Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Robust Text Classification in the Presence of Confounding Bias

Virgile Landeiro and **Aron Culotta**
Department of Computer Science
Illinois Institute of Technology
Chicago, IL 60616
vlandeir@hawk.iit.edu, aculotta@iit.edu

Abstract

As text classifiers become increasingly used in real-time applications, it is critical to consider not only their accuracy but also their robustness to changes in the data distribution. In this paper, we consider the case where there is a confounding variable Z that influences both the text features X and the class variable Y. For example, a classifier trained to predict the health status of a user based on their online communications may be confounded by socioeconomic variables. When the influence of Z changes from training to testing data, we find that classifier accuracy can degrade rapidly. Our approach, based on Pearl's back-door adjustment, estimates the underlying effect of a text variable on the class variable while controlling for the confounding variable. Although our goal is prediction, not causal inference, we find that such adjustments are essential to building text classifiers that are robust to confounding variables. On three diverse text classifications tasks, we find that covariate adjustment results in higher accuracy than competing baselines over a range of confounding relationships (e.g., in one setting, accuracy improves from 60% to 81%).

1 Introduction

While researchers have investigated automatic text classification algorithms for over fifty years (Maron 1961), they have mostly applied them to topical categorization of documents. More recently, however, emerging cross-disciplinary fields like computational social science (Lazer et al. 2009) and computational epidemiology (Marathe and Ramakrishnan 2013) have created new demand for text classification in areas such as public health surveillance (Dredze 2012), political science (Dahllöf 2012), crisis response (Verma et al. 2011), and marketing (Chamlertwat et al. 2012).

These new domains are notably different from those studied historically; the objects to be classified are often people and their online writings, and the predicted labels may be health status, political affiliation, or personality type. Despite these differences, standard supervised classification algorithms are often used with no customization.

However, to ensure the validity of studies based on the output of text classification, these new domains require classifiers that are robust to *confounding variables*. Confounding variables appear in text classification when a variable is correlated with both the input variables (text features) and the output variable (class label). For example, a classifier trained to predict the political affiliation of a Twitter user may be confounded by an unobserved age variable. This may inflate the coefficients of age-related terms in the classifier (a result of *omitted-variable bias* (Clarke 2005)).

While identifying and controlling for confounding variables is central to much of empirical social science, it is mostly overlooked in text classification, presumably because *prediction*, rather than causal inference, is the primary goal. Indeed, if we assume that the confounding variable's influence 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, 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 confounder and the target variable is likely to shift over time, leading to poor accuracy. For example, diseases may spread to new populations or a new political candidate may attract a unique voter demographic. Without properly controlling for confounding variables, studies based on the output of text classifiers are at risk of reaching erroneous conclusions.

In this paper, we present a text classification algorithm based on Pearl's back-door adjustment (Pearl 2003) to control for confounding variables. The approach conditions on the confounding variable at training time, then sums out the confounding variable at prediction time. We further investigate a parameterized version of the approach that allows us to modulate the strength of the desired adjustment.

We evaluate our approach on three diverse classification tasks: predicting the location of a Twitter user (confounded by gender), the political affiliation of a parliament member (confounded by majority party status), and the sentiment of a movie review (confounded by genre). We find that as the mismatch between training and testing sets increases with respect to the confounder, accuracy can decline precipitously, in one dataset from 85% to 60%. By properly controlling for confounders, our approach is able to reliably improve the robustness of the classifier, maintaining high accuracy even in extreme cases where the correlation between

the confounder and target variable is reversed from training to testing sets.

2 Related Work

In the social sciences, many methods have been developed to control for confounders, including matching, stratification, and regression analysis (Rosenbaum and Rubin 1983; Pourhoseingholi, Baghestani, and Vahedi 2012). Pearl (2003) developed tests for causal graphical models to determine which structures allow one to control for confounders using covariate adjustment, also known as the *back-door adjustment*. As far as we know, we are the first to use back-door adjustments to improve the robustness of text classifiers.

In the machine learning community, selection bias has received some attention (Zadrozny 2004; Sugiyama, Krauledat, and Müller 2007; Bareinboim, Tian, and Pearl 2014). Selection bias in text classification occurs when the distribution of text features changes from training to testing; i.e., $P_{train}(X) \neq P_{test}(X)$. Other work has considered the case where the target distribution $P(Y)$ changes from training to testing (Elkan 2001). In the present work, we address the more challenging case of a changing relationship between target labels Y and a confounder Z, i.e., $P_{train}(Y|Z) \neq P_{test}(Y|Z)$.

Additionally, there has been recent interest in "fairness" in machine learning (Zemel et al. 2013; Hajian and Domingo-Ferrer 2013) — for example, ensuring that a classifier's predictions are uniformly distributed across population groups. However, we do not want to optimize fairness in our domain; e.g., we expect that health status does vary by demographics. Other approaches attempt to remove features that introduce bias (Pedreshi, Ruggieri, and Turini 2008; Fukuchi, Sakuma, and Kamishima 2013); however, existing approaches are not practical in text domains with tens of thousands of features. Thus, as far as we know, our proposed approach is the first large-scale investigation of methods to reduce the effect of confounders in text classifiers.

Volkova, Wilson, and Yarowsky (2013) investigate how gender influences sentiment classification on Twitter, finding that gender-specific sentiment lexicons can improve classification accuracy. In contrast to that work, we do not assume that the confounding variable (in this case, gender) is observed at test time. Furthermore, while their work is tailored to sentiment classification, here we propose a general-purpose solution, evaluated on three different classification tasks.

3 Back-door Adjustment for Text Classifiers

Suppose one wishes to estimate the causal effect of a variable X on a variable Y, but a randomized control trial is not possible. If we have access to a sufficient set of confounder variables Z, then it can be shown that we can estimate the causal effect as follows (Pearl 2003):

$$p(y|\text{do}(x)) = \sum_{z \in Z} p(y|x,z)p(z) \quad (1)$$

This formula is called *covariate adjustment* or *back-door adjustment*. The *back-door criterion* (Pearl 2003) is a graphical test that determines whether Z is a sufficient set of variables to estimate the causal effect. This criterion requires that no node in Z is a descendant of X and that Z blocks every path between X and Y that contains an arrow pointing to X. Notice $p(y|x) \neq p(y|\text{do}(x))$: this do-notation is used in causal inference to indicate that an intervention has set $X = x$.

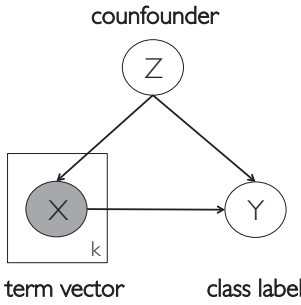


Figure 1: Directed graphical model depicting a confounder variable Z influencing both observed text features X and class variable Y.

While the back-door adjustment is well-studied in causal inference problems, in this paper we consider its application to text classification. We assume we are given a training set $D = \{(\mathbf{x}_i, y_i, z_i)\}_{i=1}^n$, where each instance consists of a term feature vector $\mathbf{x}$, a label y, and a covariate variable z. Our goal is to predict the label y_j for some new instance $\mathbf{x}_j$, while controlling for an unobserved confounder z_j. That is, we assume we observe the confounder at training time, but not at testing time.

Figure 1 displays the directed graphical model for our approach. Omitting the confounder Z, it depicts a standard discriminative approach to text classification, e.g., modeling $P(Y|X)$ with a logistic regression classifier conditioned on the observed term vector $\mathbf{x}$. We assume that the confounder Z influences both the term vector through $P(X|Z)$ as well as the target label through $P(Y|Z)$. For example, in a public health setting, y_i may be health status, $\mathbf{x}_i$ a term vector for online messages, and z_i a demographic variable. The structure of this model ensures that Z meets the back-door criterion for adjustment.

While back-door adjustment is typically presented as a method of identifying the causal effect of X on Y, here we are not attempting any causal interpretation. (Indeed, it would be strange to assert that using a term *causes* one to have a class label.) However, Equation 1 provides a framework for making a prediction for Y given X that controls for Z. In doing so, we can train a classifier that is robust to the case where $P(Y|Z)$ changes from training to testing data.

We use Equation 1 to classify test example $\mathbf{x}$. We assume that z is observed for training examples, but not for testing examples. Thus, we need to estimate two quantities from the labeled training data, $p(y|\mathbf{x}, z)$ and $p(z)$. For simplicity, we assume in this paper that $\mathbf{x}_i$ 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:

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

where $\mathbf{1}[\cdot]$ is an indicator function. For $p(y|\mathbf{x}, z)$, we use L2-regularized logistic regression, described in more detail below.

3.1 Tuning the Adjustment Strength

From an implementation perspective, the approach above is rather straightforward: $p(z)$ is computed using the maximum likelihood estimate above. We compute $p(y|\mathbf{x}, z)$ efficiently by simply appending two additional features $c_{i,0}$ and $c_{i,1}$ to each instance $\mathbf{x}_i$ representing $z = 0$ and $z = 1$. The first (resp. second) feature is set to v_1 if $z_i = 0$ (resp. $z_i = 1$) and the second feature (resp. first) is set to 0. In the default case, we let $v_1 = 1$ but we revisit this decision in the next section. To predict for a new instance, we compute posteriors using Equation 1. Here, we give some intuition as to why we expect this approach to help, as well as a method to allow the researcher to modulate the strength of the adjustment.

Given that the term vector $\mathbf{x}$ often contains thousands of variables, it may seem surprising that adding two additional 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, and 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 tradeoffs 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 $\mathbf{x}$. In particular, given the objective function of L2-regularized logistic regression, we expect that undertraining will most effect 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 confounders θ^z, letting θ be the concatenation of θ^x and θ^z:

$$L(D, \theta) = \sum_{i \in D} \log p_\theta(y_i|\mathbf{x}_i, z_i) - \lambda_x \sum_k (\theta^x_k)^2 - \lambda_z \sum_k (\theta^z_k)^2$$

where the terms λ_x and λ_z control the regularization strength of the term coefficients and confounder 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 confounder 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 confounder feature value for v_1 while holding the other feature value to 0. Because we do not standardize the feature matrix, inflating the value of v_1 while keeping the same values of $\mathbf{x}$ encourages smaller values for θ^z, effectively placing relatively smaller L2 penalties on θ^z than on θ^x.

4 Experiments

Using three real-world datasets, we conducted experiments in which the relationship between the confounder Z and the class variable Y varies between the training and testing set. We consider two scenarios, one in which we directly control the discrepancy between training and testing, and another in which the relationship between Z and Y is suddenly reversed.

To sample train/test sets with different $P(Y|Z)$ distributions, we assume we have labeled datasets D_{train}, D_{test}, with elements $\{(\mathbf{x}_i, y_i, z_i)\}$, where y_i and z_i are binary variables. 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.

We evaluate our approach on three different text classification datasets, each of which has different properties relative to the distribution of the label classes and the confounder classes.

Twitter Dataset The task here is to predict the location of a Twitter user from their messages, where gender is a potential confounder. 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. We then label unique users with their gender using U.S. census name data, removing ambiguous names. 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.

For this paper, we predict location with gender as the confounding variable.[1] Thus, we let $y_i = 1$ indicate NYC and $z_i = 1$ indicate Male. Due to how we build this dataset, the data is evenly distributed across the four possible y/z pairs. We refer to this as the **Twitter** dataset.

IMDb Dataset In this task, we predict the sentiment of a movie review confounded by movie genre using the IMDb data from Maas et al. (2011). It contains 50,000 movie reviews from IMDb labeled with positive or negative sentiment. We remove English stopwords, terms that appear fewer than 10 times, and we use a binary vector to represent the presence or absence of features.

We consider as a confounder whether the movie is of the "horror" genre, as determined by the IMDb classification. Thus, we let $z_i = 1$ for horror movies, and $z_i = 0$ otherwise. Contrary to the Twitter dataset, this data is unevenly distributed amongst the four possible label/confounder 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.

Canadian Parliament Dataset Our final task is to predict the party affiliation of a member of parliament based on the text of their floor speeches, which is used by political scientists to quantify the partisianship of political debates. The confounder 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, and 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 in Dahllöf (2012)), the text of her speeches, and whether she is from the governing or opposition party. It has been observed in Dahllöf (2012) that governing party is a confounding variable for this task. Thus, we set the confounding variable z_i to 1 if speaker i is a member of the governing party or 0 otherwise. We set $y_i = 1$ for Liberal members and $y = 0$ for Conservative members.

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 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 to D_{train} additional instances from the 39th Parliament. When each additional instance is added, we refit our classification model and predict on a held-out set in the 39th Parliament. Thus, we report the learning curve showing how each method performs as the training data become more similar to the testing data.

[1] We also predicted gender with location as the confounder and obtained similar results as those below; we omit these for brevity.

Note that initially this task is more difficult than the prior two, since D_{train} begins only with examples where $P(z = 1|y = 1) = 1$ (because all Liberal members are also members of the governing party in the 36th Parliament). For the testing data, $P(z = 1|y = 1) = 0$, since the Conservatives have become the governing party. We refer to this as the **Parliament** dataset.

Experimental settings: For **Twitter** and **IMDb**, we simulate shifts in train/test confounding as described above. 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 several classification models. For each b_{train}, b_{test} pair, we sample 5 train/test splits and report the average accuracy. For **Parliament**, we use 5-fold cross-validation on the 39th Parliament; each fold reserves a different 20% of the 39th Parliament for testing. The remaining instances are added to the 36th Parliament data incrementally to construct a learning curve.

4.1 Models

We compare the following models:

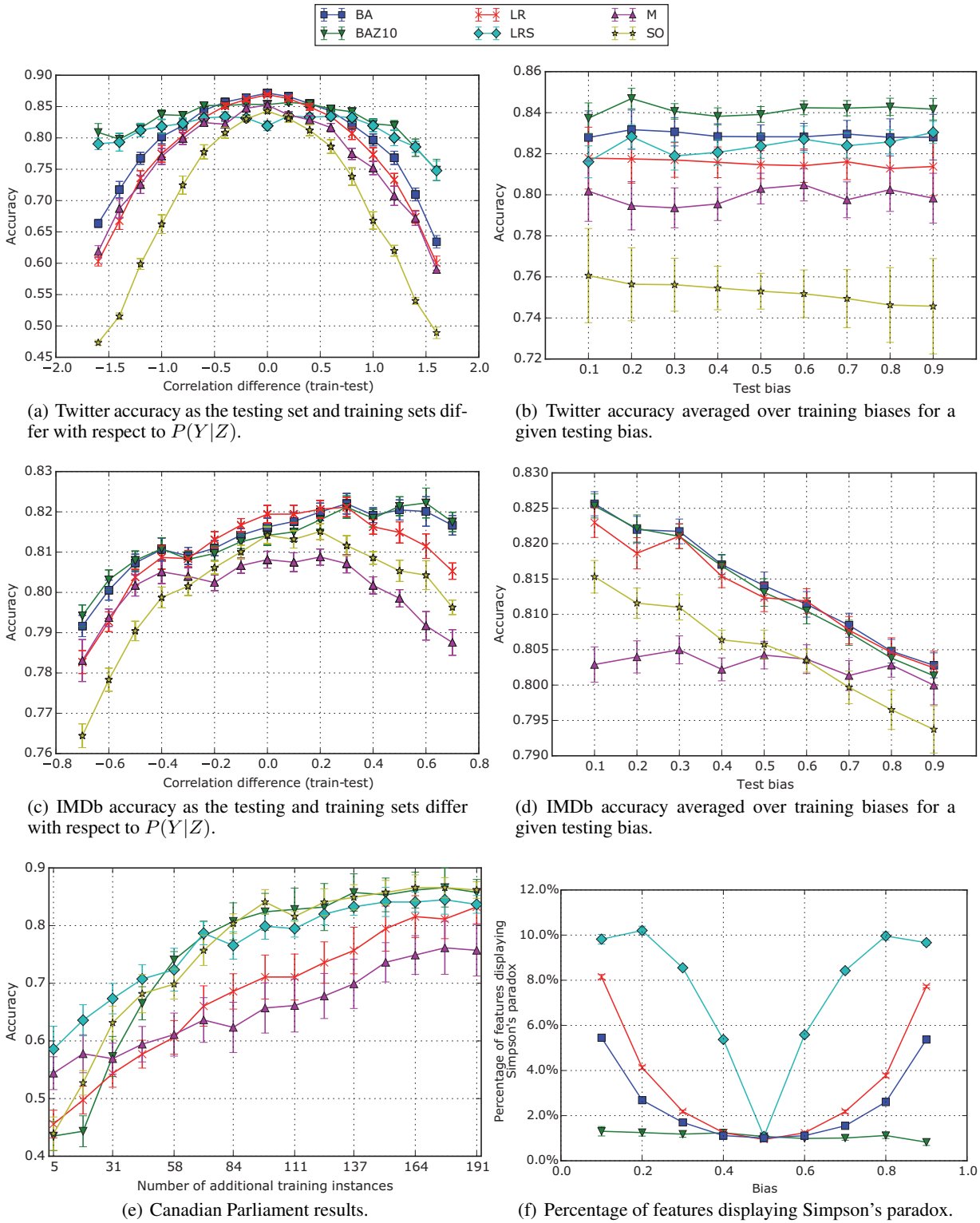
Logistic Regression (LR) Our primary baseline is a standard L2-regularized logistic regression classifier that does not do any adjustment for the confounder. It simply models $P(Y|X)$.

Back-door Adjustment (BA) The approach we have advocated in this paper. We also consider the model that makes a stronger covariate adjustment by setting the confounding feature value $v_1 = 10$, 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 confounders grow.

Matching (M) Matching is commonly used to estimate causal effects from observational studies (Rosenbaum and Rubin 1983; Dehejia and 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, and 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.



(a) Twitter accuracy as the testing set and training sets differ with respect to $P(Y|Z)$.

(b) Twitter accuracy averaged over training biases for a given testing bias.

(c) IMDb accuracy as the testing and training sets differ with respect to $P(Y|Z)$.

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

(e) Canadian Parliament results.

(f) Percentage of features displaying Simpson's paradox.

Figure 2: Experimental results. Error bars show the standard error of the mean.

5 Results

For the **Twitter** and **IMDb** tasks, we construct two plots each. For the first plots (Figures 2(a) and 2(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 differ-

ence between the testing and training correlations. In the second set of plots (Figures 2(b) and 2(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.

5.1 Twitter Experiment

In Figures 2(a) and 2(b), the best method in the extreme areas are BAZ10 and LRS. They outperform all the other classifiers in the interval $[-1.6, -0.6] \cup [0.6, 1.6]$: 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 0. This suggests that BAZ10 is significantly more robust to confounders than LR, while only sacrificing a minimal amount of accuracy when the confounders have little impact. In Figure 2(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.

To understand why BAZ10 is more accurate and more robust than the other methods, 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 3, we display these coefficients for the ten features that are most predictive of the class label according to the χ^2 statistic (top) and the ten features that are most predictive of the confounding variable (bottom). The weights of location-related features (top) decrease a little in the back-door adjustment methods but stay relatively important. On the contrary, the weights of gender-related features (bottom) 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 3 that back-door adjustment will impact features correlated with the confounder the most through under-training.

As another way of considering the effect of BA, recall the notion of Simpson's paradox (Simpson 1951). In causal studies, Simpson's paradox arises when the effect of X on Y is found to be positive in the general population, but negative in each subpopulation defined by the confounder variable Z. For example, suppose smoking is found to cause cancer in the general population, but is found not to cause cancer when considering male and female populations separately. 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

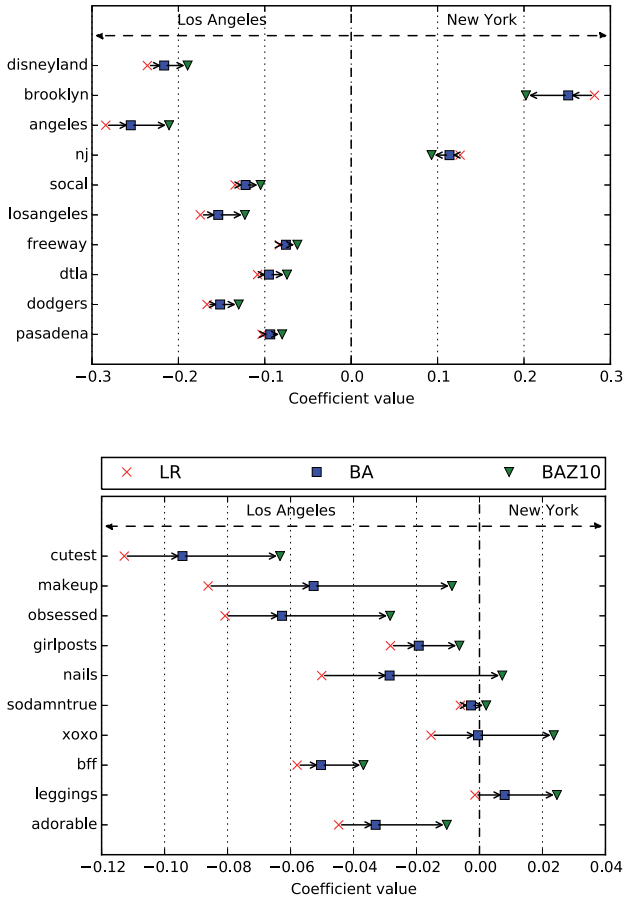


Figure 3: Fit coefficients for the LR, BA, and BAZ10 classifiers with a bias of 0.9 in the Twitter experiment. The top panel shows the 10 features most correlated with the label (location), and the bottom panel shows the 10 features most correlated with the confounder (gender). BAZ10 tends to drive coefficients associated with the confounder to 0.

fit in each subgroup (and vice versa). Figure 2(f) plots the percentage of features that display Simpson's paradox given the strength of the bias in the fitted data. The **Twitter** data contain 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 on IMDb data.)

From Figures 3 and 2(f), 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 confounder Z; (2) by correcting the sign of coefficients that are predictive of Y but have been misled by the confounder.

Finally, Figure 4 displays the effect of the v_1 parameter in the BA approach, which controls the strength of the back-door adjustment. This figure shows the change of the scaled coefficients in absolute value for c_0 and c_1 (dashed lines) as well as the accuracy (solid line) when v_1 is in-

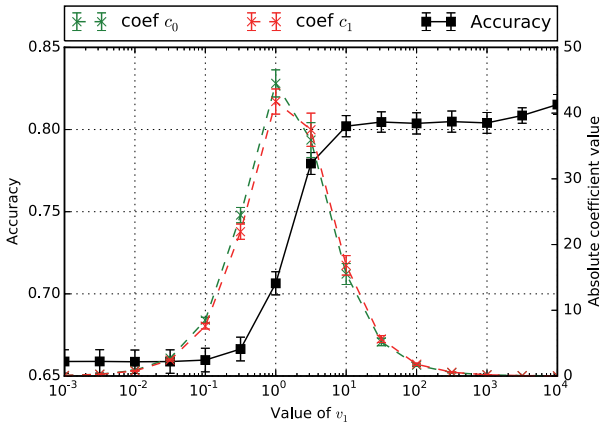


Figure 4: Effect of adjustment strength v_1 on confounder feature coefficients c_0, c_1 and accuracy on Twitter dataset.

creasing in **Twitter**. 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_1 is less than 10^{-1}. It then increases and begins to plateau starting at $v_1 = 10$. For this dataset, the accuracy gain is a considerable 15 points between the two plateaus. While here we have considered $v_1 = 10$ in all experiments, cross-validation may be used to select a value that produces the desired robustness level for a given task.

5.2 IMDb Experiment

Figures 2(c) and Figure 2(d) display the results for IMDb data. 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 **Twitter**, 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 **Twitter** experiments: the confounding effect is minimized because there are relatively few horror movies in the data.

For the IMDB and Twitter 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 2(a) and 2(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 confounder, 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 in Twitter).

5.3 Canadian Parliament Experiment

Finally, we consider the Canadian Parliament task, in which the confounder relationship flips suddenly, and we report the lag required for the methods to recover from this shift. Figure 2(e) 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 confounder's influence, it may be best to simply discard much of the data from prior to that shift (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.

6 Conclusion

In this paper, we have proposed an efficient and effective method of using back-door adjustment to control for confounders in text classification. Across three different datasets, we have found that back-door adjustment improves classifier robustness when the confounding relation varies from training to testing data, and that an additional parameter can be used to strengthen the adjustment for cases of extreme confounding bias. We have found that back-door adjustment both reduces the magnitude of coefficients correlated with the confounder, as well as corrects the sign of coefficients associated with the target class label.

In our experiments, we have assumed that we observe the confounding variable at training time, and that the confounder is a single binary variable. In future work, we will consider the case where we only have a noisy estimate of Z at training time (Kuroki and Pearl 2014), as well as the case where Z is a vector of variables.

Acknowledgments

This research was funded in part by support from the IIT Educational and Research Initiative Fund and in part by the National Science Foundation under grant #IIS-1526674. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

References

Bareinboim, E.; Tian, J.; and 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*.

Chamlertwat, W.; Bhattarakosol, P.; Rungkasiri, T.; and Haruechaiyasak, C. 2012. Discovering consumer insight from twitter via sentiment analysis. *J. UCS* 18(8):973–992.

Clarke, K. A. 2005. The phantom menace: Omitted variable bias in econometric research. *Conflict Management and Peace Science* 22(4):341–352.

Dahllöf, M. 2012. Automatic prediction of gender, political affiliation, and age in swedish politicians from the wording of their speechesa comparative study of classifiability. *Literary and linguistic computing* fqs010.

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

Dredze, M. 2012. How social media will change public health. *IEEE Intelligent Systems* 27(4):81–84.

Elkan, C. 2001. The foundations of cost-sensitive learning. In *International joint conference on artificial intelligence*, 973–978.

Fukuchi, K.; Sakuma, J.; and Kamishima, T. 2013. Prediction with model-based neutrality. In *Machine Learning and Knowledge Discovery in Databases*. Springer. 499–514.

Hajian, S., and 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.

Hirst, G.; Riabinin, Y.; and 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)*, 731–742.

Kuroki, M., and Pearl, J. 2014. Measurement bias and effect restoration in causal inference. *Biometrika* 101(2):423–437.

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.

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

Maas, A. L.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and 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*, 142–150. Portland, Oregon, USA: Association for Computational Linguistics.

Marathe, M., and Ramakrishnan, N. 2013. Recent advances in computational epidemiology. *IEEE intelligent systems* 28(4):96.

Maron, M. E. 1961. Automatic indexing: an experimental inquiry. *Journal of the ACM (JACM)* 8(3):404–417.

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

Pedreshi, D.; Ruggieri, S.; and Turini, F. 2008. Discrimination-aware data mining. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 560–568. ACM.

Pomerleau, D. 1996. Neural network vision for robot driving. *Early visual learning* 161–181.

Pourhoseingholi, M. A.; Baghestani, A. R.; and Vahedi, M. 2012. How to control confounding effects by statistical analysis. *Gastroenterology and Hepatology from bed to bench* 5(2):79.

Rosenbaum, P. R., and 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.

Simpson, E. H. 1951. The interpretation of interaction in contingency tables. *Journal of the Royal Statistical Society. Series B (Methodological)* 238–241.

Sugiyama, M.; Krauledat, M.; and 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.; and 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*, 89–95. Association for Computational Linguistics.

Verma, S.; Vieweg, S.; Corvey, W. J.; Palen, L.; Martin, J. H.; Palmer, M.; Schram, A.; and Anderson, K. M. 2011. Natural language processing to the rescue? extracting" situational awareness" tweets during mass emergency. In *ICWSM*.

Volkova, S.; Wilson, T.; and Yarowsky, D. 2013. Exploring demographic language variations to improve multilingual sentiment analysis in social media. In *EMNLP*, 1815–1827.

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

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