Journal of Artificial Intelligence Research 63 (2018) 391-419 Submitted 07/18; published 11/18

Robust Text Classification under Confounding Shift

Virgile Landeiro VLANDEIR@HAWK.IIT.EDU
Aron Culotta ACULOTTA@IIT.EDU
Department of Computer Science
Illinois Institute of Technology
Chicago, IL 60616

Abstract

As statistical classifiers become integrated into real-world applications, it is important to consider not only their accuracy but also their robustness to changes in the data distribution. Although identifying and controlling for confounding variables Z – correlated with both the input X of a classifier and its output Y – has been assiduously studied in empirical social science, it is often neglected in text classification. This can be understood by the fact that, if we assume that the impact of confounding variables does not change between the time we fit a model and the time we use it, then prediction accuracy should only be slightly affected. We show in this paper that this assumption often does not hold and that when the influence of a confounding variable changes from training time to prediction time (i.e. under confounding shift), the classifier accuracy can degrade rapidly. We use Pearl's back-door adjustment as a predictive framework to develop a model robust to confounding shift under the condition that Z is observed at training time. Our approach does not make any causal conclusions but by experimenting on 6 datasets, we show that our approach is able to outperform baselines 1) in controlled cases where confounding shift is manually injected between fitting time and prediction time 2) in natural experiments where confounding shift appears either abruptly or gradually 3) in cases where there is one or multiple confounders. Finally, we discuss multiple issues we encountered during this research such as the effect of noise in the observation of Z and the importance of only controlling for confounding variables.

1. Introduction

A common assumption of machine learning algorithms in general, and text classification in particular is that the training data and testing data are drawn from the same probability distribution. However, this assumption is often violated in practice, a scenario termed *dataset shift* (Quionero-Candela, Sugiyama, Schwaighofer, & Lawrence, 2009). That is, for feature vector x and class label y, the training data are drawn from one distribution $P_{train}(X, Y)$, while the testing data are drawn from a different distribution $P_{test}(X, Y)$. Prior research has considered several types of dataset shift, including covariate shift, where $P_{train}(X) \neq P_{test}(X)$ (Sugiyama, Krauledat, & Müller, 2007); prior probability shift, where $P_{train}(Y) \neq P_{test}(Y)$ (Webb & Ting, 2005); and concept drift, where $P_{train}(Y|X) \neq P_{test}(Y|X)$ (Widmer & Kubat, 1996).

In this paper, we investigate a type of dataset shift we call *confounding shift*. Such a setting exists when two conditions are met: (a) there exists a confounding variable Z that

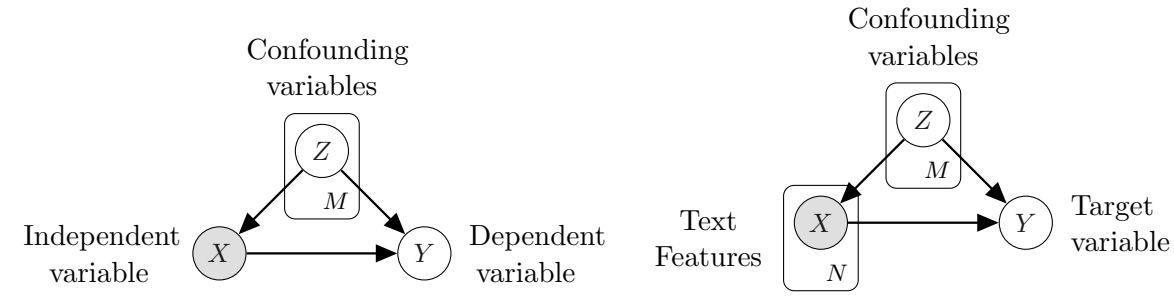
influences both X and Y through distributions $P(X|Z)$ and $P(Y|Z)$, and (b) there is a shift from $P_{train}(Y|Z)$ to $P_{test}(Y|Z)$.

Our motivation for this problem setting stems from emerging applications in computational social science (Lazer et al., 2009; Hopkins & King, 2010), in which text classifiers $P(Y|X)$ are fit to data where x represents the writings of one person and y represents a category label for that person. For example, De Choudhury et al. (2016) predict mental health status of Reddit users from their posts, and Schwartz et al. (2013) predict the personality of Facebook users from their comments. In such settings, the investigator often has *a priori* knowledge of certain factors Z that are related to both X and Y; ignoring Z can therefore introduce omitted variable bias (Lee, 1982) into the classifier $P(Y|X)$. For example, gender (Z) may correlate with both the words one uses (X) and one's mental health status (Y). There are many reasons why $P_{train}(Y|Z)$ may not equal $P_{test}(Y|Z)$. If the data arrive chronologically, there may be true underlying changes in the relationship between Y and Z — e.g., a topic may indicate one political affiliation at a time t and a different affiliation at time $t+1$. Additionally, a shift may occur due to bad luck — a small training set may exhibit a relationship between Y and Z that does not hold in the larger population. Our goal is to build text classifiers that are robust to such shifts.

Our proposed approach draws inspiration from the causal inference literature, which has developed many methods to control for confounding variables in order to more accurately estimate treatment effects. In particular, we borrow the idea of backdoor adjustment, also known as covariate adjustment or statistical adjustment (Pearl, 2003; Angrist & Pischke, 2008; Imbens & Rubin, 2015). We adapt this idea to the classification setting — the main idea is to condition on the confounding variables at training time, then sum out the confounding variables at testing time. We find that this simple, classifier-agnostic approach can greatly increase classification accuracy in the presence of confounding shift. We emphasize that while our approach is inspired by causal inference techniques, our goal is robust classification, not causal inference.

We conduct empirical evaluations on five real-word text classification tasks. For one set of experiments, we inject confounding bias directly by subsampling training and testing sets to ensure that $P_{train}(Y|Z) \neq P_{test}(Y|Z)$. These experiments allow us to investigate how different methods perform under a variety of shift magnitudes. For a second set of experiments, we consider the natural setting where data arrive chronologically, training on data up to time t and predicting on future data from time $t+k$. These experiments allow us to investigate the extent to which confounding shift happens naturally over time and how robust classifiers are in such a setting. We find that our proposed approach outperforms a number of baselines across multiple datasets. For example, when confounding shift is very high, we find that our approach improves absolute accuracy by up to 20% over a standard classification baseline. We also investigate limitations of our approach, finding some settings in which controlling for confounders can actually reduce accuracy. We discuss future avenues of research to address these cases.

The remainder of this paper is organized as follows: Section 2 formalizes the problem definition, and Section 3 summarizes related work. Section 4 presents our proposed approach when there is only one confounding variable, including experiments in both controlled and natural settings, as well as proposing a method to automatically tune the hyperparameters of our approach. Section 6 extends our approach to the case where there is more than



(a) Causal graph: confounds Z influence both an independent variable X and a dependent variable Y.

(b) Non-causal graph for text classification: confounds Z influence both the text variables X and the class label Y.

Figure 1: Confounding variable in the causal framework and in text classification.

one observed confounding variable, including experiments on additional datasets. We also provide the code and datasets required to reproduce our experiments on GitHub[1].

1.1 Published Work and New Contributions

The primary method of Section 4 has been published in AAAI 2016 (Landeiro & Culotta, 2016). In addition to an expanded related work and discussion, this paper extends this prior work in three key ways:

1. To better establish the real-world effectiveness of our approach, in Section 5.3 we conduct new experiments in which confounding shift appears naturally over time in a political classification task. We find that our approach produces more accurate classifications than the baseline on average, while also exhibiting lower variance over time.

2. In Section 5.4, we present a new method to automatically perform hyperparameter selection for our approach, allowing the practitioner to specify the trade-off between average accuracy and the robustness to confounding shift.

3. Finally, Section 6 is a completely new contribution in which we extend our approach to handle multiple confounding variables. We conduct experiments on an additional dataset for product review classification, again finding improved robustness with our method. We also identify cases of "over-adjustment" for certain confounding variables, which can degrade accuracy, and outline future research directions to address this challenge.

2. Problem Definition

In causal inference, one wishes to estimate the effect of an independent variable X on a dependent variable Y. To do so, one must take into account confounding variables. A confounding variable (also called confound) Z is a covariate that is correlated with both

1. https://github.com/tapilab/jair-2018-confound

X and Y and that partly or completely explains the correlation between X and Y (see Figure 1(a)). If confounding variables are not controlled for in a causal analysis, then one is at risk of reaching a false conclusion concerning the effect of X on Y, as part or all of this effect might be due to Z. For example, a study to estimate the effect of exercise on health may be confounded by age, which is related both to exercise levels and health.

In text classification, one wishes to estimate the quantity $p(Y = y|\vec{X} = \vec{x})$ (also noted $p(y|\vec{x})$ for simplicity): the probability of a class variable Y to take the value y given that the text features $\vec{X}$ are equal to the features vector $\vec{x}$.[2] The impact of confounding variables $\vec{Z}$ (Figure 1(b)) on $\vec{X}$ through $p(\vec{X}|\vec{Z})$ and on Y through $p(Y|\vec{Z})$ are often overlooked, presumably because *prediction*, rather than causal inference, is the primary goal. Indeed, if we assume that the influence of a confounding variable is consistent from training to testing data, then there should be little harm to prediction accuracy. However, this assumption often does not hold in practice when one is working with user-generated data, for at least two reasons:

- First, due to the cost of annotation, training sets are typically quite small, increasing the chance that the correlation between the confounding variable and target variable varies from training to testing data.
- Second, and in our view most importantly, in many domains the relationship between the confound and the target variable is likely to shift over time, either gradually or suddenly, leading to poor accuracy. For example, a discussion topic on social media might be indicative of a certain political affiliation at a time t but indicative of a completely opposite political affiliation at a time $t + 1$. Without properly controlling for confounding variables, studies based on the output of text classifiers are at risk of reaching erroneous conclusions.

In the rest of this paper, we denote the influence of the confound(s) on the target variable at training time $p_{train}(Y|\vec{Z})$; and $p_{test}(Y|\vec{Z})$ stands for the same quantity but at testing/prediction time. Additionally, we define *confounding shift* as the situation where $p_{train}(Y|\vec{Z}) \neq p_{test}(Y|\vec{Z})$. In other words, confounding shift happens when the influence of one or more confounding variables varies between training and testing times. In the presence of confounding shift, text classifiers are at risk of performance loss. For instance, say we wish to classify whether a Twitter user smokes based on their tweets. Suppose that due to the data collection process, there is confounding bias in the training data: say 90% of the female users in the training data are smokers and 50% of the male users in the training data are smokers, such that we have $p_{train}(Smoker|Female) = 0.9$ and $p_{train}(Smoker|Male) = 0.5$. Using such a dataset to train a text classifier will lead to high positive coefficients for features correlated with females because gender and smoking status are so highly correlated. Once the classifier is trained, we wish to use it to predict the smoking habits of new Twitter users. In the case where the confounding bias in the new users is the same as in the training data, then our classifier should not be affected by the user's gender. Indeed, female-related

2. While one can argue that the true text generation process is $P(X|Y)$, not $P(Y|X)$ (Hopkins & King, 2010), modeling the latter is more common in text classification as the *discriminative* model $P(Y|X)$ has lower asymptotic error rates than the *generative* model $P(X|Y)$, given sufficient training examples (Ng & Jordan, 2002).

features will be rightly predictive of the smoking habits of a user. However, imagine we now have $p_{test}(Smoker|Female) = 0.1$ and $p_{test}(Smoker|Male) = 0.5$. In this situation, only 10% of the new female users we wish to classify smoke. Our text classifier will now wrongly associate female-related features with being a smoker, leading to a performance loss at prediction time.

Therefore, the problem we wish to address in this paper is how to obtain accurate and robust predictions in text classification in presence of *confounding shift*. In other words, we want to build text classifiers that have high prediction accuracy and are not sensitive to the fluctuating impact of confounding variables on the target variable. In particular, we study the case where there is one or more confounding variable that is observed at training time but unobserved at testing time.

3. Related Work

In this section, we summarize existing work related to the core problem we try to solve – robust text classification under confounding bias shift – as well as how we improve on this existing work.

3.1 Dataset Shift

Dataset shift (Quionero-Candela et al., 2009) characterizes the issue that appears when the joint distribution of features and labels changes between the training dataset and the testing dataset (i.e. $p_{train}(X, Y) \neq p_{test}(X, Y)$). In particular, covariate shift (Sugiyama et al., 2007; Bickel, Brückner, & Scheffer, 2009; Chen, Monfort, Liu, & Ziebart, 2016) denotes the case of dataset shift in which only the input distribution is different from training to testing (i.e. $p_{train}(X) \neq p_{test}(X)$). Similarly, when the underlying target distribution $p(Y)$ changes over time, either in a sudden way or gradually, then this is called concept drift (Tsymbal, 2004; Widmer & Kubat, 1996). Finally, selection bias has also received some attention (Zadrozny, 2004; Bareinboim, Tian, & Pearl, 2014). It arises when the population of a study is not selected randomly. Instead, some users are more inclined to be selected for the study than others, making it more difficult to draw conclusions for the general population. If a binary variable S indicates whether or not an element of the population is selected, there is presence of selection bias when $p(S = 1|X, Y) \neq p(S = 1)$.

Although it is important for the validity of an observational study to control for these different types of changes in the data distribution, in this paper we direct our attention to the problem of learning under confounding shift. Our goal is to build a classifier that is robust to changes in the relation between the target variable Y of a classifier and an external confounding variable Z. More specifically, we look at the case where the amount of confounding bias is changing between the training set and the testing set, noted $p_{train}(Y|Z) \neq p_{test}(Y|Z)$.

3.2 Fairness in Machine Learning

An emerging research area called Fairness in Machine Learning also tackles problems due to data bias in order to reduce the inequalities of an algorithm between people from different ethnicities, gender, or other socio-demographic attributes. Zemel et al. (2013) and Hajian and Domingo-Ferrer (2013) aim to compensate for what has recently been described as

machine bias or algorithmic bias. This kind of bias exists because supervised machine learning models are trained on historical data that might itself contain bias towards a gender, age category, or ethnicity. For instance, a bank might use a machine learning system to accept or deny a loan to a customer. Because this system has been trained with a set of historical loans' applications and their decisions made by humans, the system will be biased if the original decisions were biased. Results by Islam, Bryson, and Narayanan (2016) on text data indicate that "language itself contains recoverable and accurate imprints of our historic biases, whether these are morally neutral as towards insects or flowers, problematic as towards race or gender, or even simply veridical". This area is related to our work as the goal is to remove the predictive power of some protected variables in order to achieve fairness. A key distinction is that most work in algorithmic bias assumes the protected attributes are known at prediction time; in contrast, we do not observe confounds at prediction time. Additionally, our overall goal is robustness, not algorithmic fairness.

3.3 Social Media Analysis

In social media analysis, methodologies inspired by causal inference have been adapted to learn features arguably more effective than using correlational analyses. Paul (2017) tackled the problem of learning causal associations between word features and class labels using a matching technique and showed that this was useful when creating cross-domain features for sentiment analysis. Although this work does not require to know the confounding variable at training time, it is not yet easily scalable to large text datasets as this method involves training a logistic regression model for every unique word in the corpus. De Choudhury et al. (2016) applied a propensity score matching technique on mental health communities in Reddit to "probe attributes of individuals contemplating suicide in the future." While matching techniques are becoming increasingly used in social media studies of human behavior to control for confounds, they are typically done as a separate step after using off-the-shelf classifiers. Instead, the present work aims to directly control for confounds in a text classifier to improve the validity of observational studies built upon them.

4. Controlling for Confounds in Text Classification

In this section, we present our approach to build a model robust to confounding shift, and we show on controlled and natural experiments that this approach is able to outperform baselines.

Our approach builds upon a common practice in the social sciences known variously as backdoor adjustment, covariate adjustment or statistical adjustment (Pearl, 2003; Gelman & Hill, 2006; Angrist & Pischke, 2008; Imbens & Rubin, 2015). As motivation, consider a standard linear regression model of a dependent variable y. Assume the true data generation process is a linear function of two variables x and z:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 z_i + \epsilon_i \tag{1}$$

with parameters $(\beta_0, \beta_1, \beta_2)$ and error term ϵ_i. In a causal inference setting, x_i may be a treatment indicator, and z_i may be a confounding covariate. Suppose we instead omit the

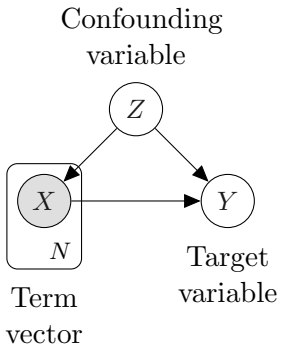


Figure 2: Directed graphical model depicting a confound z influencing both observed text features $\vec{x}$ and class variable y.

confounding covariate from the model:

$$y_i = \hat{\beta}_0 + \hat{\beta}_1 x_i + \hat{\epsilon}_i \tag{2}$$

If we also assume a linear relationship between x and z,

$$z_i = \delta_0 + \delta_1 x_i + \tau_i, \tag{3}$$

then we can substitute (3) into (1) to obtain:

$$y_i = \beta_0 + \beta_2\delta_0 + (\beta_1 + \beta_2\delta_1)x_i + \epsilon_i + \beta_2\tau_i \tag{4}$$

The difference between the coefficient for x_i in (1) and in (4) is known as *omitted variable bias*, in this case $\beta_2\delta_1$. Thus, the bias introduced by omitting the confounding covariate z is a product of two terms: β_2, the strength of association between the confound and y, and δ_1, the strength of association between the confound and x.

Statistical adjustment is simply the process of adding confounds z to a model in order to reduce omitted variable bias, i.e. "controlling for z". While the goal in the social sciences is to produce more precise estimates of causal effects, here we use statistical adjustment to produce more accurate classifiers in the presence of confounding shift. The main idea is rather straightforward: we condition on the confounding variables at training time, then sum out the confounding variables at testing time. We find that this simple, classifier-agnostic approach can greatly increase classification robustness.

More formally, we assume we are provided a training set $D_{train} = \{(\vec{x}_i, y_i, z_i)\}_{i=1}^n$, where each instance consists of a term feature vector $\vec{x}$, a label y, and a single confound z.[3] After fitting on D_{train}, we are then provided a testing set $D_{test} = \{(\vec{x}_i)\}_{i=1}^n$, where both y and z are unobserved. Our goal is to predict the label y_j for each testing instance $\vec{x}_j$, while controlling for an unobserved confound z_j. That is, we assume *we observe the confound at training time, but not at testing time.*

3. In Section 6 we will consider multiple confounds.

Figure 2 displays the directed graphical model for our approach. Omitting the confound z, it depicts a standard discriminative approach to text classification, e.g., modeling $p(y|\vec{x})$ with a logistic regression classifier conditioned on the observed term vector $\vec{x}$. We assume that the confound z influences both the term vector through $p(\vec{x}|z)$ as well as the target label through $p(y|z)$. For example, in a public health setting, y_i may be health status, $\vec{x}_i$ a term vector for online messages, and z_i a demographic variable. Since we do not observe Z at testing time, we sum it out. Based on this graphical model, we can compute the posterior probability at testing time for class y given term vector x as:

$$p(y|x) = \sum_{z \in Z} p(y|x, z)p(z) \tag{5}$$

Thus, to compute Equation 5, we need to estimate two quantities from the labeled training data: $p(y|\vec{x}, z)$ and $p(z)$. For simplicity, we assume in this section that $\vec{x}_i$ is a vector of binary features and that y_i and z_i are binary variables. For $p(z)$, we use the maximum likelihood estimate:

$$p(z = k) = \frac{\sum_{i \in D} \mathbf{1}[z_i = k]}{|D|} \tag{6}$$

where $\mathbf{1}[\cdot]$ is an indicator function. For $p(y|\vec{x}, z)$, any discriminative classifier can be used in which the term features are augmented with confound z. In our experiments, we use L2-regularized logistic regression. To summarize the overall approach:

- At training time, we use the labeled data to:
  - fit a logistic regression model to $p(y|\vec{x}, z)$;
  - compute $p(z)$ using Equation 6.
- At prediction time, for each instance $\vec{x}$ in the testing set:
  - we compute $p(y|\vec{x}, z)$ for all possible values of z using the logistic regression model fit at training time;
  - we use $p(z)$ computed at training time and Equation 5 to compute $p(y|\vec{x})$.

Due to its similarity to the backdoor adjustment equation from Pearl (2003), we refer to this approach as **backdoor adjustment for text classification**.

4.1 Leveraging Undertraining to Tune Adjustment Strength

In this section, we provide some informal intuition as to why we should expect the approach above to result in a more robust classifier, as well describe as a method to allow the researcher to modulate the strength of the adjustment. We compute $p(y|\vec{x}, z)$ efficiently by simply appending two additional features $c_{i,0}$ and $c_{i,1}$ to each instance $\vec{x}_i$ representing $z = 0$ and $z = 1$. If $z_i = 0$, then the first feature is set to v and the second feature is set to 0. Similarly, if $z_i = 1$, then the first feature is set to 0 and the second to v. When $v = 1$, this is equivalent to a one-hot encoding of z_i. By default, we set $v = 1$, but we revisit this decision below. To predict for a new instance, we compute posteriors using Equation 5.

Given that the term vector $\vec{x}$ often contains thousands of variables, it may seem surprising that adding two features for z can have much of an impact on classification. One way to understand this is to consider the problem of *weight undertraining* (Sutton, Sindelar, & McCallum, 2006) in regularized logistic regression. Given the thousands of correlated and overlapping variables used in text classification, optimizing a logistic regression model involves subtle trade-offs among coefficients of related variables, as well as with the magnitude of the coefficients as determined by the L2 regularization penalty. In such settings, it has been observed that the presence of a small number of highly predictive features can lead to smaller than desired coefficients for less predictive features. Sutton et al. reference as an example the autonomous driving system of Pomerleau (1996), in which the presence of a prominent ditch on the side of the road at training time (a highly predictive feature) dominated the model, leading to poor performance in settings where the ditch was not present.

Here, we use undertraining to our advantage. By introducing features for z (a potentially highly predictive feature), we deliberately undertrain the coefficients for terms in $\vec{x}$. In particular, given the objective function of L2-regularized logistic regression, we expect that undertraining will most affect those terms that are correlated with z. For example, if z is gender, then we expect gender-indicative terms to have relatively lower magnitude coefficients using back-door adjustment than other terms. This interpretation allows us to formulate a method to tune the strength of the back-door adjustment. First, we re-write the L2-regularized logistic regression log-likelihood function, distinguishing between coefficients for the term vector θ^x and coefficients for the confounds θ^z, letting θ be the concatenation of θ^x and θ^z:

$$L(D, \theta) = \sum_{i \in D} \log p_\theta(y_i | \vec{x}_i, z_i) - \lambda_x \sum_k (\theta_k^x)^2 - \lambda_z \sum_k (\theta_k^z)^2 \tag{7}$$

where the terms λ_x and λ_z control the regularization strength of the term coefficients and confound coefficients, respectively. A default implementation would set $\lambda_x = \lambda_z = 1$. However, by setting $\lambda_z < \lambda_x$, we can reduce the penalty for the magnitude of the confound coefficients θ^z. This allows the coefficients θ^z to play a larger role in classification decisions than θ^x, thereby increasing the amount of undertraining in θ^x. Our implementation achieves this effect by increasing the confound feature value for v while holding the other feature value to 0. Because we do not standardize the feature matrix[4], inflating the value of v while keeping the same values of $\vec{x}$ encourages smaller values for θ^z, effectively placing relatively smaller L2 penalties on θ^z than on θ^x.

5. Experiments with One Confound

In this section, we present text classification experiments comparing the approach above with several competing baselines, demonstrating how controlling for confounds can increase robustness of text classification. This section will present results with one confound, while Section 6 will present results with multiple confounds.

4. Standardizing values by feature is typically not done in text classification in order to maintain sparse feature vectors.

5.1 Data

Below we describe the four text datasets used to evaluate our approach assuming a single confound. For each dataset, we describe collection and preprocessing steps, and indicate which variable is the target variable Y and which one is the confounding variable Z. Table 1 summarize all data both from this section and Section 6.

Name	TLGD	TCHD	CPD	IMDb	YLCD	YLGD
Section	4				6	
Object	Twitter user	Politician		Movie review	Yelp review	
X	Tweets		Speeches	Review		
Y	Location (NY/LA)	Political party (Rep/Dem)		Sentiment (+/-)		
Z	Gender (M/F)	Topic (Health-care/Other)	In governing party (Yes/No)	Horror movie (Yes/No)	Location (EDH/WI) & Food category (Yes/No)	Location (EDH/WI) & Gender (M/F)

Table 1: Datasets used across this paper. **X** stands for the features, **Y** for the label, and **Z** for the confounding variable(s).

5.1.1 Twitter Dataset with Location Label (TLGD)

For this dataset, we set the task label Y to be user's location and the confounding variable Z to be the user's gender. X contains term features computed from a user's tweets. We refer to this as the TLGD dataset. To build this dataset, we use the Twitter streaming API to collect tweets with geocoordinates from New York City (NYC) and Los Angeles (LA). We gather a total of 246,930 tweets for NYC and 218,945 for LA over a four-day period (June 15th to June 18th, 2015). We attempt to filter bots, celebrities, and marketing accounts by removing users with fewer than 10 followers or friends, more than 1,000 followers or friends, or more than 5,000 posts. To label users by gender, we use U.S. Census data listing first names by gender. We gather the 100 most frequent men names and the 220 most common female names and compare them to the first word in the name field of every Twitter user. Note that the number of male and female names considered is different: we choose these numbers such that roughly half of the collected tweets are published by women and half by men. In the case of a name appearing in the women and men names lists, then we just discard this name and all corresponding users. We then collect all the available tweets (up to 3,200) for each user and represent each user as a binary unigram vector, using standard tokenization. Finally, we subsample this collection and keep the tweets from 6,000 users such that gender and location are uniformly distributed over the users (That is, the dataset is balanced over all possible Y/Z pairs.)

5.1.2 IMDb Dataset

In this dataset, we consider as a confound whether the movie is of the "horror" genre, as determined by the IMDb classification. The label we are trying to predict is if the review is overall positive or negative. Contrary to the Twitter datasets, this data is unevenly distributed among the four possible label/confound pairs. Roughly 18% of movies are horror movies, and 5% of reviews with positive sentiment are of horror movies. We refer to this as the `IMDb` dataset. To build this dataset, we use the data from Maas, Daly, Pham, Huang, Ng, and Potts (2011). It contains 50,000 movie reviews from IMDb labeled with positive or negative sentiment. We remove English stop words, terms that appear fewer than 10 times, and we use a binary vector to represent the presence or absence of features.

5.1.3 Canadian Parliament Dataset

This dataset is used in the task of predicting the party affiliation of a member of the Canadian parliament based on the text of their floor speeches, which is used by political scientists to quantify the partisanship of political debates. The confound here is whether the speaker's party is the governing or opposition party. We obtain data on the 36th and 39th Canadian Parliaments as studied previously (Hirst, Riabinin, & Graham, 2010; Dahllöf, 2012). For each parliament, we have the list of speakers, and for each speaker, we have her political affiliation (simplified to Liberal and Conservative as done by Dahllöf), the text of her speeches, and whether she is from the governing or opposition party. Dahllöf observed that the governing party is a confounding variable for this task. We refer to this dataset as `CPD`.

5.1.4 Congress Dataset

We use this Twitter dataset to predict the party affiliation of a member of the US Congress. The confounding variable is a binarized version of the topic discussed in the tweet: healthcare vs. not healthcare. To build this dataset, we collect the most recent tweets from 520 members of congress. We filter out independent politicians in order to simplify the classification task. Then, we use the healthcare-related keywords defined by Hemphill, Culotta, and Heston (2016) such that for every tweet, we assign a positive label if it contains at least one healthcare keyword; otherwise it is assigned a negative label. This dataset, referred to as `TCHD`, spans from January 2013 to August 2016 for a total of 977K unique tweets.

In the following two sections, we will be running two different types of experiments. First, we will look at controlled experiments in which we precisely subsample the data to observe scenarios where confounding shift happen. In the second section, we will focus on experiments in which confounding shift appears naturally, either gradually or suddenly.

5.2 Controlled Experiments

Using two real-world datasets (`TLGD` and `IMDb`), we conduct experiments in which the relationship between the confound z and the class variable y varies between the training and the testing sets (confounding shift). In these experiments, we directly control the discrepancy between training and testing.

5.2.1 Injecting Confounding Bias

In order to observe the effects of confounding shift, we need to build datasets such that the impact of the confounding variable(s) is different at training time and testing time. To do so, we sample train/test sets with different $p(y|\vec{z})$ distributions while making sure that $p(y)$ and $p(\vec{z})$ stay the same between training and testing in order to isolate the effect of $p(y|\vec{z})$. In this section, we explain our process when there is one confounding variable and both the label and confound are binary variables. Let us assume we have the labeled datasets D_{train}, D_{test}, with elements $\{(\vec{x}_i, y_i, z_i)\}$. We introduce a bias parameter $p(y = 1|z = 1) = b$; by definition, $p(y = 0|z = 1) = 1 - b$. For each experiment, we sample without replacement from each set $D'_{train} \subseteq D_{train}$, $D'_{test} \subseteq D_{test}$. To simulate a change in $p(y|z)$, we use different bias terms for training and testing, b_{train}, b_{test}. We thus sample according to the following constraints:

- $p_{train}(y = 1|z = 1) = b_{train}$;
- $p_{test}(y = 1|z = 1) = b_{test}$;
- $p_{train}(y) = p_{test}(y)$;
- $p_{train}(z) = p_{test}(z)$.

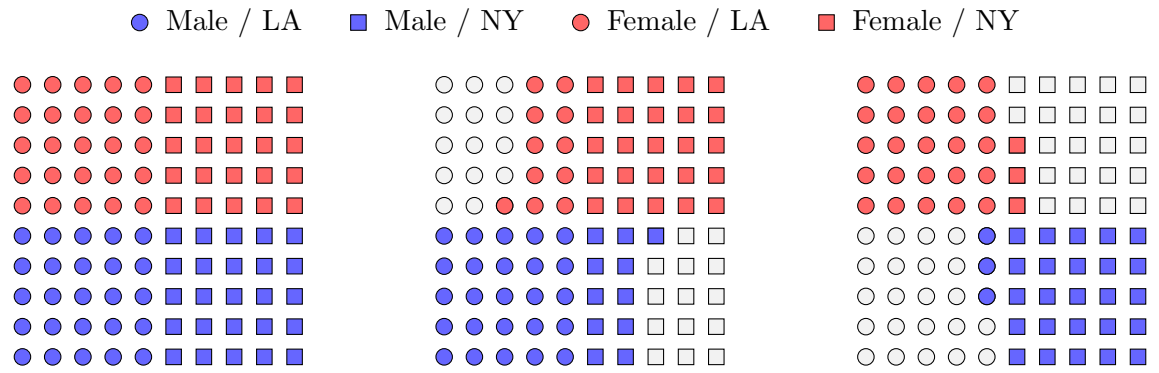
The last two constraints are to isolate the effect of changes to $p(y|z)$. Thus, we fix $p(y)$ and $p(z)$, but vary $p(y|z)$ from training to testing data. We emphasize that we *do not alter* any of the actual labels in the data; we *merely sample instances* to meet these constraints.

This notion of injecting bias is crucial to understand the results of our experiments. To illustrate, Figure 3 shows an example on the `TLGD` dataset. In this dataset, we know the gender and the location – two binary variables in this case – of each Twitter user. Figure 3(a) shows what a perfectly balanced dataset of 100 users would look like. Now, imagine we want to generate biased training and testing datasets such that $p_{train}(\text{NY}|\text{Male}) = 0.3$ and $p_{test}(\text{NY}|\text{Male}) = 0.9$. We need to subsample the original dataset of Figure 3(a) such that we meet the bias requirements as well as we keep $p(\text{Gender}) = 0.5$ and $p(\text{Location}) = 0.5$ in both training and testing datasets. Using our sampling technique, we obtain the training dataset shown in Figure 3(b) and the testing dataset shown in Figure 3(c) where the gray users are left out. We observe that we managed to create datasets differently biased by merely subsampling the original dataset. This sampling techniques tends to reduce the dataset size when we wish to build a highly biased dataset. This might be an important issue when working with a dataset of 100 instances like in this toy example but in the rest of this paper, we work with datasets large enough – several thousands of instances – that this is not a problem.

5.2.2 Experimental Settings

For `TLGD` and `IMDb`, we simulate shifts in train/test confounding as described in Section 5.2.1. We make the bias value b vary from 0.1 to 0.9 (i.e. from 10% to 90% of bias) for both the training and the testing sets and we compare the accuracy of the following models:

- **Logistic Regression (LR)**: Our primary baseline is a standard L2-regularized logistic regression classifier that does not do any adjustment for the confound. It simply models $p(y|x)$.



(a) Original balanced dataset

(b) Training dataset with injected bias of 0.3

(c) Testing dataset with injected bias of 0.9

Figure 3: Complete example on how we inject confounding bias in datasets to run experiments with confounding shift between training and testing times.

- **Back-door Adjustment (BA)**: The approach we have presented in this paper. We also consider the model that makes a stronger covariate adjustment by setting the confounding feature value $v = 10$, which we denote **BAZ10**.
- **Subsampling (LRS)**: A straightforward way to remove bias at training time is to select a subsample of the data such that $p(y, z)$ is uniformly distributed. I.e., if n_{ij} is the number of instances where $y = i$ and $z = j$, then we subsample such that $n_{00} = n_{01} = n_{10} = n_{11}$. This approach unfortunately can discard many instances when there is a strong confounding bias, and furthermore scales poorly as the number of confounds grow.
- **Matching (M)**: Matching is commonly used to estimate causal effects from observational studies (Rosenbaum & Rubin, 1983; Dehejia & Wahba, 2002; Rubin, 2006). To apply these ideas to text classification, we construct a pairwise classification task as follows: for each training instance with $y = i$ and $z = j$, we sample another training instance where $y \neq i$ and $z = j$. For example, for each horror movie with positive sentiment, we match another horror movie with negative sentiment. We then fit a logistic regression classifier optimized to discriminate between each pair of samples, using a learning-to-rank objective (Li, Wu, & Burges, 2007).
- **Sum out (SO)**: In this approach, we model the joint distribution of $p(y, z|x)$. We use a logistic regression classifier where the labels are in the product space of y and z (i.e., labels are $\{(y = 0, z = 0), (y = 0, z = 1), \ldots\}$). At testing time, we sum out over possible assignments to z to compute the posterior distribution for y.

For each b_{train}, b_{test} pair, we sample 5 train/test splits and report the average accuracy.

Notations The following notations are used across the rest of this paper:

- $r_{tr}(y, z)$ denotes the correlation between the task's label y and the task's confound z in the training data.

- $r_{te}(y, z)$ denotes for the correlation between the task's label y and the task's confound z in the testing data.
- We let $\delta_{yz} = r_{tr}(y, z) - r_{te}(y, z)$ be the difference in (y, z) correlation between training time and testing time. This is one way to measure the confounding shift.

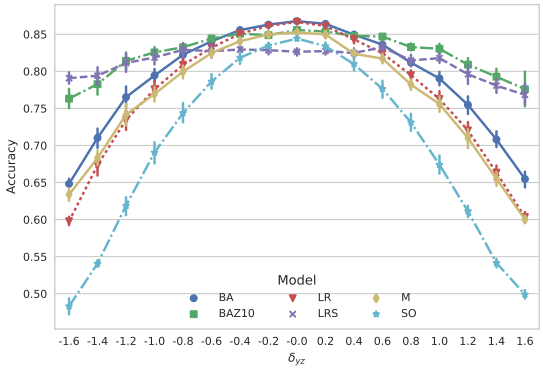
5.2.3 Results

For the `TLGD` and `IMDb` tasks, we construct two plots each. For the first plots (Figures 4(a) and 4(c)), we show testing accuracy as the difference between training and testing bias varies. To determine the x-axis, we compute the Pearson correlation between z and y, and report the difference δ_{yz} between the training and testing correlations. For instance, at $\delta_{yz} = -0.6$, we plot accuracy averaged over samples for which the training correlation is much higher than the testing correlation. In the second set of plots (Figures 4(b) and 4(d)), the x-axis is the testing bias b_{test}; the y-axis is the testing accuracy averaging over all possible training biases b_{train}. Thus, the correlation difference graphs display worst-case scenarios where the training/testing sets vary significantly; whereas the test bias graphs show the average-case accuracy.

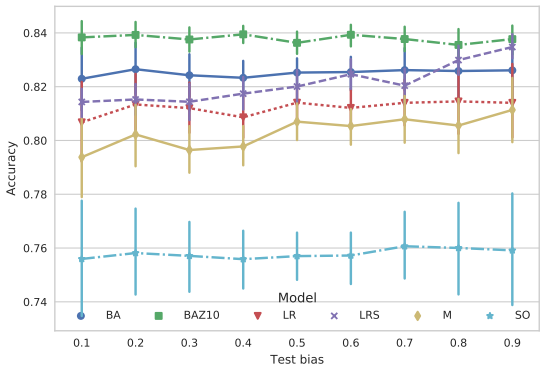
`TLGD` Experiment In Figures 4(a) and 4(b), the best method in the extreme areas are BAZ10 and LRS. They outperform all the other classifiers for $|\delta_{yz}| \geq 0.6$: they are about 15 points better compared to BA, about 20 compared to LR and M, and up to 30 points better than SO. Outside of this interval – in the middle area – BAZ10 is only bested by BA and LR. Moreover, the maximal accuracy loss of BAZ10 to the other classifiers is approximately 2 points when the correlation difference is zero. This suggests that BAZ10 is significantly more robust to confounds than LR, while only sacrificing a minimal amount of accuracy when there is only a small amount of confounding shift. In Figure 4(b), the average accuracy over all the training bias is plotted for every testing bias. BA and BAZ10 are overall more accurate than every other method. SO does poorly overall, with an accuracy between 4 and 8 points less than the other methods. We speculate that this is due to statistical inefficiency: SO must learn a joint model of Y and Z with limited training data, whereas BA models $P(Y|X, Z)$ and $P(Z)$ separately.

`IMDb` Experiment Figures 4(c) and Figure 4(d) display the results for `IMDb`. BA and BAZ10 again appear the most robust to confounding bias. The other methods perform well, except for LRS, which produces results around ten points less than the other methods (for clarity, we have omitted LRS from these figures). We attribute this poor performance to the fact that the distribution of y/z variables is much more skewed here than in `TLGD`, leading LRS to be fit on only a small percent of the training data each time. This also explains why the change in overall accuracy is not as extreme as in the `TLGD` experiments: the confounding effect is minimized because there are relatively few horror movies in the data.

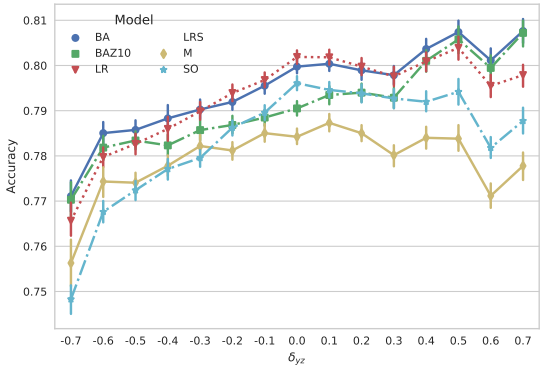
For the `IMDb` and `TLGD` experiments, we additionally compute a paired t-test to compare BAZ10 and LR for each value of the correlation difference (e.g., the x-axis in Figures 4(a) and 4(c)). We find that in 19 cases, BAZ10 outperforms LR; in 8 cases, LR outperforms BAZ10; and in 5 cases the results are not significantly different ($p < 0.01$). As the figures indicate, when the testing data are very similar to the training data with respect to the



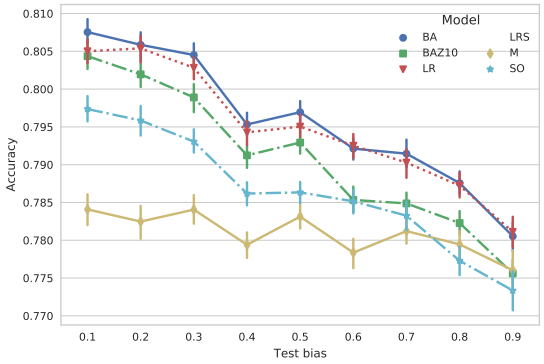
(a) Accuracy as the testing set and training sets differ with respect to $p(y|z)$ for `TLGD`.



(b) Accuracy averaged over training biases for a given testing bias for `TLGD`.



(c) Accuracy as the testing and training sets differ with respect to $p(y|z)$ for `IMDb`.



(d) Accuracy averaged over training biases for a given testing bias for `IMDb`.

Figure 4: Results of the controlled experiments.

confound, BAZ10 is comparable or slightly worse than LR; however, when the testing data differs, BAZ10 outperforms LR, sometimes by a substantial margin (e.g., 20% absolute accuracy increase for `TLGD`).

5.3 Natural Experiments

In the previous section, we subsampled existing datasets in order to introduce datasets and saw that our model is robust to such confounding shift. In this section, we show that confounding shift also happens naturally in datasets, either suddenly or gradually. For both cases, we show that back-door adjustment is able to outperform baselines.

5.3.1 Experimental Settings

`CPD` Experiment For this task, the experimental procedure is slightly different than in the previous experiments. First, recall that in this dataset we aim to predict the party of a politician (Y) confounded by whether the politician is part of the government or the

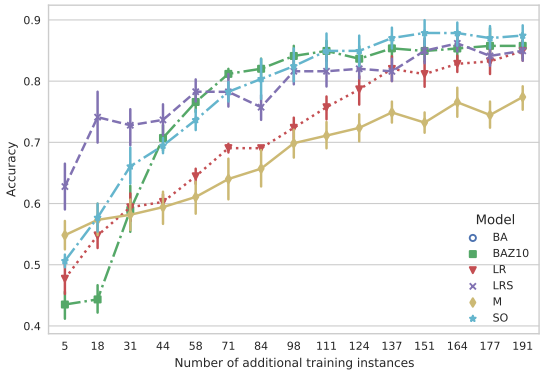
opposition (Z). Unlike the prior two tasks, we do not subsample the data to simulate shifts in $p(y|z)$. Instead, because the governing party shifted from Liberal to Conservative from the 36th to the 39th Parliament, we have a natural dataset to study how a sudden shift in the confounding variable affects accuracy. We initialize D_{train} to be all data from the 36th Parliament. Then, we incrementally add instances from the 39th Parliament to D_{train}. Every time we add an instance from the 39th parliament to D_{train}, we refit our classification model and predict on a held-out set in the 39th Parliament. Therefore, we can report the learning curve showing how each method performs as the training data become more similar to the testing data. Note that initially this task is more difficult than the prior two, since D_{train} begins only with examples where $r(y, z) = 1$ (because all Liberal members are also members of the governing party in the 36th Parliament). For the testing data, $r(y, z) = -1$, since the Conservatives have become the governing party.

`TCHD` **Experiment** The objective in this experiment is to predict the political party of members of Congress based on their tweets while controlling for the topic discussed in each tweet. In order to simplify the experiment and the results' interpretation, the topic discussed in the tweets is limited to either "Healthcare" or "Other". Our goal is to measure how confounding shift appears over time. To do so, we fix the training data to an initial time period t, then sample testing data from future time periods $t + g$. The gap size g determines the time between the training and testing set. Specifically, the training data are created by sampling contiguous 4-week periods from weeks 41-52 of 2013. The testing data are created by sampling 4-week periods from January 2014 to August 2016. In order to minimize the amount of concept drift, we subsample the tweets at training and testing time such that the class label is balanced. We repeat this subsampling step n times for each train/test pair to cover the whole dataset and to increase the amount of cases in our experiment. Additionally, we make sure that users that are present in a training dataset do not appear in a corresponding testing dataset. At training time, we know for each tweet the political party of its author (y) and if it is related to healthcare or not (z) so we can train our different models on this data. We then predict the political party for each tweet in the testing set and we average the predictions by user in order to get a prediction at the user level.

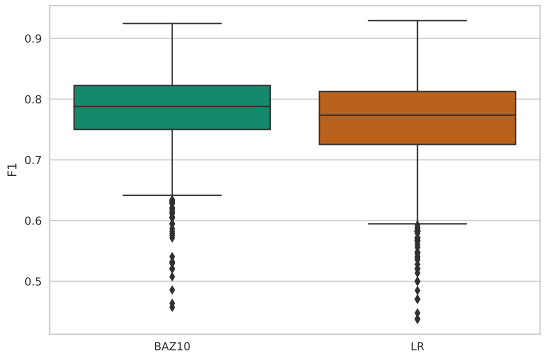
5.3.2 Results

`CPD` **Experiment** Figure 5(a) shows the accuracy for five models when we gradually add instances from the 39th Parliament to data from the 36th and predict on separate instances from the 39th Parliament using 5-fold cross-validation. Note that we do not display the BA model in this figure as it has nearly the same result as BAZ10. Initially – with 5 instances from the 39th parliament – LRS surpasses the other models by five to fifteen percent; however, BAZ10 quickly surpasses LRS once 58 instances from the 39th Parliament are obtained. In the end, SO and BAZ10 have comparable accuracies that are 1% higher than LRS. LR and M exhibit the slowest learning rates, although LR does eventually reach the same accuracy as LRS.

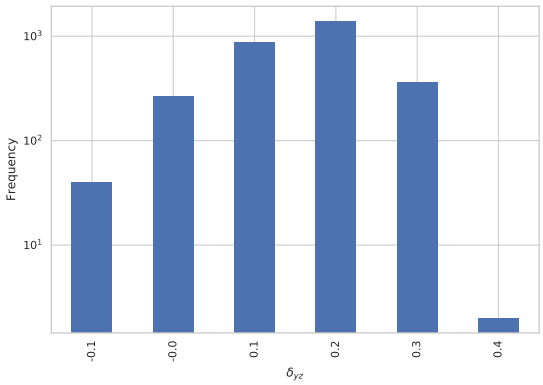
This experiment suggests that when there is an extreme and sudden shift in the confound's influence, it may be best to simply discard much of the data from prior to that shift



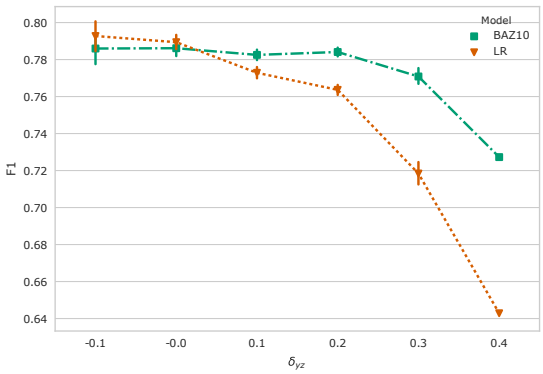
(a) Results for different models on the `CPD` task. BA is not displayed as it performs similarly to BAZ10.



(b) Distribution of F1 scores for BAZ10 and LR on the `TCHD` task. BAZ10 is slightly more performant (higher F1 values) but also more robust (tighter interval).



(c) Confounding shift arises naturally in the `TCHD` task due to the gap between training and testing data.



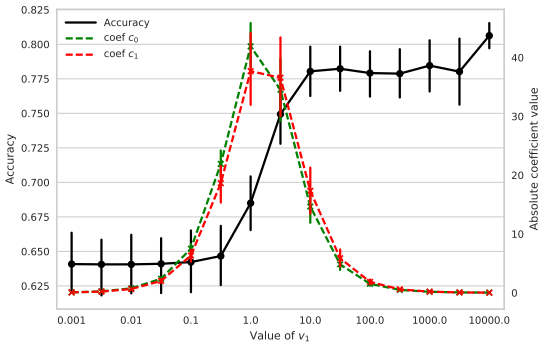
(d) Comparing the performance of LR and BAZ10 given δ_{yz} shows that even though the confounding shift is not as strong as in the controlled experiments, the improvement of BAZ10 over LR is still significant.

Figure 5: Results of the natural experiments.

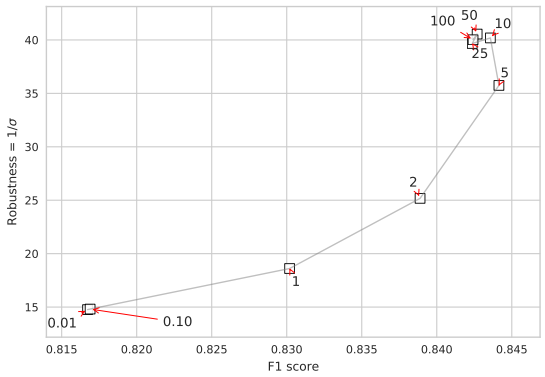
(e.g., the LRS approach). However, once a modest number of instances are available after the shift, BAZ10 is able to make adjustment to overcome the confounding bias.

`TCHD` Experiment The box plot of Figure 5(b) shows the distribution of the F1 scores for both models. We see that overall, BAZ10 slightly outperforms LR – shown by the higher median value – and it is also more robust – shown by the tighter bounds of the box for BAZ10. Figure 5(c) shows the frequency of each confounding shift for each training/testing dataset sampled in this experiment. We observe that δ_{yz} ranges from -.1 to .4 for this natural experiment, indicating that confounding shift does indeed appear naturally over time for this data (though at less extreme values than in the controlled experiments).

Finally, Figure 5(d) displays the F1 score of the two models we compare given the amount of confounding shift, similarly to Figure 4(a). We can see that despite the smaller



(a) Effect of adjustment strength v on confound feature coefficients c_0, c_1 and accuracy on `TLGD`.



(b) Performance (F1) and Robustness of BA on the `TLGD` task for multiple tuning values.

Figure 6: Effect of BA adjustment strength

range of δ_{yz}, it has a significant impact on the performance of LR as its F1 score decreases from .785 when there is no confounding shift to .64 when $\delta_{yz} = .4$. BAZ10 is less affected by this shift; although it has a performance very close to LR when there is no confounding shift, it outperforms LR by around 8 points when $\delta_{yz} = .4$. This demonstrates that confounding shift appears naturally in some cases. Additionally, it shows that even though it appears in smaller scale than in controlled experiments, it is still important to control for it and we are able to do so using our BAZ10 model.

5.4 Tuning Back-Door Adjustment

Thus far, we only assigned the values 1 or 10 to the tuning parameter v of our approach. In this section, we investigate the effect of this tuning parameter, and propose an automated method to select this parameter based on the desired trade-off between accuracy and robustness. Figure 6(a) displays the effect of this parameter for the `TLGD` task. This figure shows the change of the coefficients in absolute value for the confound features c_0 and c_1 (dashed lines) as well as the accuracy (solid line) when v is increasing in `TLGD`. These results are for the case where the bias difference in the training and the testing set is large ($|\text{train bias} - \text{test bias}| > 1.2$). We observe that the accuracy is low and stable when v is less than 10^{-1}. It then increases and begins to plateau starting at $v = 10$. For this dataset, the accuracy gain is a considerable 15 points between the two plateaus. While here we have picked $v = 10$ in all experiments, to allow the practitioner to tune our model, we propose here a method to automatically pick the "best v" by leveraging the training data and our bias injection technique presented in section 5.2.1.

Suppose we are given a training dataset D_{tr} and we wish to fit a back-door adjustment model on this dataset but we do not know what value to pick for the tuning parameter v. The overall idea of our automatic tuning method is to create multiple models with various values for v and to pick the one that performs the best over multiple datasets exhibiting different amounts of confounding shift. Specifically, we create a new BA model m_v for every $v \in V$, where V is a user-defined parameter with default $V = [1, 10, 100, 1000]$. Then, we

subsample D_{tr} to create pairs of tuning and validation datasets $\mathbf{D} = \{(D_{tune}, D_{valid})_i\}_{i=1}^{M}$ with respective biases $\{(b_{tune}, b_{valid})_i\}_{i=1}^{M}$. We make both tuning and validation biases range from .1 to .9 in order to cover the largest range of confounding shift. Then, for every pair of datasets in $\mathbf{D}$, we fit every model m_v on $D_{i,tune}$ and we report the F1 score $F1_{i,v}$ of m_v on $D_{i,valid}$. In the rest of this section, we let $F1_v = \{F1_{i,v}\}_{i=1}^{M}$, $\mu(.)$ to denote the mean, and $\sigma(.)$ for the standard deviation. The objective of the tuning process is to select v such that the model is both accurate (high F1) and robust (low standard deviation) across all possible confounding shifts. Therefore, we pick $\hat{v}$ as follows:

$$\hat{v} = \arg\max_{v} \mu(F1_v) - \frac{1}{\lambda}\sigma(F1_v) \tag{8}$$

where λ is a parameter that defines the trade-off between F1 and robustness. If $\lambda > 1$, then we will favor the overall F1 score rather than robustness when picking v, and vice-versa if $\lambda < 1$. This value defaults to 1 and can be modified by the user at tuning time. To better illustrate this parameter, Figure 6(b) shows the F1 score on the x-axis and the robustness – defined as $\frac{1}{\sigma}$ – on the y-axis for BA models tuned with v ranging from .1 to 100. Our auto-tuning method – aiming at high F1 and high robustness – will pick the model closest to the top right corner ($v = 10$) of the plot when $\lambda = 1$. However, if we increase λ, our tuning technique will be more inclined to lose some robustness in order to achieve better F1, hence it will pick the model tuned with $v = 5$ when we set λ high enough. The behavior will be similar when setting λ closer to 0 with the difference that it will advantage robustness over F1, leading eventually to pick the model tuned with $v = 50$.

Thus, using a default value for λ will pick the value of v that weights F1 and robustness equally, as estimated from internal cross-validation on the training data. In practice, the researcher may select lower values of λ if she has prior expectations of an imminent shift in the data.

5.5 Discussion

In this section, we first explain how back-door adjustment is able to achieve confounding shift robustness by studying the changes in the coefficients of confounder related terms and target variable related terms. Then we discuss the impact of noisy observations of Z and we refer the reader to additional work on this issue.

5.5.1 How Does BA Achieve Robustness?

To understand why BAZ10 is more accurate and more robust than the other methods, we revisit the `TLGD` experiment. Recall that in this task, Y is the location of a user (LA or NY) and Z is the gender of that same user. We plot the coefficients of LR, BA, and BAZ10 classifiers when the bias is 0.9 (i.e. 90% of the New Yorkers are men). In Figure 7, we display these coefficients for the ten features that are most predictive of the class label according to the χ^2 statistic (left) and the ten features that are most predictive of the confounding variable (right). The weights of location-related features (left) decrease a little in the back-door adjustment methods but stay relatively important. On the contrary, the weights of gender-related features (right) are moving very close to zero in the back-door adjustment

methods. Even though these features already have low coefficients in logistic regression, it is important to completely remove them from the classification process so it is not biased by gender. Note that using BAZ10 instead of BA has more of an impact on the gender-related features. These results support the intuition in Section 4.1 that back-door adjustment will impact features correlated with the confound the most through under-training.

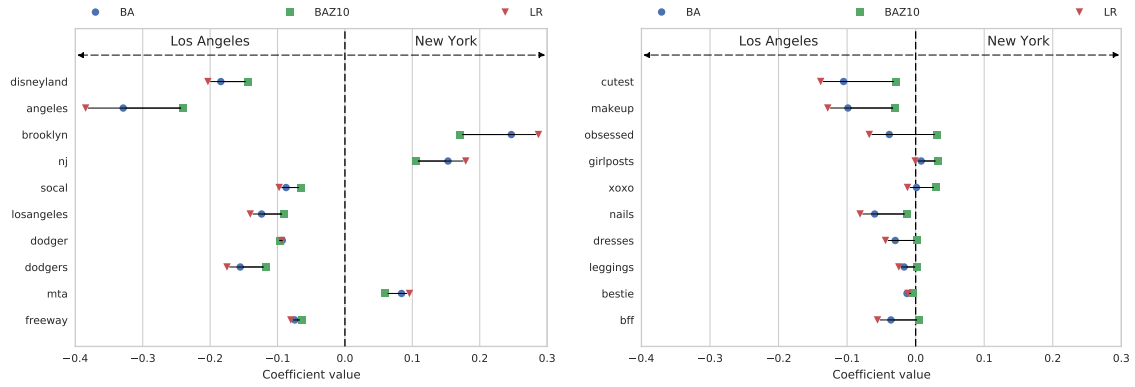


Figure 7: Fit coefficients for the LR, BA, and BAZ10 classifiers with a bias of 0.9 for `TLGD` The left panel shows the 10 features most correlated with the label (location), and the right panel shows the 10 features most correlated with the confound (gender). BAZ10 tends to drive coefficients associated with the confound to 0.

As another way of considering the effect of BA, recall the notion of Simpson's paradox (Simpson, 1951) that describes a situation where the association between two variables X and Y changes direction when we account for a third variable Z (e.g. if X and Y are positively correlated in the general population but are negatively correlated when we condition on Z). For a given classifier, we can compute the number of text features that exhibit Simpson's paradox by identifying coefficients that have one sign when fit to all the data, but have the opposite sign when fit separately to the instances of data where $z = 0$ and again for instances where $z = 1$. That is, we identify coefficients that are predictive of $y = 1$ when fit in aggregate, but are predictive of $y = 0$ when fit in each subgroup (and vice versa). Figure 8 plots the percentage of features that display Simpson's paradox given the strength of the bias in the fitted data for `TLGD` (approximately 22K features). In the BAZ10 case, the number of features displaying Simpson's paradox stays relatively constant; whereas it grows quickly when the bias gets to the extreme values for the other methods. (We observed similar results for `IMDb`.)

From Figures 7 and 8, we conclude that there are two ways in which back-door adjustment improves robustness: (1) by driving to zero coefficients for terms correlated with the confound z; (2) by correcting the sign of coefficients that are predictive of y but have been misled by the confound.

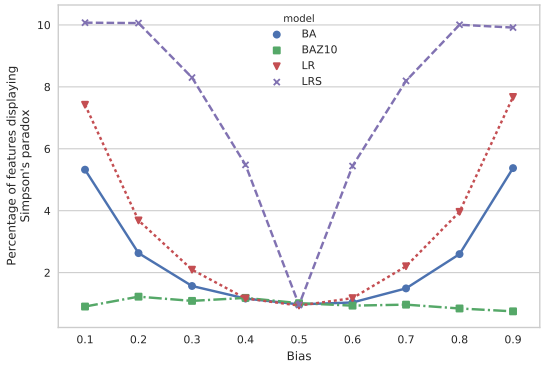


Figure 8: Percentage of features displaying Simpson's paradox in `TLGD`.

5.5.2 Sensitivity of BA to Noise in Z

In our approach, it is assumed that we have access to a training set $D = \{(\vec{x}_i, y_i, z_i)\}_{i=1}^{n}$; that is, each instance is annotated both for the label y and confound z. This may be a prohibitive assumption when we must control for many possible confounds (e.g., gender, race/ethnicity, age, etc.). Because many of these confounds are unobserved and/or difficult to obtain, we develop adjustment methods that can handle noise in the assignment to z in the training data (Landeiro & Culotta, 2017). While beyond the scope of this paper, we present one experiment to provide insight into how our approach degrades with noise in z. In this experiment, we observe z, but we inject increasing amounts of noise in z (i.e., with probability p, change the assignment to z_i to be incorrect). In other words, we artificially decrease the quality of our observations of z and observe how this affects on the quality of back-door adjustment. We then measure how the accuracy of the primary classifier for y varies on a testing set in which the influence of z is decreased (i.e., z correlates strongly with y in the training set, but only weakly in the testing set).

We can see in Figure 9 (built using `TLGD`) that the F1 score decreases as we add more noise to the confounding variable annotations. Notice that when noise is 0, back-door adjustment improves F1 (from 0.78 F1 with no adjustment to 0.82 F1), demonstrating the effectiveness of this approach when the confound is observed at training time. However, when the amount of noise increases, the back-door adjustment model loses much of its robustness, indicating the need for new methods to adjust for unobserved confounds. This can be understood as an instance of *attenuation bias* (Chesher, 1991). We refer the reader to the work of Landeiro and Culotta (2017) for additional methods to handle this setting.

6. Experiments with Multiple Confounds

In the previous section, we directed our attention on the issue of confounding shift when only one confounding variable display such a shift. In practice, there may be several confounding variables displaying shift between training and testing times. In this section, we adjust our binary back-door adjustment to the case where there is more than one confounding variable.

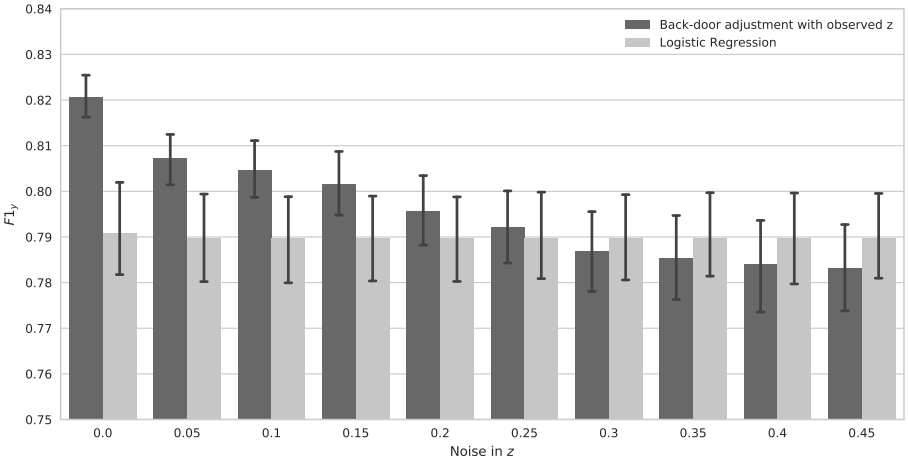


Figure 9: As measurement error in confound z increases, the effectiveness of back-door adjustment decreases.

6.1 Methods

The process to conduct our experiments is similar than when experimenting with a unique confounding variable. Additionally, although the back-door adjustment equation is flexible with the number of confounders, there are some differences in the implementation to compute $p(y|\vec{x})$.

6.1.1 Estimating the Distribution over z

In the case where $\vec{z}$ is a unique binary variable, we used a simple counting method to compute $p(\vec{z})$ (see Section 4). With multiple (m) confounding variables, there might exist interactions we wish to take into account between confounding variables while computing $p(\vec{z}) = p(z_1, z_2, ..., z_m)$. To address this point, we offer three methods to estimate $p(\vec{z})$ in our implementation of the back-door adjustment model:

1. The first method simply uses maximum likelihood estimate to compute $p(\vec{z})$. It is the ideal method when one does not know if there are relationships between z_i and z_j for $i, j \in \{1, ..., m\}$ and $i \neq j$. However, this method requires a large number of instances such that every combination of $\vec{z} = (z_1, z_2, ..., z_m)$ exists in the training dataset. Therefore, this method is not adapted when m is large.

2. The second approach we implemented to compute $p(\vec{z})$ assumes independence between every pair of confounding variables in $\vec{z}$. This approach scales well with the number of confounding variable m as it allows us to compute $p(\vec{z}) = \prod_{i=1}^{m} p(z_i)$, but it makes strong assumptions of independence that might not represent the underlying relationship between confounding variables.

3. Our third and last approach builds assumes a pairwise factorization of the confounding variables. We compute $p(\vec{z})$ as the product $\frac{1}{K}\prod_{i\neq j}\phi_{i,j}$ where $\phi_{i,j}$ is the factor for z_i and z_j and K is the normalizing constant.

In the experiments of this section, we use the first method to estimate $p(\vec{z})$ as we are experimenting with two confounding variables in datasets of several thousand instances.

6.1.2 Adjustment Strength

To model $p(y|\vec{x},\vec{z})$, we add each confound $\vec{z}$ as a separate feature in the classifier. As in Section 4, we may used different feature values to tune the adjustment strength for each variable, optionally using the auto-tuning method of Section 5.4. However, for simplicity, the experiments below use a feature value of 1 for each confound.

6.2 Data

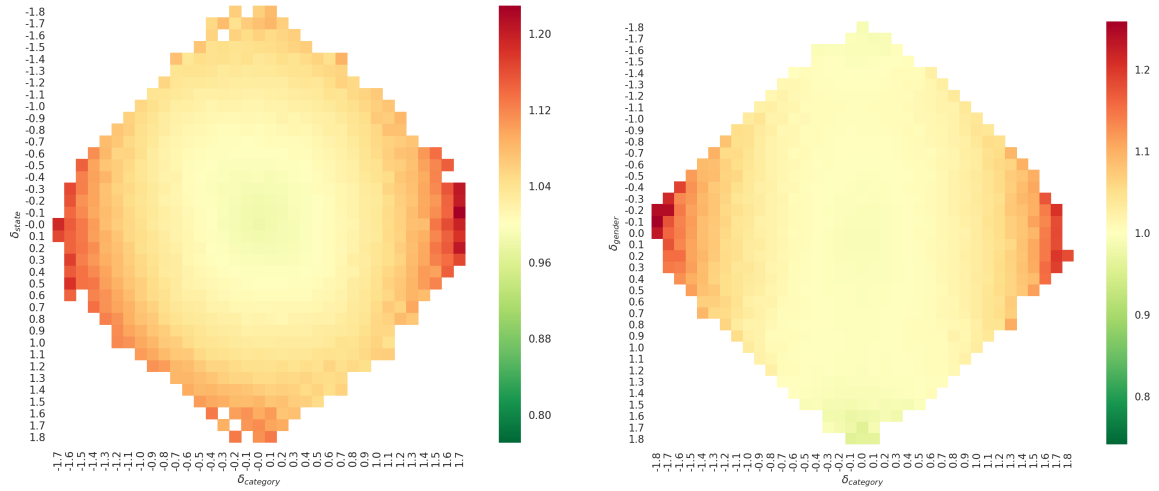
To study multiple confounders, we considered additional data where multiple confounders are present and annotated. For these experiments, we obtain the data from the 8th round of the Yelp Dataset Challenge. It contains around 2.7M text reviews written by 687K unique users for 85.9K different businesses in the USA, Canada (including Quebec), Scotland, and Germany. We restrict the dataset to businesses in the USA and Scotland and our objective is to predict the sentiment of a review. We create the sentiment label by binarizing the reviews' ratings, initially between 1 and 5. Every review with a rating of 1 or 2 is assigned a negative label, and every review with a rating of 4 or 5 is assigned a positive label. Reviews with a rating of 3 are discarded from our analysis. Similarly to the Twitter datasets, we infer the gender of reviewers using the US Census data and remove users for which the gender inference is inconclusive or ambiguous. We also create an additional characteristic for businesses to split them in to group: one containing businesses related to food and drinks (restaurants, food, bars) and the second group encompassing the remaining business categories (services, health). Finally, we keep only reviews from Scotland (noted EDH) and Wisconsin (noted WI), and we create two subtasks with multiple confounding variables.

1. Predict the sentiment of a review while making the confounding impact of the business' location and binary category (i.e. food related or not) vary. We call this dataset `YLCD`.

2. Predict the sentiment of a review while making the confounding impact of the reviewer's gender and the business' binary category change. We call this dataset `YLGD`.

6.3 Experiments

Using two datasets extracted from the Yelp Dataset Challenge (`YLCD` and `YLGD`), we create training and testing datasets with varying amounts of confounding bias. In these datasets, there are two confounders so we adapt our process to inject bias such that we can make the impact of both confounders on the target variable vary independently. We then compare the F1 score of an L2-regularized logistic regression model and the F1 score of our back-door adjustment implementation using simple counting to compute $p(\vec{z})$ without tuning the adjustment strength.



(a) `YLCD` – State can be EDH for Yelp reviews of Scotland-based businesses or WI for reviews of Wisconsin-based businesses. Category can be either Food or Not food.

(b) `YLGD` – Gender can be Female or Male and Category can be Food or Not food.

Figure 10: F1 improvement of back-door adjustment over Logistic Regression (i.e. $F1_{BA}/F1_{LR}$) with two confounding variables.

6.4 Results

We display the obtained results for the two Yelp datasets in Figures 10(a) and 10(b). In these figures, each axis represents the difference between the training correlation and the testing correlation for one confounding variable. For instance, $\delta_{\text{gender}} = r_{tr}(y, gender) - r_{te}(y, gender)$. These axes are identical to the x-axes of umbrella-shaped figures shown presented in earlier experiments. Let us note that predicting the sentiment of a Yelp review is fairly easy as logistic regression achieves an F1 score of 0.95 when there is no confounding shift. In these figures, we display in color the improvement of back-door adjustment over logistic regression as $improvement = \frac{F1_{BA}}{F1_{LR}}$. Therefore, red areas (where $improvement > 1$) mark areas where back-door adjustment is outperforming logistic regression and green areas (where $improvement < 1$) show areas where back-door adjustment falls behind logistic regression. Finally, the white areas of the plot indicates areas where no experiments have been run because training/testing datasets could not be created with the required amount of confounding bias.

6.4.1 Results for `YLCD`

Figure 10(a) shows results for experiments where both the business category/review sentiment and location/review sentiment correlations change between training and testing datasets. Similarly to experiments with one confounding variable, we observe that back-door adjustment is more robust to confounding shift, improving logistic regression results by up to 15%. In the same time, the F1 score of back-door adjustment is always at least 0.98 of the F1 score for logistic regression. These results are encouraging as we manage

to perform robust predictions when two confounding variables are displaying confounding shift.

6.4.2 Results for YLGD

In Figure 10(b), we ran similar experiments but instead of using category and location as confounds, we used category and gender of the user writing the review. Controlling for the business category again shows important improvement – up to 20% – over logistic regression. However, controlling for gender is actually less accurate than using logistic regression, hence the light green areas visible at the top and bottom of the diamond-shaped plot. Although the loss of back-door adjustment is small when controlling for gender, it does indicate that occasionally controlling for a variable can reduce performance. We offer explanations for why this happens in the next section, and propose solutions on how to solve this issue.

6.5 Discussion

In order to better understand why back-door adjustment is outperformed by logistic regression when gender displays confounding shift, we inspected this dataset and this prediction task and we propose the following explanations:

- The first point to recall is that predicting the sentiment of a review (positive or negative) from the text of the review is fairly easy in this dataset as logistic regression manages a F1 score of 0.95 when there is no confounding shift. This shows that there are strong and consistent features across the whole dataset to identify the sentiment of a review.

- Second, we observe by manually inspecting reviews written by men and reviews written by women that these reviews are very similar and offer an objective point of view regardless of the reviewer's gender. This was not the case in Twitter datasets where messages posted by users are much more personal and biased by gender. To get a better understanding of this observation, we train a classifier to predict the gender of a user given the text of a review he/she had written. Using 10-fold cross validation, we obtain a low 0.64 F1 score for this task, supporting the intuition that Yelp reviews are highly objective and not biased by gender.

- Combining the first two points, we note that when the amount of confounding shift is large for gender in our experiments, the features correlated with gender see their coefficients reduced by back-door adjustment but this does not have an important impact on the overall prediction accuracy. Because gender is hard to distinguish from the text in this dataset, confounding shift in gender only has a very weak effect on prediction accuracy. Additionally, by appending features encoding the user's gender to our features matrix, we add features more or less correlated with the label – depending on the confounding bias – while not being predictive, leading to overfitting and loss of performance.

In this case, we showed that controlling for a confounding variable can hurt the performance of back-door adjustment when the influence of this variable on the input data is

limited. This follows from the motivating discussion in Section 4, which indicates that the magnitude of omitted variable bias is a function of the strength of association between the confound and x.

Determining which variables to control for is a long-standing issue in the social sciences (Pearl, 2003; Gelman & Hill, 2006; Angrist & Pischke, 2008; Imbens & Rubin, 2015), and is an important area for future work in text classification. We describe two possible approaches below.

1. A first simple approach would be to use domain knowledge and only control for covariates that have an important impact on the input of the model. This naive approach would be simple to implement but requires domain knowledge and might not take into account covariates with low impact on the features.

2. A more adaptive approach would be to estimate the influence of each covariate one wishes to control for – using the F1 score for instance – and set the adjustment strength of back-door adjustment as a function of this estimate. Implementing this approach would enable back-door adjustment to adapt to the impact of each covariate and we would expect back-door adjustment to not be outperformed by logistic regression in situations of high confounding shift anymore.

7. Conclusion

In this paper, we investigated the problem of confounding shift in text classification, and proposed a solution based on statistical adjustment. This problem finds applications in the field of computational social science where researchers take advantage of the data created by online users and hope to extract behavioral information. Because randomized controlled trials are not possible on this type of data, researchers turn towards observational studies that are sensitive to confounding variables. Due to the fact that the relationship between the confounder and the target variable is likely to shift over time, classifiers used in computational social science studies are at risk of performing poorly and therefore leading to invalid conclusions.

We focus on the problem that rises when there is one or more observed confounding variables. In the case of a single confounder, we observed an improvement of 10 to 20 points of accuracy in the most extreme cases of confounding shift. Additionally, we offered an explanation of how back-door adjustment is robust by reducing the importance of features related to the confounding variable. In the case of multiple confounding variables, we showed that back-door adjustment was also able to make more robust predictions than logistic regression. Additionally, we discussed the effects of back-door adjustment when controlling for a covariate that is not confounding and proposed avenues for future work to solve this problem.

Acknowledgments

This research was funded in part by the National Science Foundation under awards #IIS-1526674 and #IIS-1618244.

References

Angrist, J. D., & Pischke, J.-S. (2008). *Mostly harmless econometrics: An empiricist's companion*. Princeton university press.

Bareinboim, E., Tian, J., & Pearl, J. (2014). Recovering from selection bias in causal and statistical inference. In *Proceedings of The Twenty-Eighth Conference on Artificial Intelligence (CE Brodley and P. Stone, eds.). AAAI Press, Menlo Park, CA*.

Bickel, S., Brückner, M., & Scheffer, T. (2009). Discriminative learning under covariate shift. *Journal of Machine Learning Research*, *10*(Sep), 2137–2155.

Chen, X., Monfort, M., Liu, A., & Ziebart, B. D. (2016). Robust covariate shift regression. In *Artificial Intelligence and Statistics*, pp. 1270–1279.

Chesher, A. (1991). The effect of measurement error. *Biometrika*, *78*(3), 451–462.

Dahllöf, M. (2012). Automatic prediction of gender, political affiliation, and age in swedish politicians from the wording of their speeches - A comparative study of classifiability. *LLC*, *27*(2), 139–153.

De Choudhury, M., Kiciman, E., Dredze, M., Coppersmith, G., & Kumar, M. (2016). Discovering shifts to suicidal ideation from mental health content in social media. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pp. 2098–2110. ACM.

Dehejia, R. H., & Wahba, S. (2002). Propensity score-matching methods for nonexperimental causal studies. *Review of Economics and statistics*, *84*(1), 151–161.

Gelman, A., & Hill, J. (2006). *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press.

Hajian, S., & Domingo-Ferrer, J. (2013). A methodology for direct and indirect discrimination prevention in data mining. *Knowledge and Data Engineering, IEEE Transactions on*, *25*(7), 1445–1459.

Hemphill, L., Culotta, A., & Heston, M. (2016). #polar scores: Measuring partisanship using social media content. *Journal of Information Technology & Politics*, *0*(ja).

Hirst, G., Riabinin, Y., & Graham, J. (2010). Party status as a confound in the automatic classification of political speech by ideology. In *Proceedings of the 10th International Conference on Statistical Analysis of Textual Data (JADT 2010)*, pp. 731–742.

Hopkins, D. J., & King, G. (2010). A method of automated nonparametric content analysis for social science. *American Journal of Political Science*, *54*(1), 229–247.

Imbens, G. W., & Rubin, D. B. (2015). *Causal Inference in Statistics, Social, and Biomedical Sciences*. Cambridge University Press.

Islam, A. C., Bryson, J. J., & Narayanan, A. (2016). Semantics derived automatically from language corpora necessarily contain human biases. *CoRR*, *abs/1608.07187*.

Landeiro, V., & Culotta, A. (2016). Robust text classification in the presence of confounding bias. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Landeiro, V., & Culotta, A. (2017). Controlling for unobserved confounds in classification using correlational constraints. In *Proceedings of the Eleventh International Conference on Web and Social Media, ICWSM 2017, Montréal, Québec, Canada, May 15-18, 2017.*, pp. 580–583.

Lazer, D., Pentland, A. S., Adamic, L., Aral, S., Barabasi, A. L., Brewer, D., Christakis, N., Contractor, N., Fowler, J., Gutmann, M., et al. (2009). Life in the network: the coming age of computational social science. *Science (New York, NY)*, *323*(5915), 721.

Lee, L.-F. (1982). Specification error in multinomial logit models: Analysis of the omitted variable bias. *Journal of Econometrics*, *20*(2), 197–209.

Li, P., Wu, Q., & Burges, C. J. (2007). Mcrank: Learning to rank using multiple classification and gradient boosting. In *Advances in neural information processing systems*, pp. 897–904.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Ng, A. Y., & Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems*, pp. 841–848.

Paul, M. J. (2017). Feature selection as causal inference: Experiments with text classification. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pp. 163–172.

Pearl, J. (2003). Causality: models, reasoning, and inference. *Econometric Theory*, *19*, 675–685.

Pomerleau, D. A. (1996). Neural network vision for robot driving. In *The Handbook of Brain Theory and Neural Networks*. Citeseer.

Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., & Lawrence, N. D. (2009). *Dataset shift in machine learning*. The MIT Press.

Rosenbaum, P. R., & Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, *70*(1), 41–55.

Rubin, D. B. (2006). *Matched sampling for causal effects*. Cambridge University Press.

Schwartz, H. A., Eichstaedt, J. C., Kern, M. L., Dziurzynski, L., Ramones, S. M., Agrawal, M., Shah, A., Kosinski, M., Stillwell, D., Seligman, M. E., et al. (2013). Personality, gender, and age in the language of social media: The open-vocabulary approach. *PloS one*, *8*(9), e73791.

Simpson, E. H. (1951). The Interpretation of Interaction in Contingency Tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, *13*(2), 238–241.

Sugiyama, M., Krauledat, M., & Müller, K.-R. (2007). Covariate shift adaptation by importance weighted cross validation. *The Journal of Machine Learning Research*, *8*, 985–1005.

Sutton, C., Sindelar, M., & McCallum, A. (2006). Reducing weight undertraining in structured discriminative learning. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pp. 89–95. Association for Computational Linguistics.

Tsymbal, A. (2004). The problem of concept drift: definitions and related work. *Computer Science Department, Trinity College Dublin*, *106*.

Webb, G. I., & Ting, K. M. (2005). On the application of roc analysis to predict classification performance under varying class distributions. *Machine learning*, *58*(1), 25–32.

Widmer, G., & Kubat, M. (1996). Learning in the presence of concept drift and hidden contexts. *Machine learning*, *23*(1), 69–101.

Zadrozny, B. (2004). Learning and evaluating classifiers under sample selection bias. In *Proceedings of the twenty-first international conference on Machine learning*, p. 114. ACM.

Zemel, R., Wu, Y., Swersky, K., Pitassi, T., & Dwork, C. (2013). Learning fair representations. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pp. 325–333.