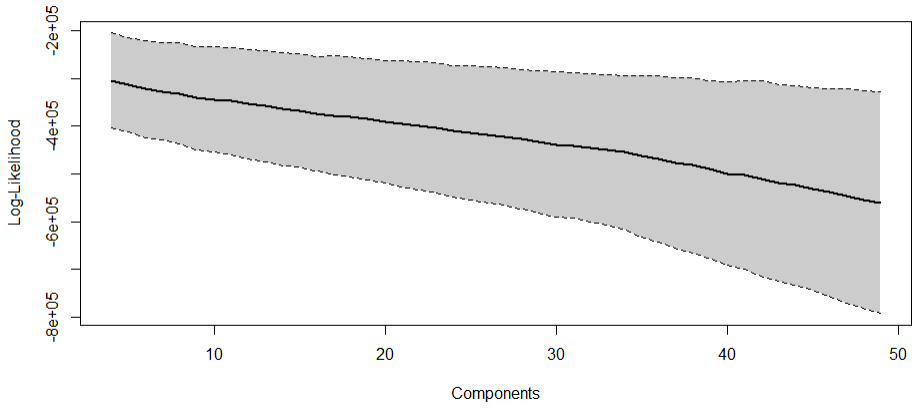

Figure 3.4: Log-likelihood of LDA model by number of topics.

the best fit with 4 topics and declines as the number of topics increases.

As another metric to compare how LDA performs on the corpus, we consider the mean of the posterior probability distribution for each tweet that belongs to each user. Results using this metric can be seen in appendix E. Our method still outperforms LDA when using this metric.

To compare these methods to another topic modeling method designed explicitly for this format of text documents (i.e. multiple tweets by a number of users), we apply Twitter-LDA to our validation set of users (Zhao et al. 2011)[1]. Twitter-LDA follows a framework similar to standard LDA, but assumes the unigram model. This model assumes some number of underlying topics shared between all of the users. Note that it does not assume that any single topic is present in only one user, as is roughly the assumption for the setup of this validation set. This model requires several tweets from a single user; if tweets are not collected in this manner, the model may not perform as well. The results from Twitter-LDA are in table 3.5.

[1]Java code to implement Twitter-LDA is available at `https://github.com/minghui/Twitter-LDA`

	PPFA	UMichFootball	breakingweather	vegancook101
1	83	393	64	110
2	106	83	903	84
3	879	233	118	29
4	999	79	125	43
5	89	149	114	32
6	116	189	209	52
7	61	1157	323	13
8	37	58	185	489
9	92	109	104	78
10	403	166	14	81
11	60	163	959	29
12	39	64	30	2034
13	185	107	43	79

Table 3.4: Users versus sorted topic as given by LDA.

	PPFA	UMichFootball	breakingweather	vegancook101
1	812	51	72	1
2	2	26	768	0
3	2	6	1129	0
4	4	2	2	1524
5	0	0	0	1650
6	12	6	715	0
7	7	15	396	1
8	499	138	1	5
9	859	13	5	0
10	934	11	40	0
11	11	1242	12	0
12	40	420	55	19
13	0	1304	5	0

Table 3.5: Users versus sorted topic as given by Twitter-LDA.

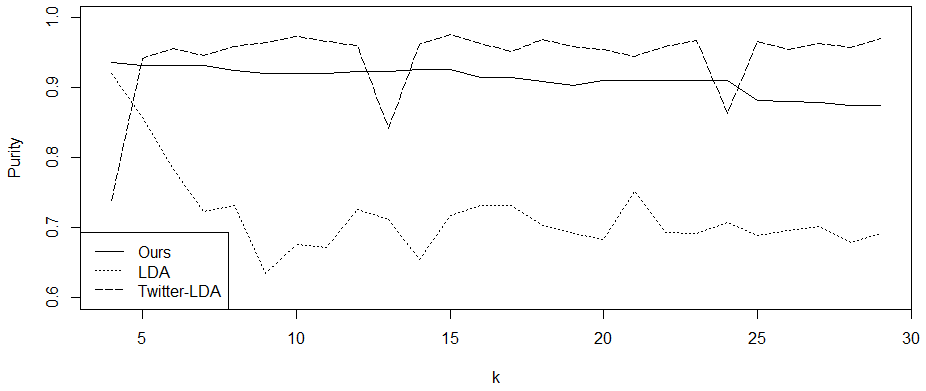

Figure 3.5: Purity for our method compared to LDA for various number of clusters.

We compare the performance of these three methods using a measure of purity. A 'perfect' fit of the clustering would have each cluster consist of tweets from just one user. We count the maximum number of tweets belonging to a single user for each cluster, added up for each cluster, and divide by the total number of tweets. This gives a purity measure of 1 if each cluster consists of tweets from a single user. Figure 3.5 give this for various number of clusters. The purity of our method stays relatively high throughout, outperforming LDA for every cluster size. Twitter-LDA, on the other hand, often performs better than our method on the validation set. However, Twitter-LDA has a serious advantage in that it is given an accurate starting point; it is given the user that each tweet belongs to (i.e. the larger topic that the tweet belongs to) and finds the subtopics. User is not taken into account using either our algorithm or LDA. This is especially an advantage in this specific case, as user and topic are synonymous.

3.3.2 Application to "Jobs" Tweets

Relationships found between sentiment of tweets containing a given word and public opinion surveys have sparked optimism in the potential for tracking public opinion with social media data. For example, O'Connor et al. (2010) found relationships between sentiment of "jobs" tweets and consumer confidence, sentiment of "Obama" tweets and presidential approval, and frequency of "Obama" tweets and Obama's standings in 2008 election polls. Furthermore, results in Cody et al. (2016) and Daas and Puts (2014) support the results in O'Connor et al. (2010), also finding seemingly strong relationships between tweets containing a given word and survey responses. This is discussed in greater detail in chapter 2.

However, a closer examination of these results concluded that the seemingly strong relationships between tweets and survey responses were likely spurious. In chapter 2 we performed a sensitivity analysis on the relationship between "jobs" tweets and consumer confidence as originally presented in O'Connor et al. (2010), finding that seemingly small changes in how sentiment is calculated can result in large changes in the resulting

correlation.

While the criteria for tweets being in the corpus (i.e. a tweet containing the word "jobs") was specifically chosen to reflect feeling towards the economy, not every tweet in the corpus was relevant to the economy. For example, many tweets were about Steve Jobs, the co-founder of Apple. Even among tweets that were relevant to actual economic jobs, the content of the tweets varied greatly. Some of these tweets were statements of a personal job search (e.g. "applying for jobs"), while others dealt with the economy as a whole (e.g. "company cuts many jobs"). We created an algorithm by hand to sort tweets into one of five categories: news/politics, personal, advertisements, other, and junk. See section 2.2.1 and appendix C for further details on this hand-created classification algorithm. However, despite evidence that the hand-classification algorithm worked fairly well as intended, it did not help to restore or strengthen the relationship between sentiment of "jobs" tweets and consumer confidence.

We apply our topic modeling algorithm to a sample from the same set of "jobs" tweets from section 2.2.1, limited to tweets from 2008-2009. For each day in the time frame we sample 300 tweets, then take a random sample of size 20000 from those tweets. We sample in this way to get roughly the same number of tweets from each day in our final sample since there was a higher frequency of tweets as time went on since Twitter was gaining in popularity throughout the time period of 2008 and 2009. We consider tweets that contain at least one word that was mentioned at least 10 times in the entire corpus. This removed only 0.3% of the tweets as pointless babble. We calculate distance between words using conditional probability.

Figure 3.6 gives the silhouette measure for our method for clustering tweets. According to silhouette, four is the optimal number of clusters. However, from inspection we know that there are more than four topics present in the corpus; in section 2.2 we created an algorithm by hand to classify these "jobs" tweets into five topics. Instead, we choose there to be 10 latent topics. This is twice the number of clusters that was chosen in chapter 2.2.1. Previously, we relied on humans to determine the categories of "jobs" tweets and create the classification algorithm based on a relatively small set of words. We chose broad categories when creating the topics by hand, some of which contained several sub-topics. For example, *junk* tweets contained tweets about Steve Jobs and the TV show "Dirty Jobs"; tweets in the *other* category were not necessarily on the same topic, but just contained links to articles online. With an automated algorithm, we can more easily allow for more, and hopefully better, categorization of "jobs" tweets.

Table 3.6 compares the category as given by the topic modeling algorithm to the classification as given by the hand-created algorithm. Keep in mind that neither the hand-classifications nor the topic modeling clusters should be considered ground truth. Furthermore, with lack of ground truth, we do not know which algorithm is in any sense 'better'. From table 3.6 we can see that many clusters are not clearly dominated by a single hand-created classification. However, most of the *advertisement* tweets were clustered together, similarly with *irrelevant* tweets. There is some agreement between to two classification methods, but not an incredibly large overlap. The hand-created algorithm does have an advantage in that it does use the user name to classify tweets *news/politics* and *advertisements*, whereas the automated topic modeling algorithm

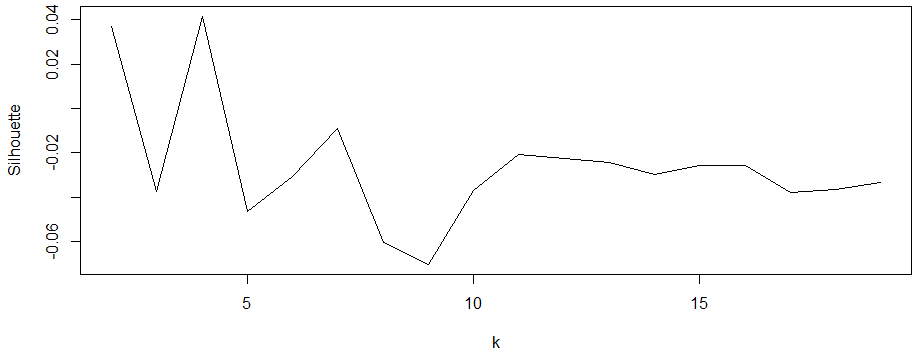

Figure 3.6: Silhouette by number of clusters for "jobs" tweets.

	Advertisements	Irrelevant	News/Politics	Other	Personal
1	26	9	33	74	386
2	0	1	0	2	8
3	25	35	23	58	286
4	272	2520	314	1145	1896
5	72	11	29	121	143
6	1256	252	526	4074	2467
7	85	7	367	468	265
8	0	1	0	1	11
9	138	3	10	90	5
10	144	40	191	1536	515

Table 3.6: Number of tweets classified in each cluster as given by the topic modeling algorithm compared to classification as given by the hand-created classification algorithm.

by design uses just the words within the tweets. This is especially helpful when searching for certain topics, such as *advertisements*, which very often contain the word "job" or "career" in the user name and do not have many common words between then. Similarly for *news/politics* tweets containing the word "news" in the user name.

For each topic, we give general themes of the topic by looking at individual tweets from that topic, calculate the proportion of tweets that are sorted into that topic, average sentiment as calculated by Vader (Hutto and Gilbert 2014), and correlation between sentiment of tweets from the cluster and consumer confidence (both with 30-day smoothing and no lag). These results are in Table 3.7. Previous results in section 2.2 suggest that the relationship between sentiment of "jobs" tweets and consumer confidence is likely spurious, so we do not expect to observe any strong relationships; a seeming strong observed relationship could likely be spurious.

The cluster having the highest correlation with consumer confidence, cluster 10, contained many tweets about local news events, such as local companies cutting jobs. Many of these contained links to articles online, which is why many of these tweets were sorted into the *other* category. Since many economic news stories

Cluster	Themes	Proportion	Mean Sent.	Correlation
1	Personal: applying for jobs	0.0264	0.0936	-0.1819
2	Personal	0.0006	-0.1210	0.1035
3	Personal: working two jobs	0.0214	0.0736	0.0984
4	Steve Jobs, personal	0.3074	0.1049	0.1343
5	Personal, Advertisements	0.0188	0.1128	-0.1720
6	Mixed	0.4288	0.1060	0.2356
7	News: adding jobs	0.0596	0.2035	0.0524
8	Personal	0.0006	0.1407	0.2522
9	Advertisements	0.0123	0.0336	0.3127
10	News: cutting jobs	0.1213	-0.1491	0.4628

Table 3.7: Information on each cluster of "jobs" tweets: cluster number, proportion of "jobs" tweets belonging to each cluster, average sentiment of a cluster, and correlation between sentiment of tweets from each cluster and consumer confidence (with 30-day smoothing) for 2008-2009.

from 2008-2009 were about cutting jobs rather than adding jobs, it makes sense that this cluster also has the lowest mean sentiment. Looking at the content cluster 9, nearly all of these tweets were advertisements. This cluster had a mean sentiment close to 0, which makes sense given that job advertisements do not contain much sentiment. However, this cluster also had the second highest correlation with consumer confidence. This correlation is presumably spurious since advertisements do not give any opinion or statement about the economy. Many tweets in cluster 7 were about job gains, either at the individual or community level. With this theme in mind for cluster 7, it makes sense why cluster 7 has the highest mean sentiment. However, it curiously has the weakest correlation with consumer confidence. The content for cluster 4 appears to be very mixed, from talking about Steve Jobs to individuals commenting on their job status to new about job losses.

We did try various parameter values, such different number of clusters and different number of minimum word thresholds, but results were relatively consistent throughout; results did not appear to substantially weaker or stronger with neither consumer confidence nor hand-classification agreement when altering various parameters.

Overall, the topic modeling of the "jobs" tweets does not substantially improve any previously found relationship. This is not a surprising result given the conclusions in chapter 2. Furthermore, if we had initially chosen the new clusters as our categories instead of the previous hand-created classification system, we likely would have reached the same conclusions.

There are advantages and disadvantages to both the topic modeling and hand-classification methods. As humans, we can determine which tweets may fall under the same category even when their words are entirely different; understanding the larger societal context is important in accomplishing this task. The human knowledge, along with using user name, is a huge advantage for the hand-classification algorithm. However, with human-created keywords, it is difficult to create nuanced categories with any decent level of accuracy, and a human might not pick up on some of these nuanced categories to begin with. The advantages and disadvantages are reversed with the automated topic modeling: it can pick up on many more topics than

the hand-classification algorithm, but without a larger context it can be prone to error. While we do not explore the idea in this dissertation, sorting tweets into different categories may be able to be improved by combining aspects of the hand-created and automated classifications. For example, the topic modeling could be performed on tweets the hand-created algorithm classified as *personal* to reveal themes within personal tweets, similarly with *news/politics* tweets to extract themes of potentially losing jobs, creating jobs, and political aspects of the economy. This type of semi-supervised hierarchical topic modeling may be very applicable in future analyses.

3.3.3 Application to Politically Active Users

In chapter 2.4, we concluded that while a political signal was not strong in tweets containing the word "Trump" in terms of capturing presidential approval, a convincing political signal was present when following tweets from politically active users in terms of the frequency and sentiment of their tweets around the 2016 election. In finding this signal, we considered *all* original tweets from the set of Democratic and Republican users. However, these users do not tweet exclusively about politics; they tweet about sports, entertainment, and random thoughts and musings. The non-political tweets are more than just pure noise. If we can properly identify tweets as being political or not, we can hopefully strength the political signal from the politically active users. We found it difficult to even manually determine whether tweets were political or not based purely on content. For example, say a user is live-tweeting a presidential debate. If we knew that context, we would easily classify all of the tweets from that live tweeting session as political. However, if we see one of those tweets in isolation and without the larger context (e.g. "What a ridiculous answer!"), we would not necessarily hand classify that tweet as political. This was the justification for using all tweets from the politically active users in section 2.4. We see if automated topic modeling helps to achieve the goal of classifying tweets as political or not. We apply our topic modeling algorithm on a random sample of 100 Democratic users and 100 Republican users (see chapter 2.4 for details of how these sets of users were created).

The politically active users have thousands of tweets each; running our topics modeling algorithms on the entire corpus of tweets from these users would take a large amount of memory and a long time to run. Instead, we run the topic modeling algorithm on each user individually. We choose there to be 10 latent topics for each user. We then need a method to extract which clusters are political. To accomplish this goal, we take advantage of the large amount of information available online. The use of outside data is not itself a novel idea; for example, Jin et al. (2011) and Bicalho et al. (2017) also make use of outside data from long texts in topic modeling of short texts. However, we make use of this data in a different way: instead of using outside data to help in clustering, we use it to help determine which clusters are of interest to our study.

Using larger texts with known target topic (politics in this case), we insert artificial political 'tweets' into the set of tweets for each user. These are tweets that are specifically created to be political. There were many political events that occurred in the given time frame, and we do not want our personal biases of

what constitutes a meaningful political event to affect the outcome. Therefore, we do not want to create the individual tweets by hand ourselves. We download the Wikipedia pages for the 2016 presidential election (`https://en.wikipedia.org/wiki/2016_United_States_presidential_election`) and the Trump presidency (`https://en.wikipedia.org/wiki/Presidency_of_Donald_Trump`). These two Wikipedia pages account for an extensive review of nearly all political events in the given time frame. We generate artificial political tweets using Markov chains, with the transition matrix created from each of the Wikipedia pages. Resulting artificial political 'tweets' are fairly nonsensical, but nonetheless are clearly political. As examples:

- "National Committee (DCCC) and controversial nature set the Russian businessman from New York, North Carolina, 43 to be "a social discord in a Democrat and also took place between groups are paid to fix problems in several electors in 1996, while Pennsylvania (three swing states are indirect election, held on September"
- "Republican National Convention was released by the election, and statisticians, including Florida. Rubio won the Clinton said Vladimir Putin had secured Trumpś smaller victories in the campaign. The following the GOP electors) Wisconsin, and "violated U.S. democratic means The United States Code, nor military experience, denounced Trump won 2,204 pledged to"
- "Trump to ask her illness. Video footage of the 2016 Republican candidate Lawrence Lessig withdrew due to receive more than among blacks and Pennsylvania election interference in 2012, although the project Nicknames used to low-income occupations or non-employment as his lead in 1948 presidential candidate ever," exceeding George H. W. Bush"
- "Rioters also served only three recount in New York primary process, Clinton won in recent Republican National Convention, Pence head the race, he notified Congress officially sanctioned televised debates based on September and rented a private email server, in Houston, Texas. Constitution provides that the Electoral College to same-sex marriage, and"

Since the topic modeling algorithm takes a bag-of-words approach, the words being political is what matters, not their meaning as a coherent sentence. We do not assume the political tweets are generated from a Wikipedia pseudo-document (like Zuo et al. (2016) do) since Wikipedia states what events happened, whereas tweets also include reactions to those events and generally in less formal language. Political tweets and artificial political tweets are related in that they share some similar words, but not exactly generated from the same pseudo-document. We insert 20 artificial political tweets into each user's set of tweets.

After running the topic modeling on the tweets, it remains to extract which resulting clusters are political. We consider a cluster to be political if it contains at least three artificial tweets. Using this definition of a political cluster, most users only had 1 or 2 political clusters out of 10 total clusters. Out of the 200 users in our study, 63.5% have one political cluster, 32% have two political clusters, and 4.5% have 3 political clusters; none had more than three political clusters. The fact that the artificial tweets end up in the same

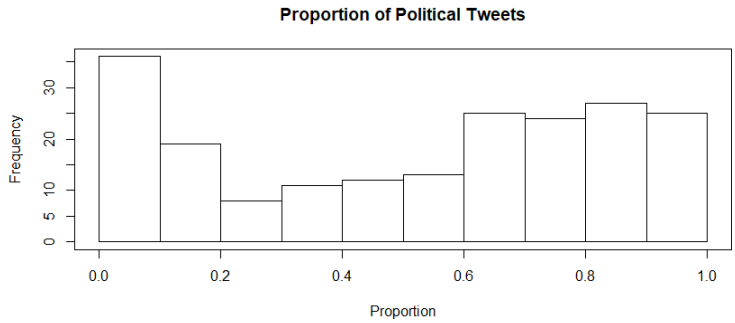

Figure 3.7: Proportion of tweets that are labeled political for each user.

small number of clusters confirms that our clustering method is working as intended, with political tweets being clustered together. The artificial tweets are used for clustering purposes only; they are removed from the analysis after we have determined which clusters are political. We consider a tweet to be political if it belongs to a cluster that has at least three of the artificial political tweets. Figure 3.7 gives the distribution of the proportion of tweets that we consider political for each user. From this distribution we see that some users tweets almost exclusively about politics, and others hardly at all. This is not an unexpected result, as we did not have any requirements about what proportion of tweets had to contain a political words for a user to be considered politically active, and many users have interests outside of politics. This distribution looks similar for Democrats and Republicans.

Similar to chapter 2.4, we compare the frequency and sentiment of political versus non-political tweets around the 2016 presidential election.

First we consider frequency of political and nonpolitical tweets. Figure 3.8 gives the frequency of political and nonpolitical tweets. Frequency of political tweets spike around political events, such as the election and debates. However, we see similar spikes in nonpolitical tweets. We also spikes in August for the nonpolitical tweets, when there were no notable political events happening. Political tweets see a spike in March 2016. This was right around Super Tuesday (when multiple states voted in presidential primaries for both parties), and was not seen in the nonpolitical tweets. While the political and nonpolitical look to be capturing some signals that the other is not, overall the signal does not appear to be much stronger in the political tweets versus nonpolitical.

Next we compare sentiment of political and nonpolitical tweets for Democrats and Republicans. To ensure that every user is weighted equally regardless of how often they tweet, we create a political and nonpolitical daily sentiment score for each user. The weighted sentiment score for a given day weights the sentiment of a user's tweets from the past 30 days, with more recent tweets being more heavily weighted. For similar reasons as in chapter 2.4, we are interested in the difference in sentiment between Democrats and Republicans.

Figure 3.9 gives the difference in sentiment (Democrats-Republicans) for political and nonpolitical tweets,

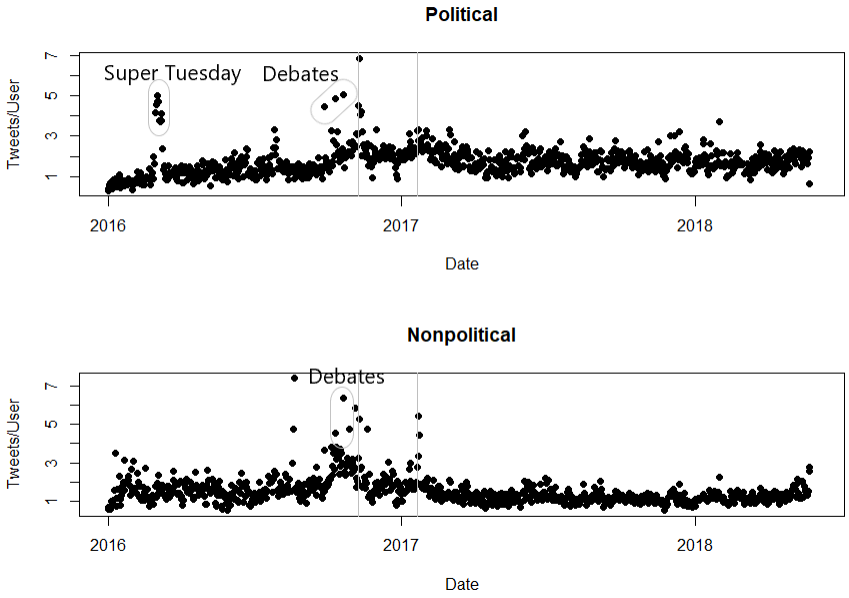

Figure 3.8: Frequency of political and nonpolitical tweets.

with a smoothed trend line for each. We found in chapter 2.4 that Democrats were generally happier before the election, and Republicans generally happy after the election. From Figure 3.9, we see that this is due to what our algorithm classified as nonpolitical tweets. This difference in sentiment peaks in late summer 2016, with Democrats being happier than Republicans. We also do see a decrease in difference in sentiment of political tweets in the months leading up to the election. We do see a decrease in difference in political sentiment immediately following the election, which is not see in the nonpolitical tweets. Democrats were presumably not happy politically during that time, which indicates that we may be capturing a true political signal, however small. The political tweets were more positive for Republicans in for nearly the entire time period we consider.

3.4 Discussion

In this chapter we developed a new method for topic modeling of short texts that is based on clustering on distances between tweets. This method assigns a latent topic to each tweet or word found in the corpus based on the distances calculated using corpus-level word co-occurrence. This method is flexible in that it allows multiple methods for calculating the distance between words. More importantly, unlike many other topic modeling algorithms for tweets, this method has the advantage of being applicable to any corpus of tweets or short texts and does not require any auxiliary information for each text.

Applying this method to a validation set of users confirmed that this algorithm does indeed work as intended. It also validated the manually-chosen categories of "jobs" tweets used in Conrad et al. (2019).

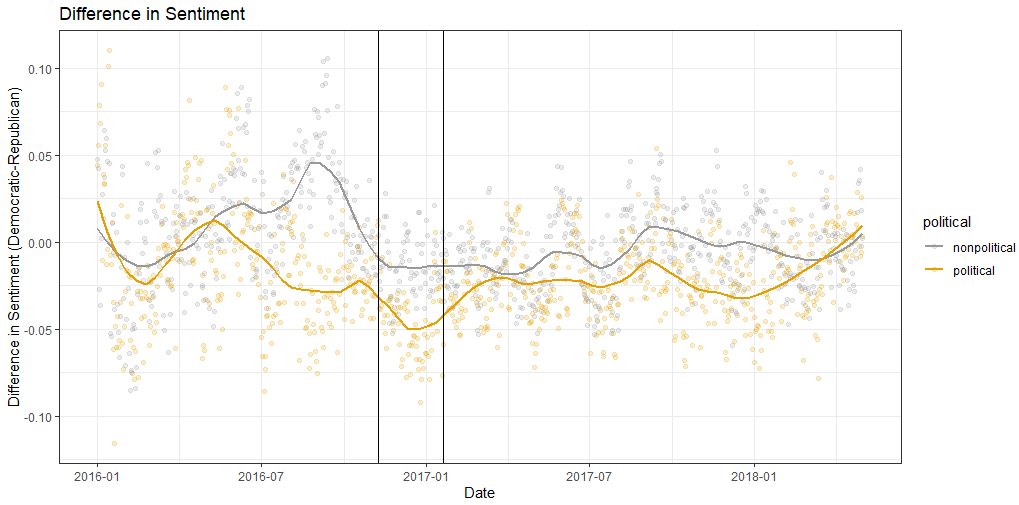

Figure 3.9: Difference in sentiment (Democrat-Republican) for political and nonpolitical tweets.

However, it did not greatly improve the strength of the observed signal when tracking politically active users over time. Previous results have concluded that signals in social media data are not simple and straightforward, and the topic modeling of these tweets adds to the literature in that it is not necessarily irrelevant 'noisy' tweets included in the corpus that are at fault. We do not conclude that there is *no* signal in Twitter data, but rather it is difficult to extract.

This algorithm may also be applied to document summarization, when a larger body of text is to be summarized by a few sentences. For example, say we want to automatically summarize a research paper with a few sentences. There are likely several topics present in the paper, such as background/literature review, methods used, results, and discussion. We can treat the entire paper as a corpus, and each sentence as a 'tweet'. After apply our methods outlined in this chapter to the corpus of sentences in the paper, we have k resulting topics. We can choose the sentences chosen as the centers of the cluster to summarize the entire paper. If we want to summarize the paper in only one sentence, we can choose the sentence that has the smallest average distance between itself and all other sentences in the corpus.

While our algorithm worked fairly well on the validation set of tweets, especially in comparison with LDA, its performance was lacking for the politically active users. This suggests that the differences between political and nonpolitical tweets in not purely the words used in tweets. It is important to remember that tweets do not happen in a vacuum; events happening outside of Twitter are very relevant to the meaning, intention, and interpretation of tweets, and individual tweets must be understood in that larger context. For example, we know that events are of particular importance in term of frequency of tweets relating to some topic. While our method was specifically designed to not incorporate auxiliary information so it could be applied to all types of corpora, we do acknowledge that this auxiliary information can be valuable and it may be worth considering in future work of social media data.

Chapter 4

Unbiased Survey Estimation with Population Auxiliary Variables

Another method for improving survey estimates is by taking advantage of available population data and advances in machine learning predictive modeling algorithms. Using these methods can decrease standard error of estimates, therefore reducing the sample size required for a given level of uncertainty.

In many applications, population auxiliary variables can be utilized to increase the precision and accuracy of survey estimates. We develop an imputation-based estimation method that produces an unbiased estimate of the mean response of a finite population from a simple random sample when population auxiliary data are available. Our method allows for any prediction function or machine learning algorithm to be used to predict the response for out-of-sample observations, and is therefore able to accommodate a high dimensional setting and many covariate types. Exact unbiasedness is guaranteed by estimating the bias of the prediction function using subsamples of the original simple random sample. Importantly, the unbiasedness property does not depend on the accuracy of the imputation method. We apply this estimation method to simulated data, college tuition data, and the American Community Survey, showing a decrease in variance compared to the sample mean and increased accuracy compared to standard adjustment methods.

4.1 Introduction

Traditional probability-based samples are the "gold-standard" of survey sampling due to the sample being representative in expectation of the population from which it is drawn. Many methods for making inferences about a population from a sample rely on this property of probability-based samples. However, due to randomness in sampling, a single sample may not be exactly representative of the population. If population-level auxiliary data are known (i.e. covariates known for all individuals in the population), however, adjustments can be made to account for the differences between the population- and sample-level covariates to improve

inference on a population response. There currently exist many methods for integrating population data and survey data, but these methods are often biased (e.g. Breidt and Opsomer (2017)), sensitive to misspecification (Hansen, Madow, and Tepping 1983), or fail to work with high-dimensional data.

Population auxiliary data are available in a wide variety of applications and are often provided by government agencies. For example, the Census Bureau provides demographic characteristics from the person level to the federal level, the National Center for Education Statistics provides data for educational institutions, and the Department of Health and Human Services provides data for all substance abuse treatment centers. If population-level data are not directly available, a synthetic population can be generated from a well-designed, probability-based reference survey. In one recent example, Rafei, Flannagan, and Elliott (2020) suggest potentially generating a synthetic population through finite population bootstrap.

Population auxiliary information has long been used to improve population point estimates from a sample. Ratio estimators were an early method of incorporating population information. Ratio estimators historically only assumed knowledge of a single population mean covariate, making calculations easier before computers were widely available. There are many ratio estimators, typically functions of the population covariate mean and the ratios of the sample means of the covariate and response. Standard ratio estimators are biased; an unbiased ratio estimator is derived in Hartley and Ross (1954). A class of unbiased ratio estimators are derived in Mickey (1959) and Williams (1961) (Williams (1962) shows these are equivalent). Ratio estimators can be improved upon by using more than a single covariate and more information than simply the population mean of the covariate, see Subramani (2013) for an extensive list of ratio estimators.

Many modern methods for incorporating population auxiliary information with sample data can be considered predictive inference methods, where response values are imputed for out-of-sample observations using some prediction function trained on in-sample observations. We describe some of these methods below. See Valliant, Dorfman, and Royall (2000) and Buelens, Burger, and Brakel (2015) for further details.

The first approach we consider is the model-assisted approach, where individual out-of-sample responses are imputed using predictive modeling algorithms trained on the sample. We refer to these as standard adjustment methods. A variety of prediction algorithms can be used, such as GLMs, random forest, or k-Nearest Neighbors (Hastie, Tibshirani, and Friedman 2009). As an example, when performing regression adjustment, all of the observations in the sample are used to fit a regression function that predicts the response variable given the covariates. Then using that regression function, response values are predicted for each observation not in the sample. This is also simply refered to as the regression estimator (Mickey 1959; Williams 1961; Särndal, Swensson, and Wretman 2003). A number of these estimators have been studied with different prediction functions. For example, McConville et al. (2017) discuss finite population lasso imputatation under sparsity assumptions, Breidt and Opsomer (2000) discuss model-assisted estimation with local polynomail regression, and Dagdoug, Goga, and Haziza (2020) use random forest for finite population inference. These adjustment methods can also be applied in a Bayesian setting, where values are imputed for non-sampled units based on the posterior predictive distribution (Dong, Elliott, and Raghunathan 2014).

Imputation methods can also be used when combining probability and non-probability surveys, with prediction models for a response being trained on the non-probability sample and applied to the probability sample (Kim and Rao 2011; Chipperfield, Chessman, and Lim 2012). One caution with these standard adjustment methods is overfitting a prediction model to the sample data, which can lead to biased response estimates for the out-of-sample observations (Hawkins 2004).

Another method for integrating population data and sample data is weighting, where responses in the sample are weighted with respect to how prevalent similar subjects are in the population, or probability of inclusion in the sample. Post-stratification and the Horvitz-Thompson estimator are two examples of weighting a sample to resemble the population of interest (Horvitz and Thompson 1952). Both of these estimators give an unbiased estimate of the population mean response. Post-stratification can also be considered a predictive inference method in which the response for each out-of-sample observation is imputed as the mean of the sample observations in the same strata (Buelens, Burger, and Brakel 2015). Breidt and Opsomer (2017) discuss the model-assisted estimator, which uses the Horvitz-Thompson estimator with imputation to estimate a mean population response.

Buelens, Burger, and Brakel (2015) and Liu, Chen, and Gelman (2019) compare through simulation the performance of multiple predicitive inference methods. Buelens, Burger, and Brakel (2015) simulate non-probability samples from the Online Kilometer Registration in the Netherlands, using as comparative predictive inference methods the sample mean, stratification, GLM, k-nearest neighbors, artificial neural nets, regression tress, and support vector machines. Liu, Chen, and Gelman (2019) consider the sample mean, post-stratification, raking weights, Bayesian Additive Regression Trees (BART) prediction, and Soft Bayesian Additive Regression Tree (SBART) prediction, finding that the SBART prediction model often performs the best on the simulated data, having the lowest bias and lowest RMSE. BART closely followed. They then apply the methods to data on substance abuse treatment centers to predict the total number of patients receiving substance abuse treatment in the United States.

There are several drawbacks to predictive inference methods: estimated parameters in prediction models are subject to error, model assumptions can be violated, and prediction models can be biased. These errors carry through to the predicted responses and ultimately the final estimate for the population mean response (Buelens, Burger, and Brakel 2015). And while post-stratification works well when there are a manageable number of strata, when the number of variables grows and there are relatively few observations in each strata, population response estimates have high variance (Little 1993).

In this paper we consider a finite population model with fixed response values in which the population covariates are known for each member of the population and the response is known for only a simple random sample of the population. We derive an unbiased imputation-based estimation method for estimating the population mean response. We obtain this estimation method by estimating the bias of the chosen predicting function using subsamples of the original sample, specifically in a leave-one-out fashion. This estimation method has the flexibility of a user-chosen prediction function. With this flexibility we contribute to the

literature on bringing machine learning techniques into survey adjustment, accommodating any covariate type (continuous or categorical) and a high-dimensional setting where the number of covariates is greater than the sample size. Our estimation method potentially opens the door to make use of nontraditional data, such as social media data, for which formal modeling methods with realistic assumptions may not be fully developed. Because the prediction function is arbitrary, we can achieve unbiased estimates regardless of how well a particular machine learning algorithm fits the data.

This paper is organized as follows. In section 4.2 we introduce our estimator, show it is unbiased, and provide a variance estimate. In section 4.3 we apply our estimator to simulated data, college data, and the American Community Survey using population tax data. Section 4.4 concludes.

4.2 Method

In this section we introduce the estimation method, which provides an unbiased estimate of a population mean response from a simple random sample when we have full knowledge of population covariates. We show that this estimator is unbiased and provide an estimate of the variance.

4.2.1 Estimator

Consider a finite population in which there are N members, indexed by $i = 1, 2, \ldots, N$, where N is known. For each member of the population we observe a fixed p-dimensional covariate vector $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$. Each member of the population has a fixed scalar response value y_i. Our primary parameter of interest is the population mean response:

$$\mu = \frac{1}{N} \sum_{i=1}^{N} y_i$$

A simple random sample (without replacement) $\mathcal{S}$ of size n is taken from the population. Let s_i be the indicator variable for whether observation i is in $\mathcal{S}$. That is, $s_i = \mathbb{1}(i \in \mathcal{S})$. s_i is independent of $\boldsymbol{x}_i$ and y_i, so $P(s_i = 1) = \frac{n}{N}$ for all i. For each member in $\mathcal{S}$ we observe the response value y_i and the covariates $\boldsymbol{x}_i$. In general, let bold letters denote a vector or matrix and let nonbold letters denote a scalar.

In this setting, the sample mean $\frac{1}{n} \sum_{i \in \mathcal{S}} y_i$ gives an unbiased estimate of the population mean μ, but by incorporating the additional information in the covariates we may be able to decrease the standard error of the estimate. A common method of incorporating the covariates is through predictive modeling: use observations in the sample to train a function that predicts the response given the covariates, and apply that function to predict the response values for observations not in the sample. The estimate of the population mean μ is taken as the mean of the responses in the sample and predicted responses for observations not in the sample. However, in a finite population setting with fixed response values, imputing the response variable for individuals not in $\mathcal{S}$ can lead to biased estimates of μ. Using ideas similar to Mickey (1959), we estimate the bias using sub-samples of $\mathcal{S}$. The intuition for our estimation method is as follows. Suppose

the sample $\mathcal{S}$ is split into two sub-samples $\mathcal{S}^{(1)}$ and $\mathcal{S}^{(2)}$. Using the observations in $\mathcal{S}^{(1)}$, train some function $f(\cdot)$ to predict y from $\boldsymbol{x}$. The function $f(\cdot)$ could be biased for the observations not in $\mathcal{S}^{(1)}$. $\mathcal{S}^{(2)}$ is a random sample of the observations not in $\mathcal{S}^{(1)}$, so the bias observed in $\mathcal{S}^{(2)}$ is an unbiased estimate of the bias in $\mathcal{S}\backslash\mathcal{S}^{(1)}$. Then for all observations not in $\mathcal{S}$ we can impute the estimated y value using $f(\cdot)$ and subtract the estimated bias. We implement this idea in a leave-one-out fashion: for each observation $i \in \mathcal{S}$, we train $f(\cdot)$ on $\mathcal{S}\backslash i$, estimate the bias as the difference between y_i and the predicted value for y_i using $f(\cdot)$ trained on $\mathcal{S}\backslash i$, predict the response for the out-of-sample observations using $f(\cdot)$ trained on $\mathcal{S}\backslash i$ and subtract the bias, and take the mean of the known and estimated response values.

To define the estimator formally, we first define several variables. Let $f(\cdot; \boldsymbol{x}_{(n-1)\times p}, \boldsymbol{y}_{(n-1)\times 1})$ be a prediction function parameterized by $n-1$ $(\boldsymbol{x}, y)$ pairs that predicts y given $\boldsymbol{x}$. Let

$$f_i = f(\boldsymbol{x}_i; \boldsymbol{x}_{\mathcal{S}\backslash i}, \boldsymbol{y}_{\mathcal{S}\backslash i})$$

be defined if $i \in \mathcal{S}$. f_i is the predicted response value for observation $i \in \mathcal{S}$ when the prediction function is trained on $\mathcal{S}\backslash i$, so f_i and y_i are independent. Let

$$g_i = \frac{1}{n} \sum_{j \in \mathcal{S}} f(\boldsymbol{x}_i; \boldsymbol{x}_{\mathcal{S}\backslash j}, \boldsymbol{y}_{\mathcal{S}\backslash j})$$

be defined if $i \notin \mathcal{S}$. g_i is the average of the response value predictions for observation $i \notin \mathcal{S}$ from each of the n leave-one-out prediction models. Let

$$h_i = \begin{cases} f_i, & \text{if } i \in \mathcal{S} \\ g_i, & \text{if } i \notin \mathcal{S} \end{cases}$$

and let

$$\begin{aligned} \hat{y}_i &= y_i + (1 - s_i)(h_i - y_i) + s_i \left(\frac{N-n}{n} \right) (y_i - h_i) \\ &= \begin{cases} \frac{N}{n} y_i - \frac{N-n}{n} f_i, & \text{if } i \in \mathcal{S} \\ g_i, & \text{if } i \notin \mathcal{S} \end{cases} \end{aligned}$$

$\hat{y}_i$ can be thought of as an estimate of y_i since, as we will show below, $E(\hat{y}_i) = y_i$. Our estimator for the population response mean, which we denote $\hat{\mu}$, is the mean of the $\hat{y}$s:

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} \hat{y}_i \tag{4.1}$$

When OLS regression is used as the prediction function, our estimator is the same as the leave-one-out regression estimator given in Mickey (1959). In the case of regression, we can relax the assumption of *full* knowledge of population covariates; it is sufficient to know the population means of the covariates and the

individual covariate values for only the sample. We discuss this case in greater detail in the Appendix.

Another special case of interest is mean imputation. That is, ignoring covariate values and letting $f(\cdot; \boldsymbol{x}_{\mathcal{S}\setminus i}, \boldsymbol{y}_{\mathcal{S}\setminus i})$ be the mean of $\boldsymbol{y}_{\mathcal{S}\setminus i}$. In this case, our $\hat{\mu}$ estimator reduces to the sample mean $\frac{1}{n}\sum_{i\in\mathcal{S}} y_i$. We show this in the Appendix. Note that when no population auxiliary information is available, the sample mean is the best (least squares) estimate of the population mean response. The sample mean can be considered a predictive inference adjustment method since we are in essence predicting the response for every out-of-sample response to be the sample mean response. It is well known that the sample mean gives an unbiased estimate of the population mean response for a simple random sample. The standard error of this estimate can potentially be decreased by improving upon mean imputation and incorporating additional covariate information.

Our estimation method allows for any prediction function to be chosen for $f(\cdot)$, allowing the prediction function $f(\cdot)$ to be as simple or complex as one desires and be appropriate for the given data set. For example, one can use lasso, random forest, or neural networks in cases where the number of covariates is greater than the number of observations. The estimator is similar to the *model-assisted difference estimator* presented in Breidt and Opsomer (2017) under simple random sampling, which also allows for an arbirary prediction function. Breidt and Opsomer (2017) use the entire sample to train the prediction function, making their method asymptotically unbiased, whereas we use a leave-one-out procedure to ensure exact unbiasedness at all sample sizes and for any prediction function. This can make a notable difference. For example, under simple random sampling, the model-assisted difference estimator with regression reduces to the regression estimator, which as we show below can lead to biased estimates under slight model misspecification. The closed-form for our estimator with regression prediction can be found in the Appendix. The leave-one-out nature of our method also protects against overfitting of the prediction function to the sample data, whereas overfitting to the sample data is possible using the difference estimator, which can introduce bias and affect the estimated standard errors.

4.2.2 Unbiased

We show that our estimation method is unbiased for μ, regardless of the chosen prediction function $f(\cdot)$. First, define $e_i = \mathbb{E}(h_i)$, and note that e_i is the same regardless of whether observation i is in the sample or not: $e_i = \mathbb{E}(h_i) = \mathbb{E}(h_i \mid s_i = 1) = \mathbb{E}(h_i \mid s_i = 0)$. Further, let $\delta_i = h_i - e_i$. Then $\mathbb{E}(\delta_i) = \mathbb{E}(\delta_i \mid s_i = 1) = \mathbb{E}(\delta_i \mid s_i = 0) = 0$. Now $\hat{y}_i$ can be rewritten as

$$\hat{y}_i = y_i + (1 - s_i)(e_i + \delta_i - y_i) + s_i \left(\frac{N-n}{n}\right)(y_i - e_i - \delta_i)$$

In the above formula, only the variables s_i and δ_i are random; the remaining variables are fixed. Taking the expectation of this expression gives

$$\begin{aligned}
\mathbb{E}(\hat{y}_i) &= \mathbb{E}\left[y_i + (1-s_i)(e_i+\delta_i-y_i) + s_i\left(\frac{N-n}{n}\right)(y_i-e_i-\delta_i)\right] \\
&= \mathbb{E}\left(e_i+\delta_i-\frac{N}{n}s_ie_i-\frac{N}{n}s_i\delta_i+\frac{N}{n}s_iy_i\right) \\
&= e_i-\frac{N}{n}e_i\mathbb{E}(s_i)-\frac{N}{n}\mathbb{E}(s_i\delta_i)+\frac{N}{n}y_i\mathbb{E}(s_i) \\
&= e_i-\frac{N}{n}\frac{n}{N}e_i-\frac{N}{n}\mathbb{E}\left[\mathbb{E}(s_i\delta_i \mid s_i)\right]+\frac{N}{n}\frac{n}{N}y_i \\
&= y_i
\end{aligned}$$

It follows that the estimator (4.1) is unbiased for μ. This property does not depend on the prediction function $f(\cdot)$ chosen; the estimation method is unbiased even if $f(\cdot)$ is not an accurate prediction function for the given data. While the choice of $f(\cdot)$ does not affect the bias of $\hat{\mu}$, the accuracy of $f(\cdot)$ does affect the variance of $\hat{\mu}$.

4.2.3 Variance Estimation

We now provide a variance estimate of $\hat{\mu}$. The true variance of $\hat{\mu}$, as we derive in the Appendix, is

$$\begin{aligned}
\operatorname{var}(\hat{\mu}) = \frac{N-n}{nN}&\left[\frac{1}{N}\sum_{i=1}^{N}(y_i-e_i)^2+\frac{1}{N}\sum_{i=1}^{N}\operatorname{var}(f_i \mid s_i=1)\right] \\
&+\frac{N-n}{N^3}\sum_{i=1}^{N}\left[\operatorname{var}(g_i \mid s_i=0)-\operatorname{var}(f_i \mid s_i=1)\right] \\
&+\frac{1}{N^2}\sum_{i\neq j}\operatorname{cov}(\hat{y}_i,\hat{y}_j)
\end{aligned} \tag{4.2}$$

Our estimator from (4.1) can be rewritten as

$$\hat{\mu} = \frac{1}{N}\left[\sum_{i=1}^{N}h_i+\frac{N}{n}\sum_{i\in\mathcal{S}}(y_i-h_i)\right] \tag{4.3}$$

The first term in equation (4.3) is the mean of the predictions for each observation, and the second term in equation (4.3) is equivalent to the Horvitz-Thomspon estimator on the leave-one-out residuals (Horvitz and Thompson 1952). This is of similar form as the difference estimator presented in (Breidt and Opsomer 2017). Following Breidt and Opsomer (2017), we can estimate the variance of $\hat{\mu}$ as

$$\widehat{\operatorname{var}}(\hat{\mu}) = \frac{N-n}{nN}\left[\frac{1}{n}\sum_{i\in\mathcal{S}}(y_i-f_i)^2-\frac{1}{n(n-1)}\sum_{i,j\in\mathcal{S}}(y_i-f_i)(y_j-f_j)\right] \tag{4.4}$$

To relate the estimation of the variance in equation (4.4) to the true variance in equation (4.2), the first term in (4.4), $\frac{1}{n}\sum_{i\in\mathcal{S}}(y_i - f_i)^2$, is an unbiased estimate of the first term in (4.2), $\left[\frac{1}{N}\sum_{i=1}^{N}(y_i - e_i)^2 + \frac{1}{N}\sum_{i=1}^{N}\text{var}(f_i \mid s_i = 1)\right]$; and the second term in (4.4), $-\frac{1}{n(n-1)}\sum_{i,j\in\mathcal{S}}(y_i - f_i)(y_j - f_j)$, is an estimate of the last covariance term in (4.2), $\frac{1}{N^2}\sum_{i\neq j}\text{cov}(\hat{y}_i, \hat{y}_j)$. In the Appendix we give further justification for this variance estimate and derive some of its properties. Simulations in Section 4.3 suggest equation (4.4) is typically a conservative estimate of the true variance in expectation.

4.3 Results

In this section we apply the estimator to both simulated and real data. For the simulations we consider a variety of relationships between the covariate and response variables, both linear and non-linear, showing that the estimation method is unbiased, results in a reduction in variance compared to the sample mean, and that our variance estimate is accurate. We then apply the estimation method to real data.

4.3.1 Simulations

We consider three population settings: (1) a linear relationship between a covariate and a response, (2) a slightly non-linear relationship between a covariate and a response, and (3) a highly non-linear relationship between covariates and a response.

Linear Relationship

We first consider a simple setting. We let x be a one-dimensional covariate, where $x_i \sim N(0, 1)$, and $y_i = 3 + 2x_i + \epsilon_i$, where $\epsilon_i \sim N(0, 1)$. We consider population sizes of $N = 50, 500$, and 10000, where $\{(x_1, y_1), (x_2, y_2) \ldots, (x_u, y_u)\} \subset \{(x_1, y_1), (x_2, y_2) \ldots, (x_v, y_v)\}$ for $u < v$. We consider sample sizes of $n = 10, 100$, and 1000 (where $n < N$). For each of 10000 simulations, we take a simple random sample of size n from the population of size N and estimate μ using three methods: the sample mean, standard OLS adjustment, and our estimation method with OLS as the prediction function. For the OLS adjustment, we train an OLS regression model on the sample data to predict y given x, and apply that model to out-of-sample observations to predict their responses.

In Table 4.1 we present the simulation estimate of the true standard error (i.e., standard deviation of the 10000 estimates). We also present the mean of the estimated standard errors. In particular, for our method we calculate the estimated standard error of each sample using the formula in equation (4.4). For the sample mean we calculate the standard error for each sample as $\left[\frac{N-n}{nN} \times \frac{1}{n-1}\sum_{i\in\mathcal{S}}(y_i - \bar{y}_{\mathcal{S}})^2\right]^{1/2}$. Finally, for adjustment methods we calculate the standard error for each sample as $\left[\frac{N-n}{nN} \times \frac{1}{n-2}\sum_{i\in s} r_i^2\right]^{1/2}$ where r_i is the residual for the ith observation (Royall and Cumberland 1978).

The simulated standard error for the sample mean is over twice as large as the other two methods, while the simulated standard error for our method and OLS adjustment are nearly identical. This is the most

		True SE	Est. SE
$N = 50$	$n = 10$		
	Sample Mean	0.5093	0.5135
	OLS Adjustment	0.2331	0.2220
	Our Method with OLS	0.2424	0.2687
$N = 500$	$n = 10$		
	Sample Mean	0.7008	0.7017
	OLS Adjustment	0.3117	0.2925
	Our Method with OLS	0.3177	0.3502
	$n = 100$		
	Sample Mean	0.1984	0.2006
	OLS Adjustment	0.0842	0.0839
	Our Method with OLS	0.0843	0.0852
$N = 10000$	$n = 10$		
	Sample Mean	0.7118	0.7141
	OLS Adjustment	0.3381	0.3131
	Our Method with OLS	0.3428	0.3744
	$n = 100$		
	Sample Mean	0.2231	0.2246
	OLS Adjustment	0.0989	0.0987
	Our Method with OLS	0.0988	0.1022
	$n = 1000$		
	Sample Mean	0.0667	0.0667
	OLS Adjustment	0.0300	0.0297
	Our Method with OLS	0.0300	0.0298

Table 4.1: Simulation estimate of true standard error and estimated standard error for the sample mean, OLS adjustment, and our method using OLS prediction for a population with a linear relationship between x and y.

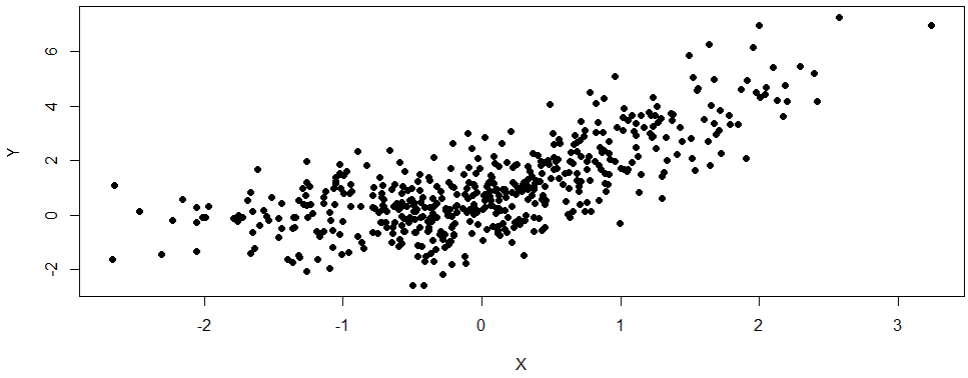

Figure 4.1: Slightly non-linear population.

generous setting for using OLS adjustment, and our estimation method has equivalent performance in terms of standard error.

For our method, the estimated standard error is slightly larger than than the simulated standard error in nearly every case. Based on this simulation, the formula for the estimated standard error gives a slightly conservative estimate of the true standard error, confirming the validity of the variance estimation equation (4.4). The conservativeness of the standard error estimate depends on n, with the overestimation decreasing as n increases. Finally, note that the bias for all three methods was not significantly different from 0; see the Appendix for further details.

Slightly Non-Linear Relationship

Next we perform a similar analysis, but with a slightly nonlinear relationship between x and y. We let $x_i \sim N(0,1)$ and let $y_i = \mathbb{1}(x_i \geq -0.5)(2x_i + 0.5) + \epsilon_i$, where $\epsilon_i \sim N(0,1)$. A scatterplot of this population can be seen in Figure 4.1. We use a population size of $N = 500$ and sample sizes of $n = 25, 50$, and 100. As before, we compare estimates using the sample mean, standard OLS adjustment, and our method with OLS. Results are in Table 4.2. In this table, the simulated estimate of the true standard error and the estimated standard error as calculated similarly as in Table 4.1. In Table 4.2 we also include a column for the estimated bias, calculated as the difference between the population response mean and the mean of the 10000 estimates, with (*) indicating that the bias is significantly different from 0 at the 0.05 level.

The bias for OLS adjustment is significantly different than 0 at all sample sizes. This demonstrates how traditional adjustment methods can fail (i.e., give a biased estimate) when model assumptions are even slightly not met. Our method, on the other hand, remains unbiased in the case of model misspecification. As with the previous simulation, the true standard error for our method and OLS imputation are nearly identical. Furthermore, the estimated standard error is generally a conservative estimate of the simulated true standard error. This simulation demonstrates how our estimation method is unbiased even when the

		True SE	Est. SE	Bias	
	Sample Mean	0.3226	0.3227	-0.0037	
$n = 25$	OLS Adjustment	0.2335	0.2229	0.0270	(*)
	Our Method with OLS	0.2346	0.2416	-0.0017	
	Sample Mean	0.2227	0.2219	-0.0020	
$n = 50$	OLS Adjustment	0.1576	0.1539	0.0116	(*)
	Our Method with OLS	0.1579	0.1601	-0.0017	
	Sample Mean	0.1464	0.1478	-0.0018	
$n = 100$	OLS Adjustment	0.1045	0.1029	0.0046	(*)
	Our Method with OLS	0.1045	0.1049	-0.0012	

Table 4.2: Simulation estimate of the true standard error, estimated standard error, and estimated bias using the sample mean, OLS adjustment, and our method with OLS for a slightly nonlinear population.

	Method	True SE	Est. SE	Bias	
	Sample Mean	0.7633	0.7698	0.0113	
	OLS Adjustment	1.3284	0.2011	0.0584	(*)
$n = 25$	Our Method with OLS	1.7033	1.6680	0.0219	
	Random Forest Adjustment	0.7286	0.3218	0.0605	(*)
	Our Method with Random Forest	0.7024	0.7317	0.0039	
	Sample Mean	0.5249	0.5305	-0.0048	
	OLS Adjustment	0.4511	0.2653	0.0169	(*)
$n = 50$	Our Method with OLS	0.4740	0.4764	-0.0015	
	Random Forest Adjustment	0.4622	0.2022	0.0364	(*)
	Our Method with Random Forest	0.4432	0.4505	-0.0057	
	Sample Mean	0.3533	0.3542	0.0028	
	OLS Adjustment	0.2603	0.2078	0.0095	(*)
$n = 100$	Our Method with OLS	0.2669	0.2687	0.0012	
	Random Forest Adjustment	0.2777	0.1179	0.0324	(*)
	Our Method with Random Forest	0.2677	0.2663	0.0023	

Table 4.3: Simulation estimate of true standard error, estimated standard error, and bias for the sample mean, OLS adjustment, our method with OLS, random forest adjustment, our method with random forest for a non-linear relationship between $\boldsymbol{x}$ and y.

prediction function is not the best fit for the given data.

Non-Linear Relationship

In the next simulation we consider a more complicated relationship between $\boldsymbol{x}$ and y. We generate a 20-dimensional $\boldsymbol{x}$, where $\boldsymbol{x}_i \sim N(0, I)$, and $y_i = x_{i,1}^3 + 2|x_{i,2}|^{1/2} + x_{i,1}x_{i,2} + \sin(x_{i,3}) + \epsilon_i$, where $\epsilon_i \sim N(0, 1)$. This simulation is nonlinear and includes interactions and noise variables. We use a population size of $N = 500$ and sample sizes of $n = 25, 50$, and 100. As estimation methods we use the sample mean, standard OLS adjustment, our method with OLS prediction, standard random forest adjustment, and our method with random forest. Results are in Table 4.3.

In this nonlinear setting, the standard random forest adjustment produces a biased estimate for the population mean response for all sample sizes. This may be due to the random forest overfitting to the sample. Our method with random forest is unbiased throughout. Despite a nonlinear relationship between

	Method	True SE	Est. SE	Bias	
	Sample Mean	1457.21	1462.54	12.78	
	OLS Adjustment	643.06	601.11	-2.57	
$n = 100$	Our Method with OLS	641.92	650.73	-0.23	
	Random Forest Adjustment	615.49	618.99	-41.46	(*)
	Our Method with Random Forest	613.04	618.66	7.29	
	Sample Mean	218.32	219.55	-1.48	
	OLS Adjustment	93.75	92.98	0.00	
$n = 1000$	Our Method with OLS	93.74	93.66	0.07	
	Random Forest Adjustment	78.16	78.33	-5.18	(*)
	Our Method with Random Forest	78.42	78.30	-0.42	

Table 4.4: Simulated true standard error, estimated standard error, and bias for average tuition from a simple random sample as estimated by the sample mean, OLS adjustment, our method with OLS, random forest adjustment, and our method with random forest.

the covariates and response, our method with OLS was unbiased throughout, while standard OLS adjustment was biased for $n = 50, 100$. However, the estimated standard error for OLS adjustment was greatly underestimated, especially for smaller sample sizes. Our estimated standard error, on the other hand, was slightly conservative throughout.

4.3.2 Application to College Tuition

We next apply the estimator to college tuition data by simulating sampling from the population of colleges. The IPEDS database (`https://nces.ed.gov/ipeds/`) provides annual data on all postsecondary educational institutions in the U.S. We focus on the public and not-for-profit private institutions that offer at least a bachelor's degree. We simulate taking samples from this population of colleges and universities. As the response variable we use tuition (in-state for public institutions) from the 2017-2018 academic year, and as covariates we use admission rate, graduation rate, student to staff ratio, highest degree offered (bachelors, master, doctoral), and whether the institution is public or private, all from the 2016-2017 academic year. There are a total of 1262 institutions in our study.

We first simulate taking simple random samples of size $n = 100, 1000$. For each sample size, we simulate 10000 simple random samples, and for each sample we predict the average tuition for all institutions using our estimation method using the sample mean, standard OLS adjustment, our method with OLS as the prediction function, standard random forest adjustment, and our method with random forest as the prediction function. Results are in Table 4.4. Additional results with more sample sizes can be found in the Appendix.

In Table 4.4 we can compare the standard error for each of the estimation methods. The standard error is nearly identical for our method with OLS and OLS adjustment, and similar for our method with random forest and random forest adjustment. All of these methods have smaller standard error than the sample mean. The reduction in true standard error from OLS to random forest demonstrate the advantages of using more sophisticated machine techniques as opposed to OLS. In fact, while the true standard error shrinks for both prediction methods of our estimator, the true standard error of our method using random forest

	Method	True SE	Est. SE	Bias
$n = 100$	Sample Mean	1480.35	1474.39	-2897.45
	OLS Adjustment	645.29	594.23	-421.88
	Our Method with OLS	644.76	642.12	-417.52
	Random Forest Adjustment	591.38	619.20	-812.23
	Our Method with Random Forest	591.40	618.40	-716.18
$n = 1000$	Sample Mean	206.00	220.34	-1192.33
	OLS Adjustment	93.24	92.60	-174.71
	Our Method with OLS	93.24	93.27	-174.62
	Random Forest Adjustment	73.84	78.64	-189.44
	Our Method with Random Forest	74.06	78.61	-184.84

Table 4.5: Simulated true standard error, estimated standard error, and bias for average tuition from a non-simple random sample as estimated by the sample mean, OLS adjustment, our method with OLS, random forest adjustment, and our method with random forest.

shrinks faster than OLS. See the Appendix for further details. A lower standard error can be one of the advantages of using machine learning methods over standard regression models.

The standard error estimate using standard OLS adjustment underestimates the true simulated standard error. This may be due to the OLS overfitting to outlier points. There are a few outlier colleges, namely two campuses of Brigham Young University and three campuses of Inter American University of Puerto Rico. These are private universities, but with much lower tuition than other private universities (about \$5000/year). Whether or not these universities are included in the sample can affect the accuracy of the estimated standard error for standard OLS adjustment. This does not affect our method with OLS.

Table 4.4 also gives the simulated bias and associated simulated estimate of the standard error for each method and sample size. Our method, with both OLS and random forest, remains fairly unbiased throughout. OLS adjustment is also unbiased, whereas the random forest adjustment is biased for every sample size. This again demonstrates how our method produces an unbiased estimate of the population mean response, even when the prediction function is itself biased.

In all of the previous simulations, we simulated taking simple random samples from the population. However, even well-designed surveys rarely result in a truly simple random sample in real life settings; if implementing an actual survey to colleges asking for the next academic year's tuition, response rates will vary. It is conceivable, for example, that colleges with lower tuition will have less money for administrative positions that would be responsible for completing surveys, leading to lower response rates for colleges with lower tuition. We are interested in how our estimation method performs under such settings. We simulate samples in which the probability of inclusion in the sample is a linear function of tuition, where the college with the highest tuition being twice as likely to be chosen as a member of the sample as the college with the lowest tuition. For each sample size of $n = 100, 1000$ we simulate 10000 nonprobability samples, estimating the mean 2017 tuition using the same estimation methods as earlier: our method with OLS and random forest prediction, OLS and random forest adjustment, and the sample mean. Results for additional samples sizes can be found in the Appendix.

Variable	ACS	CoreLogic
Lot Size	0	0.025
Value	0.138	0.002
Property Tax	0.180	0
# of People	0	N/A
Income	0	N/A

Table 4.6: Proportion of variables missing for ACS and CoreLogic data.

In Table 4.5 we give the simulated bias and simulated standard error. All five of the estimation methods are biased in estimating the mean tuition of all colleges for all sample sizes. This bias decreases as the sample size grows, as schools with lower tuition are more likely to be included in the sample, but nonetheless remains biased. The bias for OLS is smaller than the bias for random forest; this may because OLS extrapolates better than random forest. While our method and adjustment methods are biased, they are still an improvement over using simply the sample mean. The bias for our method is smaller than the associated bias using adjustment methods throughout.

4.3.3 Application to the American Community Survey

For our last application we apply our estimator to the American Community Survey, using property tax data as our population data set, to estimate average household income and the total number of people living in single family homes in Washtenaw County, MI.

As our population of interest we use parcel-level tax data provided by CoreLogic, accessed through the University of Michigan Library. CoreLogic aggregates publicly available parcel-level tax data, real estate transactions, and foreclosure data throughout the entire United States, with some records going back as far as 50 years. The CoreLogic data includes information for nearly every single land parcel in the United States, making it an ideal choice for a population data set. We restrict this data set to 2016 property tax observations located in Washtenaw County, MI, home to Ann Arbor. We consider single family residential homes. This gives 77910 observations from the CoreLogic data set.

The American Community Survey (ACS) is an annual survey given to millions of households and individuals in the United States each year. The ACS achieves a very high response rate, typically over 95%. By utilizing CoreLogic population data on individual land parcels, we can increase precision of estimates otherwise estimated using only the ACS. Similar to the CoreLogic data, we limit the ACS data to single-family homes in Washtenaw County, MI from 2016. There are 930 such households in the ACS data set. To demonstrate how our method performs when covariates have different levels of predictiveness of the response, we consider two different response variables: number of people living in each household and household income. Using our estimator, we estimate the total number of people and the average household income for families living in single-family homes in Washtenaw County, MI.

Common variables with low levels of missingness found in both the ACS and CoreLogic are lot size

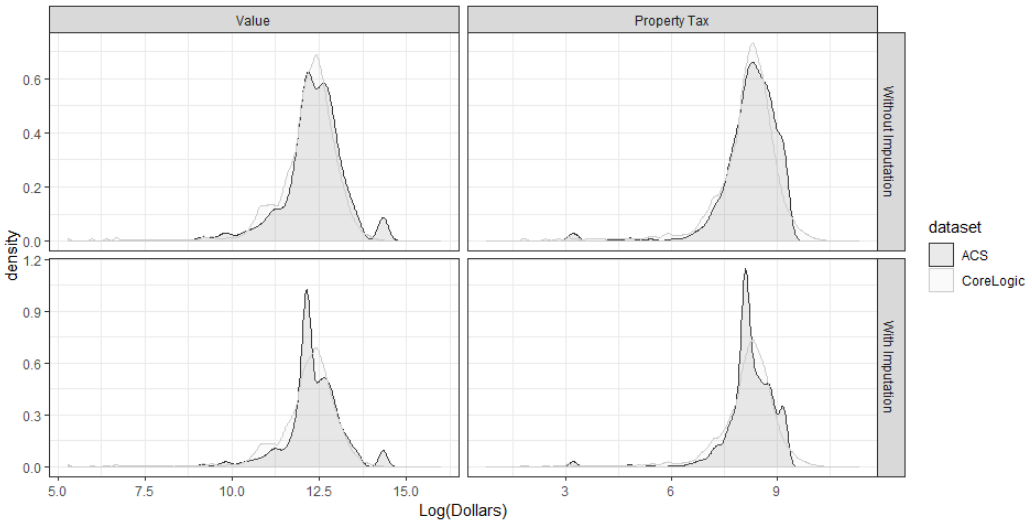

Figure 4.2: Density of home value and property tax before and after imputation for ACS and CoreLogic data sets.

	ACS	CoreLogic
<1 acre	0.6882	0.7090
between 1 and 10 acres	0.2333	0.2431
>10 acres	0.0785	0.0479

Table 4.7: Proportion of observation in each lot size category for each data set after imputation.

(categorized by <1 acre, between 1 and 10 acres, and >10 acres), property value, and property taxes paid. Table 4.6 gives the proportion of missingness for each variable. To deal with this missingness, we impute missing values as the mean of the CoreLogic variable. Note that there were some ACS observations that had a value of 0 for property tax; CoreLogic data says that zero property tax is not possible, so we consider those values to be missing. We impute missing lot size observations in the CoreLogic data set as <1 acre, the most common value in the both the ACS and CoreLogic data sets. Figure 4.2 shows the distribution of log-value and log-property tax for both data sets before and after imputation. Table 4.7 gives the distribution for lot size for both data sets after imputation.

All of the variables appear to have relatively similar distributions between both data sets. While the ACS data appears is *close* to a random sample, it is not exactly a simple random sample. This is known from the ACS sampling mechanism, but also from looking at the distributions of the variables. Evidence suggests that the ACS sample distribution of lot size was not drawn from the CoreLogic population distribution of lot size. The unbiasedness property of our method and the sample mean relies on simple random sampling, but the simulation in section 4.3.2 suggest that our method will result in a smaller bias and smaller standard error compared to using the sample mean.

Since the ACS data is sampled from a population frame, each observation in the ACS data set presumably has an exact match in the CoreLogic data. However, because of the imputed values and variables originally

	Total # of People		Mean Income	
	Estimate	Est. SE	Estimate	Est. SE
Our method with OLS	188463	3552	$109641	$3359
Our method with Random Forest	191400	3511	$117207	$3098
Sample Mean	191081	3552	$120515	$3762

Table 4.8: Mean and estimated standard error for total number of people living and household income for single family homes in Washtenaw County using the new method with OLS, random forest, and the sample mean.

measured slightly differently (e.g., exact property tax as measured by CoreLogic and binned estimates of property tax as measure by ACS), we do not have exact values to match one observation in the ACS data set to just one observation in the CoreLogic data set. For simplicity, we consider the entire population the set of observations in either ACS or CoreLogic; the observations in the CoreLogic data are considered observations not in the sample. Linking observations is not an insurmountable challenge; the institutions that performs the sampling (Census Bureau in this case) presumably has identifying information for each observation in the sample that can be used to link it to an observation in the population.

For estimating the total number of people living in single family homes and the average income of single family households in Washtenaw County, we use the sample mean and our method with both OLS and random forest as the prediction function. Results can be seen in Table 4.8.

We first estimate the total number of people in Washtenaw County that reside in single family homes. Using our method with OLS, we estimate 188463 people living in single family homes, with an estimated standard error of 3552. Using our method with random forest, we estimate 191400 people living in single family homes, with a standard error of 3511. Finally, using just the sample mean, we estimate 191081 people, with a standard error of 3552. The estimated standard error for these three estimates are very similar, and the same for the sample mean and our method with OLS. This is because the covariates of lot size, value, and property tax are not very predictive of number of people living in a single family residence; an OLS regression on the ACS data gives an adjusted R^2 of 0.008 and a residual standard error of 1.38. When the prediction function is not very predictive of the response variable, our estimation method performs roughly equivalently to using the sample mean.

Next we estimate average household income. Value, property tax, and lot size are more predictive of income than number of people in household, with an associated adjusted R^2 of 0.215. Again using our method with OLS and random forest and sample mean, we estimate mean household income to be $109641, $117207, and $120515, respectively, with estimated standard errors of $3359, $3098, and $3762. In this case the standard error does decrease when using our method compared to the sample mean. The more accurate the prediction function is, the better our method performs compared to the sample mean.

In both the number of people living in single family homes and average income examples above, the random forest outperformed OLS in terms of estimated standard error with our estimation method. This indicates that the random forest is a better prediction method than OLS for the given data, that there are

nonlinear relationships in the data that the OLS fails to accurately capture. This demonstrates how machine learning techniques can be used to improve estimates from traditional surveys.

4.4 Discussion

In this paper we developed a predictive inference method to unbiasedly estimate a population mean response when there is full knowledge of the population auxiliary data. Unbiasedness is guaranteed by estimating the bias of the prediction function using subsamples of the original random sample. Even in the case of model misspecification, the method produces accurate (i.e. unbiased) results. Our method works with a variety of data types and settings, allowing for the user to choose whichever prediction algorithm is best suited for the data at hand. This is in contrast to other methods for integrating population auxiliary data and sample data that are tailored to specific prediction methods, such as using lasso as the prediction method in McConville et al. (2017). Our method provides a general framework that can accommodate any of these prediction functions.

We demonstrated through simulation that our method outperforms standard imputation methods by guaranteeing exact unbiasedness without sacrificing standard error. Our estimator is similar to the difference estimator found in Breidt and Opsomer (2017); the leave-one-out nature of the prediction function makes our estimation method exactly unbiased, whereas the difference estimator is only asymptotically unbiased. Exact versus asymptotic unbiasedness can make a difference quantitatively and qualitatively. Theoretical guarantees rely on the sample being a simple random sample. However, even well-designed surveys are rarely truly simple random samples. As a suggested area of future research, our new estimation method could be extended to estimate functions of a population response from nonprobability samples when auxiliary population data is known.

There are many applications for this method that we have not explored. For example, our method can be applied to social media data. Social media companies have access to many features for each of their users. By having a random sample of users partake in a survey, unbiased estimates for the population of users can be obtained using our method. Our method can also be utilized by government agencies in possession of population data and the ability to link sample observations to population observations. Many of these data sets are anonymized when publicly released, such as the ACS data used earlier, but by knowing which observation in the population corresponds to each observation in the sample, unbiased estimates of the population mean response with lower standard error can be obtained.

Chapter 5

Discussion

The goal of this dissertation is to contribute to the evolving field of modern survey science. Traditional probability-based surveys are becoming increasingly difficult to implement, and it is unknown if traditional survey estimation methods will remain feasible in the future. Modern sources of data and modern predictive modeling algorithms have the potential to improve upon traditional survey estimation methods. We in particular consider the use of social media data in tracking survey responses and the use of population auxiliary data in combination with predictive modeling to improve survey estimates.

The process of finding a relationship between a given survey and data extracted from Twitter commonly consists of: collecting tweets over time that contain some word related to the given survey, calculating sentiment of the tweets over the time period, processing the sentiment of tweets (e.g. smoothing and adding lag), and finding the relationship between the processed tweets and survey responses. Our work demonstrates challenges of finding relationships in this way.

Individual researcher decisions can have a surprisingly large impact on the observed relationship. For example, choosing one sentiment method over another can change the conclusion from a strong relationship to no relationship, as we showed with "jobs" tweets and consumer confidence. This demonstrates how fragile such relationships can be and the need for robustness tests on these types of analyses. On a similar note, by optimizing over various parameters, it can be relatively easy to cherry-pick positive results. This makes standard significance tests invalid. Cherry-picking results is a problem not just in our specific example, but prevalent in nearly every application of statistics. One solution is to state decisions, with well-founded reasoning, beforehand, preventing the optimization of decisions during the course of the analysis.

Theoretical understanding for the relationship between social media data and survey responses is lacking and it is not always clear a priori what adjustments and decisions should be made, so empirical optimizations must be performed. Under this framework of optimization, if there is truly a relationship between survey responses and tweets containing a given word, we would expect the observed relationship under optimization to be much stronger than a similarly calculated relationship under optimization between survey responses

and all tweets. Thus, we should not be asking whether the observed relationship is significantly different from zero, but whether it is significantly different from *all* tweets. Our implementation of placebo tests addresses this issue.

Our work raises serious doubt as to whether survey responses can reliably be tracked using tweets containing a given word. We present one alternative: collecting tweets from a set of users over time. By focusing on tracking individual users longitudinally we ensure that the demographics of users is constant across the entire time period. We found evidence of a political signal around the 2016 election. However, similar as with the relationship between "Trump" tweets and presidential approval, it is possible that this signal is not much stronger than what we would observe among users completely unengaged with politics. Our placebo analysis framework may be able to be extended to answer such questions.

We successfully classified users as Democrats or Republicans. Future methods may be able to be developed, as well as current methods improved, to learn other demographic characteristics of users. With correct demographic information we can weight users to reflect the population of interest.

Another alternative for potentially improving alignment between survey responses and social media is to track users for which we know their survey response. That is, track users that have responded to the survey of interest. This may provide valuable insights on how users' tweets are related to various opinions. For example: do people have similar sentiment when responding to a private survey versus making a public statement on social media? Do users only tweet about topics in which they have strong feelings? Do users tweet when they change their opinion?

We may also be able to take advantage of the network structure of social media. We only lightly take advantage of this information in classifying users as Democrats or Republicans. By analyzing how information disseminates through a network, we might, for example, discover that we only need to track influential users (i.e. users with a high follower count), therefore decreasing the computation burden. We may also be able estimate a user's opinion on a particular subject based on the opinions of accounts they follow and interact with.

Many analyses with Twitter data tend to reduce the data to either sentiment or frequency. It is possible that by reducing the content of tweets to purely sentiment and frequency we are removing too much valuable information. Another suggestion that can be applied to tweets regardless of how they are collected is to more carefully consider the content of the tweets. Despite purposefully gathering tweets that are related to the signal of interest, there are likely to be tweets that are irrelevant. Furthermore, there are many aspects of public opinion, and correctly modeling the topics being discussed in a corpus might help to reveal those aspects. While our topic modeling method does not take into account auxiliary information by design, future areas of research may be able to take advantage of the auxiliary information in each tweet, such as the user and date.

As we demonstrate in chapter 3, auxiliary information can be very valuable. In the case of social media, this auxiliary information (e.g. user and account information, temporal tweeting patterns) is known for every

user (at least readily available for the social media platforms). Say, for example, that Twitter was interested in the number of political tweets that were being sent. Using a random subset of tweets (or users), topic modeling may be used to determine the proportion of tweets that are political. Then using the auxiliary information that is known about every user as covariates and the labels from topic modeling as the response, our estimation method from chapter 4 can be used to estimate the overall number of political tweets. With the large amount of covariates that can be generated from the auxiliary information, it can be relatively easy to overfit the prediction model to the sample covariates; the estimation method derived in chapter 4 ensures that the overfitting does not affect the overall bias of the estimator. Furthermore, the estimation method is unbiased and gives an accurate estimate of the standard error even if assumptions for the prediction model are not met.

The estimation method might also be useful for combining traditional survey responses and social media data. Most of the analyses we have performed with social media are tracking rather than supplementing surveys responses. Social media may be a way to gather information on otherwise hard-to-reach populations, so combining the two sources of data may provide valuable insights. For example, if we are interested in the number of people that are politically engaged, we might have daily estimates of political activity from survey responses and social media posts, and some true measure of political activeness on only certain days, such as election days. Our estimator may be able to be used in this context to estimate the average political activity over the given time frame.

Our estimation method works for simple random samples, but it is not guaranteed to be unbiased for non-simple random samples. While this is a reasonable assumption when our population is all social media posts or users, Twitter users are not a random sample of the population. As a next step, our estimation method may be able to be extended to other types of samples. Many general methods exist for various types of samples, and these methods could potentially be incorporated into our framework.

This dissertation makes valuable contributions to the field of survey science. New forms of media and data and new machine learning prediction models will be created in the future, and methods that we lay out in this dissertation provide a foundation for how to incorporate the new data and prediction models into future survey estimation.

Bibliography

Ajao, Oluwaseun, Jun Hong, and Weiru Liu (2015). "A survey of location inference techniques on Twitter". In: *Journal of Information Science* 41.6, pp. 855–864.

Alvarez-Melis, David and Martin Saveski (2016). "Topic modeling in twitter: Aggregating tweets by conversations". In: *Tenth international AAAI conference on web and social media*.

Antenucci, Dolan et al. (2014). *Using social media to measure labor market flows*. Tech. rep. National Bureau of Economic Research.

Athey, Susan and Guido W. Imbens (2017). "The State of Applied Econometrics: Causality and Policy Evaluation". In: *Journal of Economic Perspectives* 31.2, pp. 3–32. DOI: 10.1257/jep.31.2.3.

Barberá, Pablo and Gonzalo Rivero (2015). "Understanding the political representativeness of Twitter users". In: *Social Science Computer Review* 33.6, pp. 712–729.

Benoit, Kenneth et al. (2018). "quanteda: An R package for the quantitative analysis of textual data". In: *Journal of Open Source Software* 3.30, p. 774. DOI: 10.21105/joss.00774. URL: https://quanteda.io.

Berry, Michael W and Murray Browne (2005). *Understanding search engines: mathematical modeling and text retrieval*. SIAM.

Bethlehem, Jelke (2009). "The rise of survey sampling". In: CBS Discussion Paper: 1572-0314 09015.

Bicalho, Paulo et al. (2017). "A general framework to expand short text for topic modeling". In: *Information Sciences* 393, pp. 66–81.

Blei, David M, Andrew Y Ng, and Michael I Jordan (2003). "Latent dirichlet allocation". In: *Journal of machine Learning research* 3.Jan, pp. 993–1022.

Bowley, Arthur Lyon (1906). "Address to the Economic Science and Statistics Section of the British Association for the Advancement of Science, York, 1906". In: *Journal of the Royal Statistical Society* 69.3, pp. 540–558.

Boyd-Graber, Jordan L. and David M. Blei (2010). "Syntactic Topic Models". In: *CoRR* abs/1002.4665. arXiv: 1002.4665. URL: http://arxiv.org/abs/1002.4665.

Breidt, F Jay and Jean D Opsomer (2000). "Local polynomial regression estimators in survey sampling". In: *Annals of statistics* 28.4, pp. 1026–1053.

— (2017). "Model-assisted survey estimation with modern prediction techniques". In: *Statistical Science* 32.2, pp. 190–205.

Buelens, Bart, Joep Burger, and Jan van den Brakel (2015). *Predictive inference for non-probability samples: a simulation study*. Statistics Netherlands.

Ceron, Andrea et al. (2014). "Every tweet counts? How sentiment analysis of social media can improve our knowledge of citizens' political preferences with an application to Italy and France". In: *New media & society* 16.2, pp. 340–358.

Cheng, Yizong and George M Church (2000). "Biclustering of expression data." In: *Ismb*. Vol. 8. 2000, pp. 93–103.

Chipperfield, James, Julia Chessman, and Russell Lim (2012). "Ccombining household surveys using mass imputation to estimate population totals". In: *Australian & New Zealand Journal of Statistics* 54.2, pp. 223–238. DOI: 10.1111/j.1467-842X.2012.00666.x.

Choi, Hyunyoung and Hal Varian (2009). "Predicting initial claims for unemployment benefits". In: *Google Inc*, pp. 1–5.

— (2012). "Predicting the present with Google Trends". In: *Economic record* 88, pp. 2–9.

Cody, E. M. et al. (Aug. 2016). "Public Opinion Polling with Twitter". In: *ArXiv e-prints*. arXiv: 1608.02024 [physics.soc-ph].

Conrad, Frederick G et al. (2015). "A "collective-vs-self" hypothesis for when Twitter and survey data tell the same story". In: *Annual Conference of the American Association for Public Opinion Research, Hollywood, FL*.

Conrad, Frederick G et al. (2019). "Social media as an alternative to surveys of opinions about the economy". In: *Social Science Computer Review*.

Daas, Piet J. H. and Marco J.H. Puts (2014). "Social Media Sentiment and Consumer Confidence". In: ECB Statistics Paper 5. DOI: 10.2866/11606.

Dagdoug, Mehdi, Camelia Goga, and David Haziza (2020). "Model-assisted estimation through random forests in finite population sampling". In: *arXiv preprint arXiv:2002.09736*.

De Heer, W and E De Leeuw (2002). "Trends in household survey nonresponse: A longitudinal and international comparison". In: *Survey nonresponse* 41, pp. 41–54.

Deerwester, Scott et al. (1990). "Indexing by latent semantic analysis". In: *Journal of the American society for information science* 41.6, pp. 391–407.

Dhillon, Inderjit S (2001). "Co-clustering documents and words using bipartite spectral graph partitioning". In: *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 269–274.

Dhillon, Inderjit S, Subramanyam Mallela, and Dharmendra S Modha (2003). "Information-theoretic co-clustering". In: *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 89–98.

DiNardo, John E. and Jorn-Steffen Pischke (1996). "The Returns to Computer Use Revisited: Have Pencils Changed the Wage Structure Too?" In: *National Bureau of Economic Research Working Paper Series*.

Dong, Qi, Michael R Elliott, and Trivellore E Raghunathan (2014). "A nonparametric method to generate synthetic populations to adjust for complex sampling design features". In: *Survey methodology* 40.1, p. 29.

Dugas, Andrea Freyer et al. (2013). "Influenza forecasting with Google flu trends". In: *PloS one* 8.2.

Feuerriegel, Stefan and Nicolas Proellochs (2018). *SentimentAnalysis: Dictionary-Based Sentiment Analysis*. R package version 1.3-2. URL: `https://CRAN.R-project.org/package=SentimentAnalysis`.

Golder, Scott A. and Michael W. Macy (2011). "Diurnal and Seasonal Mood Vary with Work, Sleep, and Daylength Across Diverse Cultures". In: *Science* 333.6051, pp. 1878–1881. ISSN: 0036-8075. DOI: `10.1126/science.1202775`.

Graefe, Andreas (2014). "Accuracy of vote expectation surveys in forecasting elections". In: *Public Opinion Quarterly* 78.S1, pp. 204–232.

Guerra, Pedro Calais et al. (2017). "Antagonism also Flows through Retweets: The Impact of Out-of-Context Quotes in Opinion Polarization Analysis". In: *Proceedings of the Eleventh International AAAI Conference on Web and Social Media*.

Hajjem, Malek and Chiraz Latiri (2017). "Combining IR and LDA topic modeling for filtering microblogs". In: *Procedia Computer Science* 112, pp. 761–770.

Hansen, Morris H and William N Hurwitz (1943). "On the theory of sampling from finite populations". In: *The Annals of Mathematical Statistics* 14.4, pp. 333–362.

Hansen, Morris H, William G Madow, and Benjamin J Tepping (1983). "An evaluation of model-dependent and probability-sampling inferences in sample surveys". In: *Journal of the American Statistical Association* 78.384, pp. 776–793.

Hartigan, John A (1972). "Direct clustering of a data matrix". In: *Journal of the american statistical association* 67.337, pp. 123–129.

Hartley, H.O. and A. Ross (1954). "Unbiased Ratio Estimators". In: *Nature* 174, pp. 270–271. DOI: `https://doi.org/10.1038/174270a0`.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

Hawkins, Douglas M (2004). "The problem of overfitting". In: *Journal of chemical information and computer sciences* 44.1, pp. 1–12.

Hong, Liangjie and Brian D Davison (2010). "Empirical Study of Topic Modeling in Twitter". In: *1st Workshop on Social Media Analytics*, pp. 80–88.

Horvitz, D. G. and D. J. Thompson (1952). "A Generalization of Sampling Without Replacement From a Finite Universe". In: *Journal of the American Statistical Association* 47.260, pp. 663–685. ISSN: 01621459.

Hsieh, Yuli Patrick and Joe Murphy (2017). "Total Twitter Error". In: *Total Survey Error in Practice*. Wiley-Blackwell. Chap. 2, pp. 23–46. ISBN: 9781119041702. DOI: `10.1002/9781119041702.ch2`.

Hu, Minqing and Bing Liu (2004). "Mining and Summarizing Customer Reviews". In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '04.

Seattle, WA, USA: Association for Computing Machinery, 168–177. ISBN: 1581138881. DOI: 10.1145/1014052.1014073.

Hutto, CJ and Eric Gilbert (2014). "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text". In: *Association for the Advancement of Artificial Intelligence*.

Jin, Ou et al. (2011). "Transferring topical knowledge from auxiliary long texts for short text clustering". In: *Proceedings of the 20th ACM international conference on Information and knowledge management*, pp. 775–784.

Jun, Seung-Pyo, Hyoung Sun Yoo, and San Choi (2018). "Ten years of research change using Google Trends: From the perspective of big data utilizations and applications". In: *Technological forecasting and social change* 130, pp. 69–87.

Jurgens, David et al. (2015). "Geolocation prediction in twitter using social networks: A critical analysis and review of current practice". In: *Ninth International AAAI Conference on Web and Social Media*.

Karami, Amir et al. (2018a). "Characterizing diabetes, diet, exercise, and obesity comments on Twitter". In: *International Journal of Information Management* 38.1, pp. 1–6.

Karami, Amir et al. (2018b). "Fuzzy approach topic discovery in health and medical corpora". In: *International Journal of Fuzzy Systems* 20.4, pp. 1334–1345.

Kaufman, Leonard and Peter J Rousseeuw (1987). "Clustering by means of medoids". In:

Keeter, Scott (2012). "Presidential address: survey research, its new frontiers, and democracy". In: *The Public Opinion Quarterly* 76.3, pp. 600–608.

Kennedy, Courtney et al. (2016). "Evaluating online nonprobability surveys". In: *Pew Research Center* 61.

Kim, Jae Kwang and J. N. K. Rao (Dec. 2011). "Combining data from two independent surveys: a model-assisted approach". In: *Biometrika* 99.1, pp. 85–100. ISSN: 0006-3444. DOI: 10.1093/biomet/asr063.

Landauer, Thomas K, Peter W Foltz, and Darrell Laham (1998). "An introduction to latent semantic analysis". In: *Discourse processes* 25.2-3, pp. 259–284.

Lazer, David et al. (2014). "The parable of Google Flu: traps in big data analysis". In: *Science* 343.6176, pp. 1203–1205.

Li, Ximing et al. (2018). "Filtering out the noise in short text topic modeling". In: *Information Sciences* 456, pp. 83–96.

Little, Roderick JA (1993). "Post-stratification: a modeler's perspective". In: *Journal of the American Statistical Association* 88.423, pp. 1001–1012.

Liu, Bing, Minqing Hu, and Junsheng Cheng (2005). "Opinion Observer: Analyzing and Comparing Opinions on the Web". In: *Proceedings of the 14th International Conference on World Wide Web*. WWW '05. Chiba, Japan: Association for Computing Machinery, 342–351. ISBN: 1595930469. DOI: 10.1145/1060745.1060797.

Liu, Yutao, Qixuan Chen, and Andrew Gelman (2019). "Bayesian Inference for Sample Surveys in the Presence of High-Dimensional Auxiliary Information". Joint Statistical Meetings. URL: https://ww2.amstat.org/meetings/jsm/2019/onlineprogram/AbstractDetails.cfm?abstractid=306655.

Massey, Douglas S and Roger Tourangeau (2013). "Introduction: New Challenges to Social Measurement". In: *The Annals of the American Academy of Political and Social Science* 645.1, pp. 6–22.

McConville, Kelly S. et al. (Apr. 2017). "Model-Assisted Survey Regression Estimation with the Lasso". In: *Journal of Survey Statistics and Methodology* 5.2, pp. 131–158. ISSN: 2325-0984. DOI: 10.1093/jssam/smw041.

Mehrotra, Rishabh et al. (2013). "Improving lda topic models for microblogs via tweet pooling and automatic labeling". In: *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pp. 889–892.

Mickey, M.R. (Sept. 1959). "Some Finite Population Unbiased Ratio and Regression Estimators". In: *Journal of the Amerian Statistical Association* 54.287, pp. 594–612.

Mirkin, Boris (2013). *Mathematical classification and clustering*. Vol. 11. Springer Science & Business Media.

Murphy, Joe et al. (2014). "Social Media in Public Opinion Research Executive Summary of the Aapor Task Force on Emerging Technologies in Public Opinion Research". In: *Public Opinion Quarterly* 78.4, pp. 788–794. DOI: 10.1093/poq/nfu053.

Neyman, Jerzy (1934). "On the Two Different Aspects of the Representative Method: The Method of Stratified Sampling and the Method of Purposeful Selection". In: *Journal of the Royal Statistical Society* 97.4, pp. 558–625.

Nigam, Kamal et al. (2000). "Text classification from labeled and unlabeled documents using EM". In: *Machine learning* 39.2-3, pp. 103–134.

O'Connor, Brendan et al. (2010). "From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series". In: *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*.

Pasek, Josh et al. (2018). "The stability of economic correlations over time: identifying conditions under which survey tracking polls and Twitter sentiment yield similar conclusions". In: *Public Opinion Quarterly* 82.3, pp. 470–492.

Pasek, Josh et al. (2019). "Who's Tweeting About the President? What Big Survey Data Can Tell Us About Digital Traces?" In: *Social Science Computer Review*.

Pedrosa, Gabriel et al. (2016). "Topic modeling for short texts with co-occurrence frequency-based expansion". In: *2016 5th Brazilian Conference on Intelligent Systems (BRACIS)*. IEEE, pp. 277–282.

Popken, Ben (2018). "Twitter deleted Russian troll tweets. So we published more than 200,000 of them." In: *NBCNews.com*.

Preoţiuc-Pietro, Daniel et al. (2015). "Studying user income through language, behaviour and affect in social media". In: *PloS one* 10.9.

Qomariyah, Siti, Nur Iriawan, and Kartika Fithriasari (2019). "Topic modeling Twitter data using Latent Dirichlet Allocation and Latent Semantic Analysis". In: *AIP Conference Proceedings* 2194.1, p. 020093. DOI: 10.1063/1.5139825.

Quan, Xiaojun et al. (2015). "Short and sparse text topic modeling via self-aggregation". In: *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Rafei, Ali, Carol A C Flannagan, and Michael R Elliott (Feb. 2020). "Big Data for Finite Population Inference: Applying Quasi-Random Approaches to Naturalistic Driving Data Using Bayesian Additive Regression Trees". In: *Journal of Survey Statistics and Methodology* 8.1, pp. 148–180. ISSN: 2325-0984. DOI: 10.1093/jssam/smz060.

Resnik, Philip et al. (2015). "Beyond LDA: Exploring Supervised Topic Modeling for Depression-Related Language in Twitter". In: *CLPsych@HLT-NAACL*.

Rousseeuw, Peter J and L Kaufman (1990). "Finding groups in data". In: *Hoboken: Wiley Online Library*.

Royall, Richard M. and William G. Cumberland (1978). "Variance Estimation in Finite Population Sampling". In: *Journal of the American Statistical Association* 73.362, pp. 351–358. ISSN: 01621459.

Särndal, Carl-Erik, Bengt Swensson, and Jan Wretman (2003). *Model assisted survey sampling*. Springer Science & Business Media.

Schober, Michael F et al. (2016). "Social media analyses for social measurement". In: *Public opinion quarterly* 80.1, pp. 180–211.

Schubert, Erich and Peter J Rousseeuw (2019). "Faster k-Medoids Clustering: Improving the PAM, CLARA, and CLARANS Algorithms". In: *Similarity Search and Applications*. Ed. by Giuseppe Amato et al. Springer International Publishing, pp. 171–187. ISBN: 978-3-030-32047-8.

Schulz, Axel et al. (2013). "A multi-indicator approach for geolocalization of tweets". In: *Seventh international AAAI conference on weblogs and social media*.

Shen, Judy Hanwen et al. (2018). "Darling or babygirl? investigating stylistic bias in sentiment analysis". In: *5th Workshop on Fairness, Accountability, and Transparency in Machine Learning (FATML)*.

Silver, Nate (Mar. 2017). *How We're Tracking Donald Trump's Approval Ratings*. URL: https://fivethirtyeight.com/features/how-were-tracking-donald-trumps-approval-ratings/ (visited on 03/12/2018).

Smith, Benjamin K and Abel Gustafson (2017). "Using wikipedia to predict election outcomes: online behavior as a predictor of voting". In: *Public Opinion Quarterly* 81.3, pp. 714–735.

Subramani, Jambulingam (2013). "Generalized modified ratio estimator for estimation of finite population mean". In: *Journal of Modern Applied Statistical Methods* 12.2, p. 7.

Tang, Jian et al. (2014). "Understanding the limiting factors of topic modeling via posterior contraction analysis". In: *International Conference on Machine Learning*, pp. 190–198.

Tibshirani, Robert, Guenther Walther, and Trevor Hastie (2001). "Estimating the number of clusters in a data set via the gap statistic". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63.2, pp. 411–423.

Tumasjan, Andranik et al. (2010). "Predicting elections with twitter: What 140 characters reveal about political sentiment". In: *Fourth international AAAI conference on weblogs and social media*.

Valliant, Richard, Alan H Dorfman, and Richard M Royall (2000). *Finite population sampling and inference: a prediction approach*. 04; QA276. 6, V3. John Wiley New York.

Vigen, Tyler (2014). URL: `http://www.tylervigen.com/spurious-correlations` (visited on 04/29/2018).

Wang, Xuerui and Andrew McCallum (2006). "Topics over time: a non-Markov continuous-time model of topical trends". In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 424–433.

Williams, W.H. (June 1961). "Generating Unbiased Ratio and Regression Estimators". In: *Biometrics* 17.2, pp. 267–274.

— (Mar. 1962). "On Two Methods of Unbiased Estimation with Auxiliary Variates". In: *Journal of The American Statistical Association* 57, pp. 184–186.

Wilson, Theresa, Janyce Wiebe, and Paul Hoffmann (2005). "Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis". In: *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*. HLT '05. Vancouver, British Columbia, Canada: Association for Computational Linguistics, 347–354. DOI: `10.3115/1220575.1220619`.

Xu, Wei, Xin Liu, and Yihong Gong (2003). "Document clustering based on non-negative matrix factorization". In: *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pp. 267–273.

Yan, Xiaohui et al. (2012). "Clustering short text using ncut-weighted non-negative matrix factorization". In: *Proceedings of the 21st ACM international conference on Information and knowledge management*, pp. 2259–2262.

Yan, Xiaohui et al. (2013). "A biterm topic model for short texts". In: *Proceedings of the 22nd international conference on World Wide Web*, pp. 1445–1456.

Yang, Shihao, Mauricio Santillana, and Samuel C Kou (2015). "Accurate estimation of influenza epidemics using Google search data via ARGO". In: *Proceedings of the National Academy of Sciences* 112.47, pp. 14473–14478.

Yang, Shuang-Hong et al. (2014). "Large-scale High-precision Topic Modeling on Twitter". In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '14. New York, New York, USA: ACM, pp. 1907–1916. ISBN: 978-1-4503-2956-9. DOI: `10.1145/2623330.2623336`.

Young, Lori and Stuart Soroka (2012). "Affective News: The Automated Coding of Sentiment in Political Texts". In: *Political Communication* 29.2, pp. 205–231. DOI: `10.1080/10584609.2012.671234`.

Zhao, Wayne Xin et al. (2011). "Comparing Twitter and Traditional Media Using Topic Models". In: *Advances in Information Retrieval*. Ed. by Paul Clough et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 338–349. ISBN: 978-3-642-20161-5.

Zuo, Yuan, Jichang Zhao, and Ke Xu (2016). "Word network topic model: a simple but general solution for short and imbalanced texts". In: *Knowledge and Information Systems* 48.2, pp. 379–398.

Zuo, Yuan et al. (2016). "Topic modeling of short texts: A pseudo-document view". In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 2105–2114.

Appendix A: Data and Code Availability

"Jobs" Tweets Analysis

The average daily ICS from 2008 to 2014 as well as the average daily measures from the five SCA questions on which the ICS is based can be found at `https://www.openicpsr.org/openicpsr/project/109581/version/V1/view/`. The daily average sentiment scores for jobs tweets from 2008 to 2014 computed with five different tools can be found at `https://www.openicpsr.org/openicpsr/project/109581/version/V1/view/`

A script for a Shiny app that allows a user to assess the relationship between the sentiment data and survey responses, along with a script that allows the user to reproduce all results and figures reported in these analyses, can be found at `https://github.com/robynferg/Twitter_ICS`. R code for all results and figures can be found at `https://github.com/robynferg/Twitter_ICS/blob/master/ResultsFigures.R`

"Trump" Tweets and Politically Active Users Analysis

Presidential approval was downloaded from the website FiveThirtyEight, available at `https://projects.fivethirtyeight.com/trump-approval-ratings/?ex_cid=rrpromo`. Data and scripts for replicating all analyses in this paper can be found at `https://github.com/robynferg/Tracking_Presidential_Approval_with_Twitter`. The Twitter data available online used in the placebo analysis gives the daily average sentiment for tweets containing each of the placebo words. To protect the privacy of the politically active users, we have blinded the user name and tweet content in the data set available online.

Estimation Method with Auxiliary Data

Data and script to reproduce most results presented in chapter 4 can be found at `https://github.com/robynferg/Population-Auxiliary-Data-and-Sample-Data`. All analyses were done in R. We provide a script containing a function to run our estimator with OLS, random forest, or sample mean. We provide data and code to reproduce all results for simulated data and simulated sampling of colleges. The CoreLogic

data is not publicly available, so we provide population means needed to apply the new estimator using OLS.

Appendix B: SCA Questions

The five survey questions used to calculate ICS are:

1. "We are interested in how people are getting along financially these days. Would you say that you (and your family living there) are better off or worse off financially than you were a year ago?"

2. "Now looking ahead–do you think that a year from now you (and your family living there) will be better off financially, or worse off, or just about the same as now?"

3. "Now turning to business conditions in the country as a whole–do you think that during the next twelve months we'll have good times financially, or bad times, or what?"

4. "Looking ahead, which would you say is more likely–that in the country as a whole we'll have continuous good times during the next five years or so, or that we will have periods of widespread unemployment or depression, or what?"

5. "About the big things people buy for their homes–such as furniture, a refrigerator, stove, television, and things like that. Generally speaking, do you think now is a good or bad time for people to buy major household items?"

Appendix C: "Jobs" Tweets Sorting Algorithm

In this section we give examples of tweets that fall into each of the five "jobs" tweets categories: news/politics, personal, advertisements, other, and junk. We then give details of our algorithm to classify each tweet as a member of one of the classes.

1. **News and Politics**: This type of tweet generally refers to either current events on the national level or political opinions. Many of these tweets have to do with the U.S. economy as a whole.
   - "President Obama adds 244,000 new jobs this month. Thank you Mr. President @Whitehouse #tcot #teaparty #p2"
   - "Do Republicans really believe governors create jobs? This is so gross."
   - "want more jobs &better economy? elect ppl who will CUT corp taxes & regulations. It's not rocket science #teaparty #icaucus #sgp #tcot #gop"
2. **Personal**: Tweets in this category refer to one's individual job, many times commenting on job satisfaction or change in employment status.
   - "I got 3 emails about photography jobs in one day. this rocks!!!"
   - "Going out to apply for jobs. Yay?"
   - "wasting time on computer when I should be applying for jobs"
   - "Finally off from both of my 2 jobs today - need time to catch up on sleep & exercise!"
3. **Advertisements**: Tweets in this category display jobs available in various fields and various cities. Many of these are through a 'Tweet My Jobs' third party service. Despite referring to actual jobs, we don't expect these tweets to have much relationship with consumer confidence since they do not provide any opinion.
   - "Found 50 new jobs in Cincinnati, OH - check it at $<< link >>$"
   - "DENTAL ASSISTANT - Las Vegas, NV ($<< link >>$) Get Dental Assistant Jobs"

- "Sales Information Analyst - Jobs in Ireland"

4. **Other**: Tweets in the *Other* category are usually articles or lists, unrelated to current economic events, but typically having to do with employment in some way. For example, more articles may be written about recession-proof jobs during a recession.

    - "Green Jobs Czar Says 'White Polluters' Steered Poison Into Minority Communities << *link* >>"
    - "<< *link* >> Jobs, Bartending Secrets Revealed << *link* >>"
    - "The Top 5 Reccession-Proof Jobs (Chart) - << *link* >>"

5. **Junk**: The *jobs* mentioned in junk tweets refer to something other than employment. The most common include Steve Jobs, the TV show Dirty Jobs, and jobs of a sexual nature. Junk tweets should be independent of economic conditions and consumer confidence.

    - "Steve Jobs... I'm really proud of you, and I'ma let u finish... but Moses had one of the best tablets of all time"
    - "My first ex-boyfriend was in a Dirty Jobs show with Mike Rowe. The one about making shark repellent from actual sharks."

We create an algorithm to sort tweets into one of the five categories listed above. There are many aspects of tweets that make building a perfect classifier nearly impossible, such as the 140 character limit, slang words, misspellings, and unknown intentions of the user (sarcasm, etc.). Our classification algorithm works as follows for an individual tweet:

1. If the tweets contains at least one word from a user-defined list of junk words, that tweet is classified as junk.

    - For junk words we used: *steve, apple, iphone, itunes, ipad, mac, wozniak, gates, dirty, blow, hand, whack, nut*

2. Otherwise, if the tweet contains an ad word or user name contains a user ad word, that tweet is classified as an advertisement.

    - For ad words we used: *#hiring, #jobs* (these are vary common among 'tweet my jobs' tweets)
    - For ad user words we used: *job, tmj, career*

3. Otherwise if the tweet contains at least one word of a list of news/politics words or user name contain user news/politics word, the tweet is classified as news/politics.

    - For news/politics words we used: *obama, clinton, trump, mcconnell, ryan, boehner, potus, cantor, palin, teaparty, democrat, repbulic, mccain, romney, trade, taxes, senate, president, gop*

		Algorithm				
		Advertisement	Junk	News/Politics	Other	Personal
Hand-Classification	Advertisement	183	7	4	45	4
	Junk	0	41	0	5	5
	News/Politics	2	3	25	15	12
	Other	4	2	2	39	2
	Personal	0	2	2	6	90

Table 5.1: Comparison between hand classification and classification as given by the algorithm for a random sample of 500 tweets.

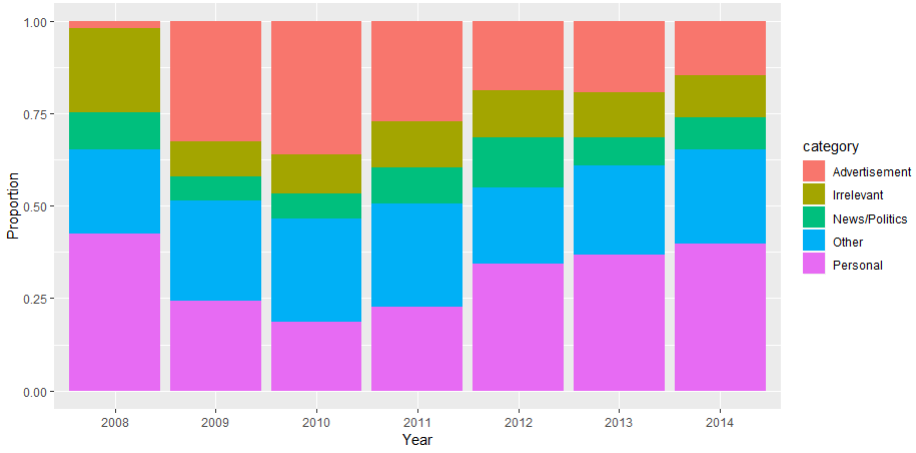

Figure 5.1: Proportion of "jobs" tweets belonging to each category by year.

- For news/politics user words we used: *news*

4. Otherwise, if the tweet contains a url, it is classified as other.

5. Otherwise, the tweet is classified as personal.

To verify the accuracy of this algorithm, we randomly sampled 500 tweets and hand-classified them into one of the above five categories. Table 5.1 compares the hand classification to the classification as given by the algorithm. About 75% of these tweets were classified correctly (Cohen's $\kappa = 0.67$). The most difficult category for the algorithm was *other*. If we remove the *other* category, the algorithm accuracy jumps to 89% (Cohen's $\kappa = 0.83$).

The proportion of "jobs" tweets in each category varies from year to year. Figure 5.1 shows the distribution from year to year. On average, 8% of the tweets were *news/politics*, 28% were *personal*, 27% were *advertisements*, 12% were *junk*, and 24% were *other*.

Appendix D: Word Clusters

In this appendix we give words from the set of validation users as sorted into clusters. Tables 5.2 and 5.2 give the words as sorted by our algorithm. Tables **??** and **??** give the most frequent words in each cluster as given by LDA.

Latent Topic	Words in Cluster
1	**storm**, 35, 60, 70, accurayno, accuweather, afternoon, ago, aim, along, alreadi, also, appear, approach, around, associ, away, batter, bay, blast, blizzard, breaking, breakingweath, brief, bring, bud, canada, capabl, carolina, categori, caus, central, chicago, chill, coast, coastal, colder, counti, country, coupl, creat, cross, current, dakota, damag, delay, deliv, despit, discuss, disrupt, dorian, drench, drop, due, east, eastern, elev, england, enough, episod, erupt, evan, even, expect, eye, farther, fast, forecast, friday, gust, gusti, hail, higher, hit, hurrican, hurricanedorian, impact, indian, intens, isol, juli, larg, latest, least, level, lightn, lik, listen, local, lorenzo, lower, major, maximum, mediterranean, mid, midweek, midwest, mile, mississippi, missouri, monday, morn, morning, mov, move, mph, multipl, myer, nebraska, night, northeast, northern, northward, northwest, northwestern, novemb, pack, part, path, peninsula, pennsylvania, period, plain, pois, possibl, potent, potenti, power, prior, produc, race, radar, region, relief, remain, resid, rest, road, rough, seek, send, separ, seri, sever, shelter, side, sight, slow, south, southern, spin, spread, states, strengthen, strong, surf, sustain, sweep, system, thanksgiv, threaten, thunderstorm, thursday, tier, tornado, toward, track, troubl, tuesday, typic, upper, valley, warn, washington, wednesday, weekend, west, wide, wind, winterlik, wintri, worst, york
2	**10**, age, cdt, evacu, fight4birthconrol, houston, nearli
3	**health**, 50, abil, abort, aca, access, accur, act, action, activist, administr, administration, advanc, advoc, affect, afford, alway, among, and, answer, appoint, appointment, ask, assault, attack, attempt, awar, ban, bansoffmybodi, basic, believ, believesurvivor, benefit, beyond, birth, black, blavityxpp, block, bodi, breast, busi, cancer, cannot, care, case, center, chat, choos, color, commit, common, commun, confirm, congratul, congress, contracept, control, cost, countri, court, cover, coverag, crisi, critic, cultur, deal, decis, deserv, die, differ, digniti, discrimin, doctor, don, door, drleanawen, ed, educ, effort, emerg, ensur, equal, essenti, everi, everyon, everyth, exam, experienc, expert, face, fact, famili, fight, find, forc, forward, freedom, full, fund, futur, gag, gender, gener, get, getcov, give, hand, harm, heard, heart, help, hiv, human, ident, iheartsex, immigr, import, includ, incom, inform, insur, intern, issu, istandwithpp, justic, keep, know, leader, learn, legal, lgbtq, life, like, live, locat, lose, low, make, manag, mani, matern, matter, mea2018, mean, medic, million, miss, more, mother, must, nation, need, news, nogagrul, now, nurs, offer, often, option, orient, outsid, parent, parenthood, patient, penc, peopl, people, piec, plan, polici, politician, pp, pregnanc, pregnant, prevent, pride, program, protect, protectx, proud, provid, public, put, qualiti, question, rate, recogn, regardless, relationship, reproduct, resourc, respect, rewire_new, right, roe, rule, safe, save, screen, seen, serv, sex, sexual, shame, sign, singl, skill, someon, speak, staff, stand, statu, stay, std, step, stigma, still, stop, stoptheban, suprem, survivor, take, talk, teen, test, thing, think, thisishealthcar, thxbirthcontrol, titl, today, tran, transgend, treat, treatment, trump, trust, updat, via, violenc, visit, voic, wade, way, we, weareunstopp, well, whether, white, without, women, won, would, you, young, youth
4	**11**, 45, edt, pm
5	**ppfa**, 800, abl, address, affili, anyon, app, bad, beauti, best, bill, bless, call, cecilerichard, check, chelsea, clear, comfort, concern, condom, connect, consent, contact, cut, decid, definit, defund, defunds, difficult, direct, donat, done, dr, email, ever, experi, explain, feder, folk, frustrat, girl, glad, grate, gt, happen, healthcar, hear, here, hey, hi, hope, inspir, instead, kendal, kind, leadership, leav, match, media, member, method, metoo, might, movement, offic, onlin, org, out, partner, pay, person, phone, pill, place, pleas, ppact, pre, presid, protectourcar, read, real, refinery29, releas, role, saw, say, senat, servic, share, social, sorri, spot, standwithpp, support, tell, text, thank, they, tip, truli, trumpcar, tweet, understand, unstopp, us, use, video, vote, went, what, woman, wonder, work, worri, ye

Table 5.2: Words in each cluster for validation set of users using our clustering-based topic modeling algorithm. Part 1/3.

Latent Topic	Words in Cluster
6	**13**, children, moon, other
7	**goblu**, 000, 100, 12, 14, 15, 16, 17, 18, 19, 20, 2018, 2019, 21, 24, 25, 28, 30, 31, 40, _dbush11, accept, all, american, ann, announc, annual, arbor, armi, athlet, atlanta, avail, award, awesom, b1g, back, ball, beat, beatnd, beatosu, beatstat, behind, bell, ben, bestchanceu, better, big, blue, book, boy, bush, came, camp, can, car, career, catch, celebr, cfapeachbowl, charbonnet, charleswoodson, chase, chase_winovich, class, co, coach, coach_gatti, coachjim4um, colleg, complet, congrat, consecut, convers, dame, day, defens, detail, devin, digit, dljxxii, draft, dream, drive, earn, elect, enjoy, et, excit, fan, fbcoachdbrown, field, final, finish, first, five, footbal, former, four, fun, game, go, goal, good, got, greatest, ground, group, guid, guy, half, hall, happi, hard, he, head, higdon, highlight, hill, histori, home, honor, hour, hous, huge, icymi, illinoi, improv, in, indiana, info, insid, interact, iowa, it, job, join, jonjansen77, josh, karan, kick, last, law, lbg_nico7, lead, left, let, lewisjeweleri, light, line, list, littl, ll, long, look, loss, lot, love, man, march, maryland, meet, men, michigan, minut, moment, name, never, nfl, nflcombin, nfldraft, nflnetwork, nice, nico, noon, noth, notr, nsd, nsd19, number, offens, offici, ohio, old, one, opportun, pass, patriot, patterson, penn, per, perform, photo, pick, play, player, podcast, point, posit, practic, prepar, problu, punt, qb, quarter, rashanagari, readi, realli, recap, repres, return, run, rush, rutger, sack, saturday, school, score, season, second, see, select, senior, seven, shea, sheapatterson_1, show, sit, six, someth, special, sport, spring, st, stadium, stage, stori, student, sure, tackl, tbt, td, team, ten, tennesse, that, the, there, third, thisismichigan, thought, three, ticket, tie, tim, time, togeth, tom, tombradi, tomorrow, tonight, top, toss, total, touchdown, trench, trip, tune, two, ubuntublu, umichathlet, umichfootbal, umichfootball, univers, up, victori, vs, wait, wallpaperwednesday, want, watch, welcom, who, win, wino, wisconsin, wolverin, wow, yard, year, yesterday, zach
8	**vegan**, ad, add, almond, amaz, appl, asparagu, avocado, bake, banana, bar, base, basil, bbq, bean, birthday, bite, blueberri, bowl, bread, breakfast, broccoli, brown, browni, buffalo, bun, burger, burrito, butter, butternut, cake, caramel, carrot, cashew, cauliflow, chees, cheesecak, chia, chicken, chickpea, chili, chip, chocol, christma, cinnamon, coconut, comment, comments, cook, cooki, corn, cream, creami, crispi, cup, curri, date, delici, dessert, dinner, dip, dish, donut, dress, easi, eat, egg, energi, falafel, favorit, fill, food, found, fre, free, fresh, fri, friend, frost, garlic, ginger, gluten, green, grill, healthi, homemad, hot, hummu, idea, ingredi, jackfruit, kale, lasagna, lemon, lentil, lime, link, lunch, mac, made, mango, meal, meat, milk, mom, muffin, mushroom, no, noodl, nut, oat, oil, onion, orang, pan, pancak, pasta, pea, peanut, pepper, perfect, pesto, pie, pizza, post, pot, potato, prep, pretti, protein, pud, pumpkin, quick, quinoa, ramen, raspberri, raw, recip, red, rice, roast, roll, salad, salt, sandwich, sauc, sausag, scrambl, seed, seitan, sesam, simpl, slice, smoothi, snack, soup, soy, spaghetti, spanish, spice, spici, spinach, sprout, squash, stew, stir, strawberri, stuf, style, sugar, sun, super, sushi, sweet, taco, tast, tasti, tempeh, thai, toast, tofu, tomato, tri, turmer, turn, veget, veggi, version, walnut, whole, wing, wrap, zucchini
9	**earthequak**, accord, feel, felt, magnitude, report, san
10	**rain**, amount, arriv, august, beach, began, break, broke, cape, citi, deadli, death, drought, event, flash, flood, freez, heavi, imelda, kansa, lo, main, month, normal, outdoor, overnight, portion, pound, previou, receiv, rememb, river, soak, southeast, start, stormi, strike, sunday, texa, torrenti, town, unleash, whip, widespread
11	**week**, 80, 90, across, ahead, air, allow, anoth, arctic, autumn, averag, begin, bermuda, bright, british, build, burn, california, centr, challeng, chanc, chang, chilli, close, cloud, condit, continu, cool, cooler, could, daili, danger, deep, degre, doesn, downpour, dri, earli, earlier, end, europ, expand, extend, extrem, far, fire, focu, follow, frequent, georgia, goblue, grip, heat, high, histor, hold, holiday, howev, humid, increas, incred, india, kingdom, known, late, later, less, mark, may, meteor, middl, monsoon, much, natur, new, next, north, northeastern, numer, octob, ongo, past, pattern, peak, persist, pose, pressur, progress, push, rainfal, rais, reach, recent, record, remind, renew, replac, rise, risk, round, santa, septemb, set, settl, shift, shot, shower, signific, sinc, sky, soon, southward, southwest, southwestern, star, state, store, stream, stretch, summer, surg, target, temperatur, though, to, unit, unusu, usher, view, wake, warm, warmth, wave, weather, western, wet, wildfir, yet

Table 5.3: Words in each cluster for validation set of users using our clustering-based topic modeling algorithm. Part 2/3.

Latent Topic	Words in Cluster
12	**tropic**, accuweath, activ, addit, africa, alabama, alert, america, area, asia, atlant, bahama, basin, bear, becom, brace, brew, brought, caribbean, china, closer, come, cyclon, days, depress, develop, disturb, erick, featur, florida, form, grow, gulf, hawaii, humberto, imag, island, japan, karen, korea, landfal, louisiana, meteorologist, mexico, moistur, monitor, near, nestor, non, ocean, open, organ, pacif, philippin, puerto, result, rico, sea, short, southeastern, strength, taiwan, threat, took, trop, typhoon, water, weaken, world, zone
13	**snow**, accumul, cold, colorado, credit, denver, eastward, effect, fall, feet, front, great, halloween, highest, ice, inch, interior, lake, mix, montana, mountain, or, plummet, rang, realfeel, roadway, rocki, said, snowfal, snowstorm, swath, syndic, throughout, travel, unload, visibl, winter

Table 5.4: Words in each cluster for validation set of users using our clustering-based topic modeling algorithm. Part 3/3.

Latent Topic	Frequent Words
1	across make right made help goblu look watch along sunday morn time problu good first sure even 10 like fall guy educ incred _dbush11 nfl show catch nfldraft yet soak
2	rain week continu bring flood recip snow need forecast night thunderstorm could downpour shower northeast flash tonight round mexico period accuweath japan wave around last everyon warn remain drench senat
3	peopl part women here live use support power two learn impact famili coverag sex life countri becom protect like would sexual inform program video without produc thank doctor major top
4	health plan unit parenthood area new includ provid center wednesday month join ppfa reproduct state sever travel potenti commun far follow abort share spread challeng late sexual us trump listen
5	weekend one come track sever hurrican via fight cold it break middl southeast florida read southwest current cooler matter wake soon 12 goblu radar first lead hot missouri second state
6	may high record control birth set work temperatur anoth thursday mani week summer full citi accuweath day world proud player discuss final insid team open build hey amaz disturb import
7	goblu day michigan week much northern coast central east monday game threat move state big today west umichfootbal latest risk stand play everi chang report saturday lead touchdown winter find
8	weather system air expect heavi south tuesday local black vegan midwest fire well pacif ocean strengthen sea that rice week activ can more thing water low tri you warm tomato
9	take earli heat today plain year california best call reach hour mph form see test india bowl hail away monsoon larg danger super vote offici ahead welcom mile fan less
10	care ppfa access go thank chelsea develop patient keep let us want feel deserv depress servic love happi thanksgiv toward abort mean alreadi blue ask person hear istandwithpp great real
11	storm wind season state way atlant dorian thunderstorm end friday eastern northeast north near western strong gusti start make portion first damag region rainfal check northeastern midweek stori half hit
12	vegan next southern condit easi tofu safe later healthi potato chocol bean chickpea comment soup sweet sauc mushroom roast free rule pasta bake cake fri delici curri lentil southwestern chees
13	tropic get we salad know like tell experi close still possibl daili don begin line say philippin pleas sorri chanc never ll pressur degre back ball seen front start long

Table 5.5: 30 most frequent words for each latent topic using LDA on tweets of control users.

Appendix E: User Clusters

To calculate the overall probability distribution of topics for each user, we consider two methods for determining topic distribution for each user. The first (as shows in Table 3.4 in the main text) was to assign each tweet to a single topic. The second being to take the mean of the probability distributions for each tweet belonging to a user. Results using this metric are in Table 5.6. Most entries are fairly small, with one or two more strongly expressed topics per user. For Planned Parenthood, these are topics 3 and 10; topic 7 for Michigan football; topics 12 and 8 for vegan cooking; and topic 11 for AccuWeather.

	PPFA	UMichFootball	breakingweather	vegancook101
1	0.061	0.114	0.052	0.052
2	0.058	0.047	0.161	0.052
3	0.177	0.077	0.059	0.036
4	0.187	0.042	0.077	0.041
5	0.048	0.057	0.066	0.028
6	0.058	0.075	0.066	0.035
7	0.046	0.287	0.102	0.027
8	0.039	0.035	0.080	0.151
9	0.059	0.051	0.066	0.047
10	0.114	0.057	0.025	0.046
11	0.040	0.068	0.161	0.033
12	0.043	0.040	0.044	0.411
13	0.071	0.052	0.041	0.041

Table 5.6: Mean topics for tweets by users using LDA.

Appendix F: Closed Form Solution for Estimator When Using Sample Mean and OLS Prediction

In this section we solve for the $\hat{\mu}$ estimate in equation (4.1) when the sample mean or OLS is used as the prediction function $f(\cdot)$.

Sample Mean

In the case of the sample mean, let $\bar{y}_{\mathcal{S}} = \frac{1}{n}\sum_{i\in\mathcal{S}} y_i$ be the sample mean and $\bar{y}_{\mathcal{S}\setminus i} = \frac{1}{n-1}\sum_{j\neq i\in\mathcal{S}} y_j$ be the sample mean of $\mathcal{S}\setminus i$ when $i \in \mathcal{S}$. Then $f_i = \frac{N}{n}y_i - \frac{N-n}{n}\bar{y}_{\mathcal{S}\setminus i}$ and

$$\begin{aligned} g_i &= \frac{1}{n}\sum_{j\in\mathcal{S}} \bar{y}_{\mathcal{S}\setminus j} \\ &= \frac{1}{n(n-1)}\sum_{j\in\mathcal{S}}\sum_{k\neq j\in\mathcal{S}} y_k \\ &= \frac{1}{n(n-1)}(n-1)\sum_{j\in\mathcal{S}} y_j \\ &= \bar{y}_{\mathcal{S}} \end{aligned}$$

Then for the overall estimator we have

$$\begin{aligned}
\hat{\mu} &= \frac{1}{N}\sum_{j\in\mathcal{S}}\left(\frac{N}{n}y_i - \frac{N-n}{n}\bar{y}_{\mathcal{S}\setminus i}\right) + \frac{1}{N}\sum_{i\notin\mathcal{S}}\bar{y}_{\mathcal{S}} \\
&= \frac{1}{n}\sum_{i\in\mathcal{S}} y_i - \frac{N-n}{nN}\sum_{i\in\mathcal{S}}\bar{y}_{\mathcal{S}\setminus i} + \frac{N-n}{N}\bar{y}_{\mathcal{S}} \\
&= \frac{2N-n}{N}\bar{y}_{\mathcal{S}} - \frac{N-n}{nN(n-1)}\sum_{i\in\mathcal{S}}\sum_{j\neq i\in\mathcal{S}} y_j \\
&= \frac{2N-n}{N}\bar{y}_{\mathcal{S}} - \frac{N-n}{N}\bar{y}_{\mathcal{S}} \\
&= \bar{y}_{\mathcal{S}}
\end{aligned}$$

Thus, our estimator reduces to the sample mean when we use the mean as the prediction function.

OLS

In this section we solve for the $\hat{\mu}$ estimate when OLS is used as the prediction function $f(\cdot)$. In this case, as is shown below, it is sufficient to know the population means of the covariates; it is not necessarily to know the individual population $\boldsymbol{x}$ values, as was assumed earlier. Let $\boldsymbol{t}_x = \sum_{i=1}^{N}\boldsymbol{x}_i$ be the known population totals for the covariates, $\boldsymbol{t}_x^{\mathcal{S}} = \sum_{i\in\mathcal{S}}\boldsymbol{x}_i$ be the sample totals for the covariates, and $\boldsymbol{t}_x^{-\mathcal{S}} = \sum_{i\notin\mathcal{S}}\boldsymbol{x}_i = \boldsymbol{t}^x - \boldsymbol{t}_x^{\mathcal{S}}$ be the covariate totals for observations not in the sample.

For a given sample $\mathcal{S}$, let $\boldsymbol{x}_{\mathcal{S}}$ be the covariates for observations in the sample and let $\boldsymbol{y}_{\mathcal{S}}$ be the response values for the sample. Let $\hat{\beta}_{\mathcal{S}} = \left(\boldsymbol{x}_{\mathcal{S}}^T\boldsymbol{x}_{\mathcal{S}}\right)^{-1}\boldsymbol{x}_{\mathcal{S}}^T\boldsymbol{y}_{\mathcal{S}}$ be the OLS coefficient estimate obtained using the full sample. Let $\boldsymbol{H}_{\mathcal{S}} = \boldsymbol{x}_{\mathcal{S}}\left(\boldsymbol{x}_{\mathcal{S}}^T\boldsymbol{x}_{\mathcal{S}}\right)^{-1}\boldsymbol{x}_{\mathcal{S}}^T$ be the hat matrix for the sample and $\boldsymbol{e} = (I - \boldsymbol{H}_{\mathcal{S}})\boldsymbol{y}_{\mathcal{S}}$. Then when observation i is left out, the OLS coefficients will be

$$\hat{\beta}_{\mathcal{S}}^{(-i)} = \hat{\beta}_{\mathcal{S}} - \frac{(\boldsymbol{x}_{\mathcal{S}}^T\boldsymbol{x}_{\mathcal{S}})^{-1}\boldsymbol{x}_i^T e_i}{1 - H_{ii}}$$

It follows that $f_i = \boldsymbol{x}_i\hat{\beta}_{\mathcal{S}}^{(-i)}$. We have that $g_i = \frac{1}{n}\sum_{j\in\mathcal{S}}\boldsymbol{x}_i\hat{\beta}_{\mathcal{S}}^{(-j)}$. Then

$$\begin{aligned}
\sum_{i\notin\mathcal{S}} g_i &= \sum_{i\notin\mathcal{S}}\left(\frac{1}{n}\sum_{j\in\mathcal{S}}\boldsymbol{x}_i\hat{\beta}_{\mathcal{S}}^{(-j)}\right) \\
&= \sum_{j\in\mathcal{S}}\left(\frac{1}{n}\sum_{i\in\mathcal{S}}\boldsymbol{x}_i\hat{\beta}_{\mathcal{S}}^{(-j)}\right) \\
&= \sum_{j\in\mathcal{S}}\left(\frac{1}{n}\sum_{i\notin\mathcal{S}}\boldsymbol{x}_i\right)\hat{\beta}_{\mathcal{S}}^{(-j)} \\
&= \frac{1}{n}\sum_{j\in\mathcal{S}}\boldsymbol{t}_x^{-\mathcal{S}}\hat{\beta}_{\mathcal{S}}^{(-j)}
\end{aligned}$$

For the $\hat{\mu}$ estimator we have

$$\begin{aligned}\hat{\mu} &= \frac{1}{N}\sum_{i\in\mathcal{S}}\left(\frac{N}{n}y_i - \frac{N-n}{n}\boldsymbol{x}_i\hat{\beta}_x^{(-i)}\right) + \frac{1}{nN}\sum_{i\in\mathcal{S}}\boldsymbol{t}_x^{-\mathcal{S}}\hat{\beta}_{\mathcal{S}}^{(-i)} \\ &= \frac{1}{nN}\sum_{i\in\mathcal{S}}\{Ny_i + \left[\boldsymbol{t}_x^{-\mathcal{S}} - (N-n)\boldsymbol{x}_i\right]\hat{\beta}_{\mathcal{S}}^{(-i)}\}\end{aligned}$$

This estimate will be unbiased, regardless of how well the regression model fits the data.

Appendix G: Variance of Estimation Method

In this section we give justification for our variance estimate of $\hat{\mu}$ in equation (4.4) presented in Section 4.2.3. First we derive the variance of $\hat{\mu}$. We then provide an estimate for that variance.

Variance Calculation

We have that

$$\text{var}(\hat{\mu}) = \frac{1}{N^2}\sum_{i=1}^{N}\text{var}(\hat{y}_i) + \frac{1}{N^2}\sum_{i\neq j}\text{cov}(\hat{y}_i, \hat{y}_j) \tag{5.1}$$

We first solve for $\text{var}(\hat{y}_i)$ directly. Recall that $\hat{y}_i = y_i + (1-s_i)(h_i - y_i) + s_i\left(\frac{N-n}{n}\right)(y_i - h_i)$. Also recall that $\delta_i = h_i - e_i$, so $\mathbb{E}(\delta_i) = \mathbb{E}(\delta_i \mid s_i) = 0$. Let $r_i = y_i - e_i$, and note that r_i is a constant.

$$\begin{aligned}
\text{var}(\hat{y}_i) &= \text{var}\left[\delta_i\left(1 - \frac{N}{n}s_i\right) + \frac{N}{n}s_i r_i\right] \\
&= \text{var}\left\{\mathbb{E}\left[\delta_i\left(1 - \frac{N}{n}s_i\right) + \frac{N}{n}s_i r_i \middle| s_i\right]\right\} + \mathbb{E}\left\{\text{var}\left[\delta_i\left(1 - \frac{N}{n}s_i\right) + \frac{N}{n}s_i r_i \middle| s_i\right]\right\} \\
&= \text{var}\left[\frac{N}{n}s_i r_i\right] + \mathbb{E}\left[\left(1 - \frac{N}{n}s_i\right)^2 \text{var}(\delta_i \mid s_i)\right] \\
&= \frac{N^2}{n^2}r_i^2\text{var}(s_i) + P(s_i = 1)\left(\frac{N-n}{n}\right)^2\text{var}(f_i \mid s_i = 1) + P(s_i = 0)\text{var}(g_i \mid s_i = 0) \\
&= \frac{N-n}{n}r_i^2 + \frac{(N-n)^2}{nN}\text{var}(f_i \mid s_i = 1) + \frac{N-n}{N}\text{var}(g_i \mid s_i = 0)
\end{aligned}$$

Plugging this expression into variance equation (5.1) gives

$$\begin{aligned}
\operatorname{var}(\hat{\mu}) &= \frac{1}{N^2}\sum_{i=1}^{N}\left[\frac{N-n}{n}r_i^2 + \frac{(N-n)^2}{nN}\operatorname{var}(f_i \mid s_i = 1) + \frac{N-n}{N}\operatorname{var}(g_i \mid s_i = 0)\right] + \frac{1}{N^2}\sum_{i\neq j}\operatorname{cov}(\hat{y}_i, \hat{y}_j) \\
&= \frac{N-n}{nN}\left(\frac{1}{N}\sum_{i=1}^{N} r_i^2 + \frac{1}{N}\sum_{i=1}^{N}\operatorname{var}(f_i \mid s_i = 1)\right) + \frac{N-n}{N^3}\sum_{i=1}^{N}[\operatorname{var}(g_i \mid s_i = 0) - \operatorname{var}(f_i \mid s_i = 1)] \\
&\quad + \frac{1}{N^2}\sum_{i\neq j}\operatorname{cov}(\hat{y}_i, \hat{y}_j) \qquad (5.2)
\end{aligned}$$

We now calculate $\operatorname{cov}(\hat{y}_i, \hat{y}_j)$.

$$\begin{aligned}
\frac{1}{N^2}\sum_{i\neq j}\operatorname{cov}(\hat{y}_i,\hat{y}_j) &= \frac{1}{N^2}\sum_{i\neq j}\mathbb{E}(\hat{y}_i\hat{y}_j) - \frac{1}{N^2}\sum_{i\neq j}\mathbb{E}(\hat{y}_i)\mathbb{E}(\hat{y}_j) \\
&= \frac{1}{N^2}\sum_{i\neq j}\mathbb{E}(\hat{y}_i\hat{y}_j \mid s_i=1, s_j=1)P(s_i=1,s_j=1) \\
&\quad + \frac{2}{N^2}\sum_{i\neq j}\mathbb{E}(\hat{y}_i\hat{y}_j \mid s_i=1, s_j=0)P(s_i=1,s_j=0) \\
&\quad + \frac{1}{N^2}\sum_{i\neq j}\mathbb{E}(\hat{y}_i\hat{y}_j \mid s_i=0, s_j=0)P(s_i=0,s_j=0) - \frac{1}{N^2}\sum_{i\neq j} y_i y_j \\
&= \frac{n(n-1)}{N^3(N-1)}\sum_{i\neq j}\mathbb{E}\left[\left(\frac{N}{n}y_i - \frac{N-n}{n}f_i\right)\left(\frac{N}{n}y_j - \frac{N-n}{n}f_j\right)\middle| s_i=1, s_j=1\right] \\
&\quad + \frac{2n(N-n)}{N^3(N-1)}\sum_{i\neq j}\mathbb{E}\left[\left(\frac{N}{n}y_i - \frac{N-n}{n}f_i\right)g_i\middle| s_i=1, s_j=0\right] \\
&\quad + \frac{(N-n)(N-n-1)}{N^3(N-1)}\sum_{i\neq j}\mathbb{E}(g_ig_j \mid s_i=0,s_j=0) - \frac{1}{N^2}\sum_{i\neq j} y_iy_j \\
&= \frac{n(n-1)}{N^3(N-1)}\sum_{i\neq j}\mathbb{E}\left(\frac{N^2}{n^2}y_iy_j - \frac{2N(N-n)}{n^2}y_if_j + \frac{(N-n)^2}{n^2}f_if_j\middle| s_i=1,s_j=1\right) \\
&\quad + \frac{2n(N-n)}{N^3(N-1)}\sum_{i\neq j}\mathbb{E}\left(\frac{N}{n}y_ig_j - \frac{N-n}{n}f_ig_j\middle| s_i=1,s_j=0\right) \\
&\quad + \frac{(N-n)(N-n-1)}{N^3(N-1)}\sum_{i\neq j}\mathbb{E}(g_ig_j \mid s_i=0,s_j=0) - \frac{1}{N^2}\sum_{i\neq j} y_iy_j \\
&= -\frac{N-n}{nN^2(N-1)}\sum_{i\neq j} y_iy_j - \frac{2(N-n)(n-1)}{nN^2(N-1)}\sum_{i\neq j} y_i\mathbb{E}(f_j \mid s_i=1,s_j=1) \\
&\quad + \frac{(N-n)^2(n-1)}{nN^3(N-1)}\sum_{i\neq j}\mathbb{E}(f_if_j \mid s_i=1,s_j=1) \\
&\quad + \frac{2(N-n)}{N^2(N-1)}\sum_{i\neq j} y_i\mathbb{E}(g_j \mid s_i=1,s_j=0) \\
&\quad - \frac{2(N-n)^2}{N^3(N-1)}\sum_{i\neq j}\mathbb{E}(f_ig_j \mid s_i=1,s_j=0) \\
&\quad + \frac{(N-n)(N-n-1)}{N^3(N-1)}\sum_{i\neq j}\mathbb{E}(g_ig_j \mid s_i=0,s_j=0)
\end{aligned}$$

For the overall variance of $\hat{\mu}$ we have:

$$\begin{aligned}
\operatorname{var}(\hat{\mu}) = \frac{N-n}{nN}\left(\frac{1}{N}\sum_{i=1}^{N} r_i^2 + \frac{1}{N}\sum_{i=1}^{N}\operatorname{var}(f_i \mid s_i = 1)\right) + \frac{N-n}{N^3}\sum_{i=1}^{N}[\operatorname{var}(g_i \mid s_i = 0) - \operatorname{var}(f_i \mid s_i = 1)] \\
- \frac{N-n}{nN^2(N-1)}\sum_{i\neq j} y_i y_j - \frac{2(N-n)(n-1)}{nN^2(N-1)}\sum_{i\neq j} y_i \mathbb{E}(f_j \mid s_i = 1, s_j = 1) \\
+ \frac{(N-n)^2(n-1)}{nN^3(N-1)}\sum_{i\neq j}\mathbb{E}(f_i f_j \mid s_i = 1, s_j = 1) + \frac{2(N-n)}{N^2(N-1)}\sum_{i\neq j} y_i \mathbb{E}(g_j \mid s_i = 1, s_j = 0) \\
- \frac{2(N-n)^2}{N^3(N-1)}\sum_{i\neq j}\mathbb{E}(f_i g_j \mid s_i = 1, s_j = 0) \\
+ \frac{(N-n)(N-n-1)}{N^3(N-1)}\sum_{i\neq j}\mathbb{E}(g_i g_j \mid s_i = 0, s_j = 0)
\end{aligned}$$

Estimate of Variance

Recall that our variance estimate for $\hat{\mu}$ in equation (4.4) is

$$\widehat{\operatorname{var}}(\hat{\mu}) = \frac{N-n}{nN}\left[\frac{1}{n}\sum_{i\in\mathcal{S}}(y_i - f_i)^2 - \frac{1}{n(n-1)}\sum_{i\neq j\in\mathcal{S}}(y_i - f_i)(y_j - f_j)\right] \tag{5.3}$$

We motive this estimate of the variance below. We do this in three steps:

(a) $\frac{1}{n}\sum_{i\in\mathcal{S}}(y_i - f_i)^2$ in (5.3) is an unbiased estimate of $\left(\frac{1}{N}\sum_{i=1}^{N} r_i^2 + \frac{1}{N}\sum_{i=1}^{N}\operatorname{var}(f_i \mid s_i = 1)\right)$ in (5.2)

(b) $\frac{N-n}{N^3}\sum_{i=1}^{N}[\operatorname{var}(g_i \mid s_i = 0) - \operatorname{var}(f_i \mid s_i = 1)]$ in (5.2) is nonpositive and small

(c) $-\frac{N-n}{n^2N(n-1)}\sum_{i\neq j\in\mathcal{S}}(y_i - f_i)(y_j - f_j)$ in (5.3) is an estimate of $\frac{1}{N^2}\sum_{i\neq j}\operatorname{cov}(\hat{y}_i, \hat{y}_j)$ in (5.2)

We show each of these below.

(a)

Below we show that $\frac{1}{n}\sum_{i\in\mathcal{S}}(y_i - f_i)^2$ is an unbiased estimate of $\left(\frac{1}{N}\sum_{i=1}^{N} r_i^2 + \frac{1}{N}\sum_{i=1}^{N}\text{var}(f_i \mid s_i = 1)\right)$:

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{n}\sum_{i\in\mathcal{S}}(y_i - f_i)^2\right] &= \mathbb{E}\left[\frac{1}{n}\sum_{i\in\mathcal{S}}(y_i - e_i - f_i + e_i)^2\right] \\
&= \frac{1}{n}\mathbb{E}\left[\sum_{i=1}^{N} s_i(r_i - \delta_i)^2\right] \\
&= \frac{1}{n}\sum_{i=1}^{N}\mathbb{E}(s_i r_i^2 - 2s_i r_i\delta_i + s_i\delta_i) \\
&= \frac{1}{n}\sum_{i=1}^{N} r_i^2\mathbb{E}(s_i) - \frac{2}{n}\sum_{i=1}^{N} r_i\mathbb{E}(s_i\delta_i) + \frac{1}{n}\sum_{i=1}^{N}\mathbb{E}\left(s_i\delta_i^2\right) \\
&= \frac{1}{N}\sum_{i=1}^{N} r_i^2 + \frac{1}{N}\sum_{i=1}^{N}\text{var}(f_i \mid s_i = 1)
\end{aligned}$$

(b)

Next we show that the second term in equation (5.2), $\frac{N-n}{N^3}\sum_{i=1}^{N}[\text{var}(g_i \mid s_i = 0) - \text{var}(f_i \mid s_i = 1)]$, is non-positive, so ignoring this term will only make the variance estimate more conservative. We first define new variables. Let d be a random indicator vector such that one entry is 1 and the remaining entries are 0 and $s \cdot d = 1$. Let $T \equiv \{i : s_i - d_i = 1\}$ and $\phi \equiv f(\cdot; x_T, y_T)$ and $\phi_i \equiv f(x_i; x_T, y_T)$. That is, ϕ_i is the estimated y response value for observation i using a prediction function trained on set of observations T, where d indicates the observation that is dropped from the sample to create the training set.

Then we have that $\phi_i \mid s_i = 0$ and $f_i \mid s_i = 1$ are equal in distribution. These two functions predict y_i using a function trained a random sample of size $(n-1)$ that does not include observation i. It follows that $\text{var}(f_i \mid s_i = 1) = \text{var}(\phi_i \mid s_i = 0)$.

We also have that $g_i \mid s_i = 0$ is equal to $\mathbb{E}(\phi_i \mid \mathcal{S}) \mid s_i = 0$. It follows that $\text{var}(g_i \mid s_i = 0) = \text{var}[\mathbb{E}(\phi_i \mid \mathcal{S}) \mid s_i = 0)]$.

Therefore, to show that $\text{var}(g_i \mid s_i = 0) - \text{var}(f_i \mid s_i = 1) \leq 0$, it is sufficient to show that

$$\text{var}[\mathbb{E}(\phi_i \mid \mathcal{S}) \mid s_i = 0)] \leq \text{var}(\phi_i \mid s_i = 0)$$

We have that

$$\text{var}(\phi_i \mid s_i = 0) = \mathbb{E}[\text{var}(\phi_i \mid \mathcal{S}) \mid s_i = 0] + \text{var}[\mathbb{E}(\phi_i \mid \mathcal{S}) \mid s_i = 0]$$

and

$$0 \leq \mathbb{E}[\text{Var}(\phi_i \mid s) \mid s_i = 0]$$

It follows that $\text{var}(g_i \mid s_i = 0) - \text{var}(f_i \mid s_i = 1) \leq 0$.

It follows that the second term in equation (5.2), $\frac{N-n}{N^3}\sum_{i=1}^{N}[\text{var}(g_i \mid s_i = 0) - \text{var}(f_i \mid s_i = 1)]$, is less

than 0. Note also the factor of $\frac{N-n}{N^3}$; for even moderate population sizes, this term will be small. Ignoring this term will give a slightly conservative estimate of the variance.

(c)

It remains to show that $-\frac{N-n}{n^2N(n-1)}\sum_{i\neq j\in\mathcal{S}}(y_i-f_i)(y_j-f_j)$ is an estimate of $\frac{1}{N^2}\sum_{i\neq j}\text{cov}(\hat{y}_i,\hat{y}_j)$. First consider the expectation of the first term.

$$\begin{aligned}
&\mathbb{E}\left[-\frac{N-n}{n^2N(n-1)}\sum_{i\neq j\in\mathcal{S}}(y_i-f_i)(y_j-f_j)\right] = -\frac{N-n}{n^2N(n-1)}\mathbb{E}\left[\sum_{i\neq j}s_is_j(y_i-f_i)(y_j-f_j)\right]\\
&= -\frac{N-n}{n^2N(n-1)}\sum_{i\neq j}\mathbb{E}\left[(y_i-f_i)(y_j-f_j)\mid s_i=1,s_j=1)\right]P(s_i=1,s_j=1)\\
&= -\frac{N-n}{nN^2(N-1)}\sum_{i\neq j}\mathbb{E}(y_iy_j-y_if_j-y_jf_i+f_if_j\mid s_i=1,s_j=1)\\
&= -\frac{N-n}{nN^2(N-1)}\sum_{i\neq j}y_iy_j+\frac{2(N-n)}{nN^2(N-1)}\sum_{i\neq j}y_i\mathbb{E}(f_j\mid s_i=1,s_j=1)\\
&\quad -\frac{N-n}{nN^2(N-1)}\sum_{i\neq j}\mathbb{E}(f_if_j\mid s_i=1,s_j=1)
\end{aligned}$$

The true sum of the covariances between the y_is, as we showed earlier, is equal to:

$$\begin{aligned}
\frac{1}{N^2}\sum_{i\neq j}\text{cov}(\hat{y}_i,\hat{y}_j) &= -\frac{N-n}{nN^2(N-1)}\sum_{i\neq j}y_iy_j-\frac{2(N-n)(n-1)}{nN^2(N-1)}\sum_{i\neq j}y_i\mathbb{E}(f_j\mid s_i=1,s_j=1)\\
&\quad+\frac{(N-n)^2(n-1)}{nN^3(N-1)}\sum_{i\neq j}\mathbb{E}(f_if_j\mid s_i=1,s_j=1)\\
&\quad+\frac{2(N-n)}{N^2(N-1)}\sum_{i\neq j}y_i\mathbb{E}(g_j\mid s_i=1,s_j=0)\\
&\quad-\frac{2(N-n)^2}{N^3(N-1)}\sum_{i\neq j}\mathbb{E}(f_ig_j\mid s_i=1,s_j=0)\\
&\quad+\frac{(N-n)(N-n-1)}{N^3(N-1)}\sum_{i\neq j}\mathbb{E}(g_ig_j\mid s_i=0,s_j=0)
\end{aligned}$$

These two term are not exactly the same, but we argue that they are similar. Consider the difference between

the two terms.

$$\mathbb{E}\left[-\frac{N-n}{n^2N(n-1)}\sum_{i\neq j\in\mathcal{S}}(y_i-f_i)(y_j-f_j)\right]-\frac{1}{N^2}\sum_{i\neq j}\text{cov}(\hat{y}_i,\hat{y}_j)$$
$$=\frac{N-n}{N^2(N-1)}\Bigg\{\frac{N-n}{N}\sum_{i\neq j}\left[\mathbb{E}(f_if_j\mid s_i=1,s_j=1)+\mathbb{E}(g_ig_j|s_i=0,s_j=0)-2\mathbb{E}(f_ig_j\mid s_i=1,s_j=0)\right]$$
$$+2\sum_{i\neq j}y_i\left[\mathbb{E}(g_j\mid s_i=1,s_j=0)-\mathbb{E}(f_j\mid s_i=1,s_j=1)\right]$$
$$+\frac{1}{N}\sum_{i\neq j}\left[\mathbb{E}(f_if_j\mid s_i=1,s_j=1)-\mathbb{E}(g_ig_j\mid s_i=0,s_j=0)\right]\Bigg\} \tag{5.4}$$

Using the following assumptions we show that each of the three sums in equation (5.4) converges to zero faster than $\frac{1}{n}$:

(A1): Assume that $\phi_i=\alpha_i^{(0)}+\sum_{j\neq i}\frac{\alpha_{ij}^{(1)}}{n}s_j+o\left(\frac{1}{n}\right)$. We assume that each of the α values are bounded: for all n and for all i,j, $|\alpha_i^{(0)}|\leq\alpha_{max}$ and $|\alpha_{ij}^{(1)}|\leq\alpha_{max}$. Then $\phi_i\mid(s_j=1)-\phi_i\mid(s_j=0)=\frac{\alpha_{ij}^{(1)}}{n}+o\left(\frac{1}{n}\right)$. In other words, we assume the function is asymptotically linear with bounded coefficients.

Note in particular that as a direct consequence of assumption (A1), $\mathbb{E}(f_i\mid s_i=1,s_j)$ and $\mathbb{E}(g_i\mid s_i=0,s_j)$ vary from $e_i=\mathbb{E}(f_i\mid s_i=1)=\mathbb{E}(g_i\mid s_i=0)$ by $\frac{c_1}{n^q}$, where $c_1>0$ and $q>0$, for all $i\neq j$. That is, for all $i\neq j$ we have that

$$e_i-\frac{c_1}{n}+o\left(\frac{1}{n}\right)\leq\mathbb{E}(f_i\mid s_i=1,s_j=1)\leq e_i+\frac{c_1}{n}+o\left(\frac{1}{n}\right)$$
$$e_i-\frac{c_1}{n}+o\left(\frac{1}{n}\right)\leq\mathbb{E}(f_i\mid s_i=1,s_j=0)\leq e_i+\frac{c_1}{n}+o\left(\frac{1}{n}\right)$$
$$e_i-\frac{c_1}{n}+o\left(\frac{1}{n}\right)\leq\mathbb{E}(g_i\mid s_i=0,s_j=1)\leq e_i+\frac{c_1}{n}+o\left(\frac{1}{n}\right)$$
$$e_i-\frac{c_1}{n}+o\left(\frac{1}{n}\right)\leq\mathbb{E}(g_i\mid s_i=0,s_j=0)\leq e_i+\frac{c_1}{n}+o\left(\frac{1}{n}\right)$$

This means that the inclusion or exclusion of observation j in the sample has minimal effect on the expected value of f_i and g_i, with the effect decreasing with $\frac{1}{n}$.

(A2): There exists some e_{max} such that $|e_i|\leq e_{max}$ for all i and for all n. We also assume that y_i is bounded for all i and for all n: $y_i\leq y_{max}\ \forall i,n$.

(A3): There exists some σ^2_{max} such that $\text{var}(f_i\mid s_i=1,s_j)\leq\sigma^2_{max}$ and $\text{var}(g_i\mid s_i=0,s_j)\leq\sigma^2_{max}$ for all i,j and let $\sigma^2_{max}\leq\frac{c_2}{n^p}$ for some $c_2>0$ and $p>0$.

There are three sums in equation (5.4) that we consider individually. Starting with the second sum, we

have that by assumption (A1)

$$\begin{aligned}\mathbb{E}(g_j \mid s_i = 1, s_j = 0) - \mathbb{E}(f_j \mid s_i = 1, s_j = 1) &= \frac{1}{n}\mathbb{E}(f_j \mid s_j = 1, s_i = 0) + \frac{n-1}{n}\mathbb{E}(f_j \mid s_i = 1, s_j = 1) \\ &\quad - \mathbb{E}(f_i \mid s_i = 1, s_j = 1) \\ &= \frac{1}{n}\left[\mathbb{E}(f_j | s_j = 1, s_i = 0) - \mathbb{E}(f_j | s_i = 1, s_j = 1)\right] \\ &= \frac{\alpha_{ji}^{(1)}}{n^2} + o\left(\frac{1}{n^2}\right)\end{aligned}$$

Therefore, for the second sum we have

$$\frac{2(N-n)}{N^2(N-1)}\sum_{i\neq j} y_i \left[\mathbb{E}(g_j \mid s_i = 1, s_j = 0) - \mathbb{E}(f_j \mid s_i = 1, s_j = 1)\right] = \frac{2(N-n)}{N^2(N-1)}\sum_{i\neq j} y_i \left[\frac{\alpha_{ji}^{(1)}}{n^2} + o\left(\frac{1}{n^2}\right)\right]$$

which is on the order of $\frac{1}{n^2}$.

Next consider the last sum in Equation (5.4). Using assumptions (A1), (A2), and (A3), we have that

$$\begin{aligned}&\frac{(N-n)}{N^3(N-1)}\sum_{i\neq j}[\mathbb{E}(f_i f_j \mid s_i = 1, s_j = 1) - \mathbb{E}(g_i g_j \mid s_i = 0, s_j = 0)] \\ &\leq \frac{(N-n)}{N^3(N-1)}\sum_{i\neq j}[|\mathrm{cov}(f_i, f_j \mid s_i = 1, s_j = 1)| + |\mathrm{cov}(g_i, g_j \mid s_i = 0, s_j = 0)| \\ &\quad + \mathbb{E}(f_i \mid s_i = 1, s_j = 1)\mathbb{E}(f_j \mid s_i = 1, s_j = 1) - \mathbb{E}(g_i \mid s_i = 0, s_j = 0)\mathbb{E}(g_j \mid s_i = 0, s_j = 0)] \\ &\leq \frac{(N-n)}{N^3(N-1)}\sum_{i\neq j}\left[\frac{2c_2}{n^p} + \frac{4c_1 e_{max}}{n} + o\left(\frac{1}{n}\right)\right] \\ &= \frac{2(N-n)}{N^2 n^p}c_2 + \frac{4(N-n)}{N^2 n}e_{max} + \frac{N-n}{N^2}o\left(\frac{1}{n}\right) \\ &\leq \frac{2(N-n)c_2}{N n^{p+1}} + \frac{4(N-n)e_{max}}{N n^2} + o\left(\frac{1}{n^2}\right)\end{aligned}$$

This term approaches 0 faster than $\frac{1}{n}$.

We have not yet been able to show that the first term converges to zero faster than $\frac{1}{n}$ but conjecture it does and continue to pursue this in future work.

Appendix H: Additional Tables from Estimation Method

Simulations

The standard error of the bias can be calculated as the simulated estimate of the true standard error divided by 100 (square root of number of simulations). To test whether or not the observed bias is due to chance, we calculate the t-test statistic of the bias as $t = \frac{Bias}{True\ SE/100}$. We consider the bias significantly different from 0 if $|t| > 2$, corresponding to a significance level of about 0.05. Results from the linear population simulation, with simulated bias and t-test statistics, are in Table 5.7. In this simulation, the only observed bias that is significantly different than 0 is the sample mean with $N = 50$ and $n = 10$, with an associated t-test statistic of -2.1957. Since the sample mean is known to be unbiased for all N and n, this was due to chance.

We also consider the setting where there is no relationship between the covariates and the response. We let the population size be 100,000, the covariates for each observation be a 20-dimensional vector , $X \sim N(0, I)$, and $Y \sim N(0, 1)$. We take 1000 random samples and estimate the population mean using the sample mean, random forest adjustment, and our method with random forest prediction. Results can be seen in Table 5.8. In the comparison with the sample mean, neither the random forest adjustment nor our method with random forest prediction had much of an adverse effect on the standard error nor bias.

College Tuition

In Tables 5.10 and 5.11 we give the simulated estimate of the true standard error and estimated bias of all five methods for $n = 100, 200, \ldots, 1000$. For the nonprobability sample, the bias was significantly different from 0 for all methods at all sample sizes. To demonstrate the improvement that more sophisticated machine learning methods can provide, we give the ratio of simulated true standard error using out method using OLS compared to random forest in Table 5.9. Our method with random forest had lower standard error throughout compared to our method with OLS. As the sample size, the standard error using random forest improved fast then with OLS. This demonstrates the advantage that machine learning models can have over less sophisticated modeling techniques.

		True SE	Est. SE	Bias	t
	$n = 10$				
$N = 50$	Sample Mean	0.5093	0.5135	-0.0112	-2.1957
	OLS Adjustment	0.2331	0.2220	-0.0015	-0.6469
	Our Method with OLS	0.2424	0.2687	0.0008	0.3479
	$n = 10$				
	Sample Mean	0.7008	0.7017	-0.0023	-0.3265
	OLS Adjustment	0.3117	0.2925	0.0047	1.4961
$N = 500$	Our Method with OLS	0.3177	0.3502	-0.0009	-0.2795
	$n = 100$				
	Sample Mean	0.1984	0.2006	0.0007	0.3413
	OLS Adjustment	0.0842	0.0839	0.0003	0.3183
	Our Method with OLS	0.0843	0.0852	-0.0004	-0.4591
	$n = 10$				
	Sample Mean	0.7118	0.7141	-0.0041	-0.5765
	OLS Adjustment	0.3381	0.3131	0.0029	0.8583
	Our Method with OLS	0.3428	0.3744	0.0036	1.0413
	$n = 100$				
$N = 10000$	Sample Mean	0.2231	0.2246	0.0037	1.6423
	OLS Adjustment	0.0989	0.0987	0.0019	1.8814
	Our Method with OLS	0.0988	0.1022	0.0019	1.9097
	$n = 1000$				
	Sample Mean	0.0667	0.0667	-0.0008	-1.1258
	OLS Adjustment	0.0300	0.0297	-0.0002	-0.5445
	Our Method with OLS	0.0300	0.0298	-0.0002	-0.5416

Table 5.7: Bias, simulated standard error, t-statistics, and estimated standard error for our method using OLS prediction, OLS adjustment, and sample mean for a population with a linear relationship between x and y.

		True SE	Est. SE	Bias	
	Sample Mean	0.2323	0.2241	0.0069	
$n = 25$	Random Forest Adjustment	0.2355	0.0970	0.0078	
	Our Method with Random Forest	0.2362	0.2395	0.0076	
	Sample Mean	0.0997	0.1004	0.0073	(*)
$n = 50$	Random Forest Adjustment	0.1022	0.0406	0.0076	(*)
	Our Method with Random Forest	0.1011	0.1030	0.0067	(*)
	Sample Mean	0.0322	0.0315	0.0004	
$n = 100$	Random Forest Adjustment	0.0324	0.0125	0.0006	
	Our Method with Random Forest	0.0323	0.0318	0.0003	

Table 5.8: Simulation estimate of the true standard error, estimated standard error, and estimated bias using the sample mean, OLS adjustment, and our method with OLS for a population with no relationship between X and Y.

n	$\frac{SE(Ours:RF)}{SE(Ours:OLS)}$
100	0.96
200	0.90
300	0.87
400	0.86
500	0.86
600	0.85
700	0.84
800	0.83
900	0.83
1000	0.84

Table 5.9: Decrease in simulated standard error when using our method with random forest as opposed to OLS.

	Ours: OLS		OLS Adj.		Ours: RF		RF Adj.		Sample Mean	
n	Sim. SE	Bias	Sim. SE	Bias	Sim. SE	Bias	Sim. SE	Bias	Sim. SE	Bias
100	641.92	-0.23	643.06	-2.57	613.04	7.29	615.49	-41.46	1457.21	12.78
200	421.75	-1.64	421.98	-2.86	383.24	-2.82	379.07	-44.61	992.38	-9.89
300	330.23	-2.82	330.40	-3.59	289.90	-1.62	287.00	-33.57	771.34	1.18
400	268.85	-2.64	268.93	-3.23	233.44	-2.77	231.18	-28.28	622.82	-8.43
500	226.40	-3.45	226.45	-3.86	191.92	-2.11	190.38	-21.92	527.22	-4.32
600	192.01	-0.10	192.04	-0.40	162.82	-0.74	161.74	-16.68	446.87	-1.21
700	163.02	-2.14	163.04	-2.36	137.92	-0.91	137.13	-13.25	382.26	-2.95
800	139.82	0.13	139.84	-0.02	116.95	0.80	116.38	-8.71	323.41	3.70
900	115.35	-2.71	115.36	-2.82	96.89	-2.09	96.48	-8.96	270.14	-6.26
1000	93.74	0.07	93.75	0.00	78.42	-0.42	78.16	-5.18	218.32	-1.48

Table 5.10: Simulated bias and simulated standard error for mean college tuition for simple random samples.

	Ours: OLS		OLS Adj.		Ours: RF		RF Adj.		Sample Mean	
n	Sim. SE	Bias	Sim. SE	Bias	Sim. SE	Bias	Sim. SE	Bias	Sim. SE	Bias
100	644.76	-417.52	645.29	-421.88	591.40	-716.18	591.38	-812.23	1480.35	-2897.45
200	422.15	-401.16	422.45	-403.33	373.66	-561.15	369.43	-620.99	997.26	-2742.66
300	324.11	-383.01	324.23	-384.33	278.61	-482.55	275.67	-524.69	773.69	-2596.72
400	269.51	-367.31	269.57	-368.24	225.75	-435.34	224.29	-465.61	623.56	-2454.67
500	226.20	-335.52	226.23	-336.14	187.05	-386.31	185.75	-409.09	529.01	-2293.47
600	192.74	-316.27	192.77	-316.70	157.92	-352.08	156.77	-368.58	447.20	-2116.24
700	162.64	-285.59	162.65	-285.89	131.13	-312.02	130.40	-324.62	371.41	-1916.76
800	140.24	-252.13	140.25	-252.34	112.68	-271.89	112.24	-281.13	318.37	-1708.96
900	116.16	-215.90	116.16	-216.04	92.67	-230.47	92.31	-237.22	261.61	-1463.60
1000	93.24	-174.62	93.24	-174.71	74.06	-184.84	73.84	-189.44	206.00	-1192.33

Table 5.11: Simulated bias and simulated standard error for estimating mean college tuition for nonprobability samples.