PA

Machine Learning Predictions as Regression Covariates

Christian Fong[1] **and Matthew Tyler**[2]

[1] *Assistant Professor, Department of Political Science, University of Michigan, Ann Arbor, MI, USA. Email: cjfong@umich.edu*
[2] *Ph.D. Candidate, Department of Political Science, Stanford University, Stanford, CA, USA. Email: mdtyler@stanford.edu*

Abstract

In text, images, merged surveys, voter files, and elsewhere, data sets are often missing important covariates, either because they are latent features of observations (such as sentiment in text) or because they are not collected (such as race in voter files). One promising approach for coping with this missing data is to find the true values of the missing covariates for a subset of the observations and then train a machine learning algorithm to predict the values of those covariates for the rest. However, plugging in these predictions without regard for prediction error renders regression analyses biased, inconsistent, and overconfident. We characterize the severity of the problem posed by prediction error, describe a procedure to avoid these inconsistencies under comparatively general assumptions, and demonstrate the performance of our estimators through simulations and a study of hostile political dialogue on the Internet. We provide software implementing our approach.

Keywords: machine learning, classification, inference, instrumental variables

1 Introduction

In many regression analyses, both the outcomes and the covariates are observed by the researcher. However, in other cases, the covariates may be missing for some observations. Perhaps the covariates are not collected for some subset of the data. For example, some states record race in their voter files, but others do not. Alternatively, the covariates might require costly hand-coding to measure, and the dataset may be too large to hand-label most of the observations.

Social scientists have begun to leverage supervised machine learning (ML) to impute the values of these missing covariates based on supplemental data. Imai and Khanna 2016 use a Bayesian algorithm to predict race (the covariate) from surname (the supplemental data) to regress turnout (the outcome) on race. Stewart and Zhukov, 2009 use an ensemble to predict whether a Russian government memo is activist or conservative with regard to use of force (the covariate) from the memo's text (the supplemental data) to measure the correlation between use of force and whether the memo's author is civilian or military. In these cases and many others (Grimmer, Messing, and Westwood, 2012; King, Pan, and Roberts, 2013; Jamal *et al.*, 2015; Anastasopoulos *et al.*, 2016; Theocharis *et al.*, 2016), using ML techniques to impute missing covariates has opened hitherto difficult to analyze data sources to large-scale regression analysis.

However, these ML predictions differ from the underlying covariate for some observations. We show that plugging in these ML predictions without regard for prediction error, as many studies do, leads to bias, inconsistency, and inappropriately small estimates of standard errors. In our example application, the bias and overconfidence are so severe that the 95% confidence interval of this plug-in estimator lies entirely outside of the 95% confidence interval of a baseline estimator known to be unbiased, consistent, and have correct coverage. In fact, simply plugging in the ML predictions as if they were the real covariates would have caused us to arrive at the complete opposite conclusion. The problem is not addressed by intuitive strategies previous studies have

Political Analysis (2021)
vol. 29: 467–484
DOI: 10.1017/pan.2020.38

Published
11 November 2020

Corresponding author
Matthew Tyler

Edited by
Jeff Gill

employed, such as bootstrapping and integrating over the uncertainty in the predictions. While statistics and econometrics provide methods for measurement error, such as multiple imputation, we show that these solutions perform poorly when most observations contain missing data, as often happens in ML.[1]

In some cases, the best option is to not use ML at all, and to instead perform the analysis in the subset of the data that has been labeled by hand. This estimator is guaranteed to be unbiased and consistent under minimal assumptions. But if an exclusion restriction that we enumerate is satisfied, we show that the analyst can use the classifier's predictions to improve statistical efficiency without introducing inconsistency or overconfidence. This approach combines a new sample splitting scheme and a general method of moments (GMM) estimator that leverages two intuitions. First, the sometimes-missing covariate is observed for some observations and regression coefficients can be consistently estimated using only that subset of the data. Second, for those observations where the covariate is missing, the ML classifier's[2] predictions can be used as instruments for the true covariates, and regression coefficients can be estimated using two-stage least squares. Our GMM combines these two estimators to make an efficient, consistent estimator that has analytic standard errors, runs quickly on large data sets, and permits tests of and sensitivity analyses for its strongest assumption. It performs well in simulations, and we highlight the substantive impact of its performance gains through an application.[3] We provide an implementation of this GMM in an R package available online.[4]

2 Challenges

The prediction error associated with using classifier outputs as regression covariates is a form of measurement error, a well-studied problem in econometrics and statistics. We first clarify the need to address prediction error by showing that plugging predictions in for the true covariates leads to bias, inconsistency, and standard errors that are too small. We then clarify how two features common in modern ML problems—endogenously derived predictions and missing covariates for a large proportion of the observations—lead theoretically appealing extant solutions to perform poorly in practice.

2.1 Measurement Error from Predictions

Suppose our goal is to fit the linear regression of y on a vector of covariates $\boldsymbol{x}$:

$$y = \boldsymbol{x}^{\mathsf{T}}\beta + \epsilon, \quad \mathbb{E}[\boldsymbol{x}\epsilon] = \mathbf{0}. \tag{1}$$

What distinguishes our task from typical regression analyses is that some covariates are missing for some observations. We denote the sometimes-missing covariates as $\boldsymbol{x}_u$,[5] the always-observed covariates as $\boldsymbol{x}_o$, and rewrite the original vector as $\boldsymbol{x} = (\boldsymbol{x}_u, \boldsymbol{x}_o)$. We assume throughout that $\boldsymbol{x}_o$ includes an intercept. In a regression of presidential vote choice on whether an individual identifies as white in their state's voter file, $\boldsymbol{x}_u$ corresponds to whether the individual identifies as white, information which is unobserved in some state voter files.

A ML approach to this task is to use some algorithm (e.g., logistic regression, random forests, support vector machines [SVMs]) to generate predictions of the covariates, $\boldsymbol{z}_u$, that are supposed to approximate the missing covariates, $\boldsymbol{x}_u$. For example, we could follow Imai and Khanna, 2016

1 Noteworthy exceptions are Hopkins and King 2010 and Jerzak, King, and Strezhnev 2018, which consistently estimate population proportions in the presence of prediction error.
2 Although our exposition focuses on classifiers, all of our work applies not just to non-binary missing variables.
3 The replication data is available interactively at Fong and Tyler (2020a) or for download at Fong and Tyler (2020b).
4 See github.com/matthewtyler/predictionError.
5 Sometimes $\boldsymbol{x}_u$ is obtained from human coding (e.g., whether a social media post is criticizing the government). Human coding can be contentious; for example, two coders might disagree about what is and what is not a criticism of the government. In the conclusion, we discuss how to think about human coding error in our framework.

in using Bayes' Rule and expectation-maximization to generate predicted race, $\boldsymbol{z}_u$, to proxy for whether the individual is actually black, $\boldsymbol{x}_u$. Note that, for our purposes, it is often convenient to use the notation $\boldsymbol{z} = (\boldsymbol{z}_u, \boldsymbol{x}_o)$ and refer to both $\boldsymbol{z}_u$ and $\boldsymbol{z}$ as "the predicted covariates" or "the predictions." The vector $\boldsymbol{z}$ is written as if it predicts the whole vector $\boldsymbol{x}$, even though we aren't predicting $\boldsymbol{x}_o$ because we already know its true value.

We call plugging in the predicted covariates for the sometimes-missing covariates the "naive estimator." To see the problems with this estimator, we rewrite Equation (1) using the same β coefficients but replacing the original covariates with the predictions (using β_u to represent the portion of β that corresponds to the covariates in $\boldsymbol{x}_u$):

$$y = \boldsymbol{z}^{\mathsf{T}}\beta + \tilde{\epsilon}, \quad \tilde{\epsilon} = (\boldsymbol{x}_u - \boldsymbol{z}_u)^{\mathsf{T}}\beta_u + \epsilon. \tag{2}$$

Fitting a linear regression of y on z will be consistent for β if the predictions are uncorrelated with the new residual, $\tilde{\epsilon}$. Otherwise, the regression will suffer from omitted variable bias. The residual in this plug-in regression, $\tilde{\epsilon}$, has two components: the prediction error, $(\boldsymbol{x}_u - \boldsymbol{z}_u)^{\mathsf{T}}\beta_u$, and the residual from the original regression of y on $\boldsymbol{x}$, ϵ. Both components must be uncorrelated with the predictions, z_u, and the observed covariates, x_o, to avoid omitted variable bias.

Equation (1) already implies that the always-observed covariates, $\boldsymbol{x}_o$, and the residual for the original regression, ϵ, are uncorrelated, $\mathbb{E}[\boldsymbol{x}\epsilon] = 0$; if they were correlated, the sum of squared residuals could be decreased by adjusting β. This, however, is not enough; we need to make three more assumptions for the naive estimator to be consistent for β.

Assumption 1 (Exclusion Restriction) The predictions, $\boldsymbol{z}_u$, and the residual from the original regression, ϵ, are uncorrelated ($\mathbb{E}[\boldsymbol{z}_u\epsilon] = \boldsymbol{0}$).

Assumption 2 (Prediction Errors Uncorrelated with Observed Covariates) The prediction errors, $\boldsymbol{x}_u - \boldsymbol{z}_u$, are uncorrelated with the observed covariates, $\boldsymbol{x}_o$ ($\mathbb{E}[\boldsymbol{x}_o(\boldsymbol{x}_u - \boldsymbol{z}_u)^{\mathsf{T}}] = \boldsymbol{0}$). This implies the prediction error must be mean zero, and hence the classifier must be unbiased.

Assumption 3 (Prediction Errors Uncorrelated with Predicted Covariates) The prediction errors, $\boldsymbol{x}_u - \boldsymbol{z}_u$, are uncorrelatedwith the predicted covariates, $\boldsymbol{z}_u$ ($\mathbb{E}[\boldsymbol{z}_u(\boldsymbol{x}_u - \boldsymbol{z}_u)^{\mathsf{T}}] = \boldsymbol{0}$).

Many readers will be more familiar with the notion of classical measurement error rather than the slightly more expansive Assumptions 1–3. In the framework of classical measurement error, Assumptions 1 and 2 are assumed to be true, but Assumption 3 is permitted to be false. Other scholars, focused on nonclassical measurement error, still take Assumption 1 for granted but allow Assumptions 2 and 3 to be false (Aigner,1973; Kane, Rouse, and Staiger, 1999).

If all three assumptions hold, the naive estimator is consistent for the true coefficient, β, from the original regression of y on $\boldsymbol{x}$.[6] The problem with the naive estimator is that these three assumptions are exceedingly restrictive. For example, suppose the sometimes-missing covariate, x_u, and the prediction of that covariate, z_u, are both binary variables. If we are regressing presidential vote choice, y, on self-reported race, then we might use $x_u = 1$ if the individual identifies as white and $x_u = 0$ otherwise, while $z_u \in \{0, 1\}$ is the prediction of x_u from a classifier that takes as inputs the individual's surname and county of residence. Even in this innocuous situation, Assumptions 2 and 3 do not hold and the naive estimator is biased (Aigner, 1973). In general, if both the covariate x_u and the prediction z_u are categorical, then Assumptions 2 and 3 are violated and the naive estimator is biased (Aigner, 1973; Kane, Rouse, and Staiger, 1999).

When any one of these three assumptions is violated, then the naive estimator is biased and inconsistent, as it would be in the omitted variable situation. Some applied researchers,

6 Proof: Since $\mathbb{E}[\boldsymbol{x}_o\epsilon] = 0$, Assumption 1 implies $\mathbb{E}[\boldsymbol{z}\epsilon] = \boldsymbol{0}$. Assumptions 2 and 3 give us that $\mathbb{E}[\boldsymbol{z}(\boldsymbol{x}_u - \boldsymbol{z}_u)^{\mathsf{T}}\beta] = \boldsymbol{0}$. Taken together, these imply $\mathbb{E}[\boldsymbol{z}\tilde{\epsilon}] = \boldsymbol{0}$.

recognizing that their regressions suffer from prediction error, argue that their estimator suffers from attenuation bias, that their estimates are conservative (with respect to a zero null hypothesis), and hence prediction error can be ignored when we only care about the sign of a coefficient (Grumbach and Sahn, 2020). However, severe enough attenuation bias prevents the finding of statistically significant effects. In an applied field that places a high priority on the discovery of significant effects, a method that could reduce or eliminate attenuation bias should be welcome. More importantly, the bias/inconsistency of the naive regression is only guaranteed to take the form of attenuation if Assumptions 1 and 2 are satisfied but Assumption 3 is violated (as with classical measurement error, see Cameron and Trivedi, 2005, §26.2.3). Otherwise, the researcher cannot know the direction or size of the bias/inconsistency. Unfortunately, any bias and inconsistency (attenuation or otherwise) also invalidates any confidence intervals as well.

Some applied researchers have recognized that using the predictions of ML algorithms as covariates requires correction, but they do not formally analyze the issue, and consequently adopt ad hoc corrections that do not resolve the underlying inconsistency. For instance, Stewart and Zhukov (2009) sample from the predictive distribution implied by their ML algorithm, apply the naive estimator to this draw, and repeat this procedure many times to generate the presumed sampling distribution of the regression estimator. This procedure does not fix the inconsistency for a number of reasons, but the following is the simplest: if Assumptions 1 and 2 are satisfied and only Assumption 3 is violated, then each draw suffers from attenuation bias, and the average of these attenuated draws must itself be attenuated. Since the attenuation bias does not converge toward 0 as the sample grows, the estimator is also inconsistent.

2.2 Further Challenges from ML

Statistics and econometrics have produced a number of methods for addressing measurement error, which we review in Online Appendix G. Two features of the data common in ML applications cause these solutions to perform poorly in practice.

First, the predictions are not exogenously given, but learned from a training set where the true values of the covariate are known. A classifier's predictions are typically more accurate in the data used to train the classifier than in other data due to overfitting. Therefore, any correction that depends on knowing the accuracy of the predictions must contend with the fact that estimates of accuracy drawn from the training set will probably be overly optimistic. Overfitting precludes regressing the outcome on the ML algorithm's predicted probability for each label, as in Theocharis *et al.*, 2016, not just because those predicted probabilities might violate the measurement error conditions, but also because the predicted probabilities tend to correlate more strongly with x_u in the labeled sample than they do in the unlabeled sample. This same concern also prevents straightforward application of existing two-stage least squares (2SLS) estimators that one might use to address measurement error. Although our approach uses 2SLS and is closely related to the idea of regressing the outcome on predicted probabilities, we are careful to account for the overfitting concern when constructing our estimator.

Second, the subset of the data where the true label is known is typically small in ML applications—virtually always less than half, sometimes only a fraction of one percent. Many seemingly attractive approaches that we describe in Online Appendix H, including multiple imputation (Rubin, 2004), full information maximum likelihood estimation, and a fully Bayesian model (Ibrahim *et al.*, 2005), attempt to identify patterns in the subset of the data where the true label is known and then extrapolate these patterns to the rest of the data. Due to possibly incorrect functional form and distributional assumptions as well as estimation error, the patterns discovered in the labeled data do not hold exactly in the unlabeled data. If the proportion of unlabeled data is small, these errors do not affect the final estimates too much. But if most of the data is unlabeled, then the analysis relies heavily on extrapolation from the labeled data,

and even tiny errors in the patterns identified in the labeled data propagate into massive errors in the estimates of the parameters. In fact, the simulations in Online Appendix H show that the performance of these estimators degrades as the amount of unlabeled data grows. Our proposed solution avoids making strong functional form or distributional assumptions about the true data generating process.

It is important to remember that the researcher usually has access to an estimator that is consistent, even when all three assumptions are violated. To fit the ML algorithm, the researcher needs a subset of observations where $\boldsymbol{x}_u$ is observed; we refer to this as the labeled sample. The "labeled-only estimator" estimates the regression of y on $\boldsymbol{x}$ within the labeled sample. If the labeled sample is a simple random sample of the data, this estimator is consistent for the same reason a standard OLS estimator is consistent.

However, an estimator that exploited the unlabeled data without sacrificing consistency could be more efficient than the labeled-only estimator. In the following section, we develop just such an estimator—one which, given only Assumption 1, is guaranteed to be more efficient than the labeled-only estimator. In cases where, for empirical or theoretical reasons, the analyst is uncomfortable with Assumption 1 (the exclusion restriction), we recommend fitting the regression in the hand-labeled sample, because that procedure is robust to violation of the exclusion restriction at the cost of efficiency.

3 A Data-Splitting, OLS + 2SLS Approach

Our proposed method exploits the natural structure of ML-assisted problems: to train the classifier, there is always a sample where $\boldsymbol{x}$ is observed, which we call the labeled sample.[7] This suggests two consistent estimators if the labeled data are simple random sample of all of the data (we address the case where it is not in Online Appendix D). First, in the labeled subset of the data, OLS can consistently estimate the coefficients β from Equation (1) using the observed values of $\boldsymbol{x}_u$ (i.e., the labeled-only estimator mentioned above). Second, if the exclusion restriction holds, $\mathbb{E}(\boldsymbol{z}_u\boldsymbol{\epsilon}) = \mathbf{0}$, the algorithm's predictions $\boldsymbol{z}$ are valid instruments for $\boldsymbol{x}$. A careful application of two-stage least squares (2SLS) that takes into account overfitting allows us to estimate the same β coefficients from Equation (1) with data from the unlabeled sample. We propose an estimator that optimally combines these two estimators via the GMM. The estimator can be used with any classifier, whether that is support vector machines, neural networks, ensembles, or tree-based models.

3.1 Sample-Splitting

As we illustrate in Figure 1, our proposed estimator requires splitting the whole dataset into three distinct parts: the primary sample, the validation sample, and the training sample.

Our focus here is on applications where we can afford to label or hand-code only a fraction of the data, and thus a random sample of all cases is labeled and the rest are unlabeled. After this hand-labeling of some observations, $\boldsymbol{x}_u$ is assumed to be missing completely at random (MCAR)—an unobjectionable assumption if the researcher decides which cases are labeled and which are not, as is typical in ML applications. Let $p_i \in \{0, 1\}$ indicate whether $i = 1, \ldots, n$ is unlabeled, where $p_i = 1$ indicates that $\boldsymbol{x}_u$ is unobserved for observation i.[8] The p stands for "primary" sample, with $n_p = \sum_{i=1}^{n} p_i$ the number of observations in the primary sample. In most ML applications, $n_p >> n - n_p$.

We further split the labeled data (but not the primary sample) into two samples with different purposes: the training sample used to fit the ML algorithm and the validation sample used to

7 In practice, labels produced by human coders may be subject to coding error. We consider the problem of coding error and its relation to out framework in the conclusion.

8 If MCAR is violated, a modified version of our estimator can be used if $\mathbb{E}[\boldsymbol{x} \mid \boldsymbol{z}, p] = \mathbb{E}[\boldsymbol{x} \mid \boldsymbol{z}]$ and $\mathbb{E}[\boldsymbol{x}_u \mid \boldsymbol{z}]$ is a linear function of $\boldsymbol{z}$ (see Online Appendix D).

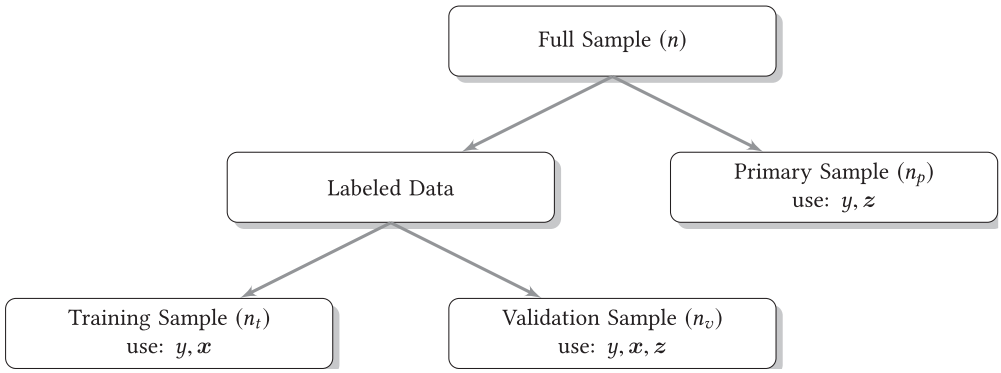

Figure 1. A graphical summary of the data-splitting strategy required for the proposed GMM estimator.

estimate the relationship between predictions and sometimes-missing covariates in the primary sample. We will elaborate upon the necessity of a validation sample when we describe the 2SLS estimator. The training sample is endowed with indicator $t_i \in \{0,1\}$ and count n_t, and the validation sample is endowed with indicator $v_i \in \{0,1\}$ and count n_v. The three samples are mutually exclusive and exhaustive such that $n = n_p + n_t + n_v$.

The researcher should divide the labeled data between the training and validation subsets completely at random. Labeling more observations and adding them to either the training or validation sample or both improves the performance of our estimator, but how much of the labeled data should go into each subset is a more difficult question. A larger training sample improves the accuracy of the classifier; a larger validation sample improves understanding of how these predictions relate to the missing data in the primary sample. The performance of our estimator depends on both. The optimal split depends on the type of ML algorithm being employed and the parameters of the data-generating process. For example, we employ a training-focused split in the application presented in Section 5 because the prediction problem in that case was particularly difficult. However, other applications may benefit from a more validation-focused split if the ML algorithm can achieve reasonable accuracy with a small training sample. In any case, our proposed estimator is always more efficient than the labeled-only estimator, regardless of how much we prioritize the training or validation samples (see Online Appendix A).

3.2 Component 1: Labeled-Only OLS

With our three samples, one option is to ignore the unlabeled primary sample and run OLS in the training and validation samples. This is the labeled-only estimator from Section 2. If linear regression would be a consistent estimator for β if $\boldsymbol{x}$ could be perfectly observed, then this labeled-only estimator is also consistent for β, because the labeled sample is assumed to be a simple random sample of the full data. However, the labeled-only OLS estimator is inefficient because it does not use any of the information contained in the usually much larger primary sample.

3.3 Component 2: Two-Stage Least Squares

Alternatively, we can draw inspiration from instrumental variables to incorporate the primary sample into our analysis. Social scientists often use instrumental variables and the accompanying 2SLS estimator to estimate causal relationships when some regressors are correlated with the error term, a pathology more popularly known as endogeneity.

We can apply these same ideas to address the problem of prediction error. First, as alluded to above, we use the training sample to train a classifier to predict the sometimes-missing covariates, $\boldsymbol{x}_u$. These predictions, $\boldsymbol{z}_u$, can then be used as instruments for $\boldsymbol{x}_u$. So long as the predictions are correlated with the outcome only through their correlation with the missing covariate (the

exclusion restriction from Section 2, which we dubbed Assumption 1), 2SLS consistently estimates β from Equation (1). In Section 3.5 we argue that the exclusion restriction is more likely to hold in these predicted covariate settings than in traditional instrumental variable analyses with convenient instruments and provide a hypothesis test for it.

To review the mechanics of 2SLS for our particular problem, define $\boldsymbol{\Gamma}$ as a matrix that encodes the linear projection of $\boldsymbol{x}$ on $\boldsymbol{z}$.

$$\boldsymbol{x} = \boldsymbol{\Gamma}\boldsymbol{z} + \boldsymbol{\eta}, \quad \mathbb{E}[\boldsymbol{z}\boldsymbol{\eta}^{\mathsf{T}}] = \mathbf{0}. \tag{3}$$

Now, let us repeat the same exercise from Section 2, except this time instead of using the predictions $\boldsymbol{z}$ directly, we substitute $\boldsymbol{\Gamma}\boldsymbol{z}$ in for $\boldsymbol{x}$.

$$y = (\boldsymbol{\Gamma}\boldsymbol{z})^{\mathsf{T}}\beta + \tilde{\epsilon}, \quad \tilde{\epsilon} = (\boldsymbol{x} - \boldsymbol{\Gamma}\boldsymbol{z})^{\mathsf{T}}\beta + \epsilon.$$

We can revisit the three assumptions for consistency from Section 2 with this new estimator. As before, we still require the exclusion restriction: $\mathbb{E}[\boldsymbol{z}\epsilon] = \mathbf{0}$. This reliance on the exclusion restriction is unsurprising, since instrumental variables in the usual endogenous regressor case also requires the exclusion restriction.

But now there is no longer an omitted variable problem since $\mathbb{E}[\boldsymbol{\Gamma}\boldsymbol{z}(\boldsymbol{x} - \boldsymbol{\Gamma}\boldsymbol{z})^{\mathsf{T}}] = \mathbf{0}$. Why? Because $\boldsymbol{x} - \boldsymbol{\Gamma}\boldsymbol{z} = \boldsymbol{\eta}$, and $\boldsymbol{\eta}$ is uncorrelated with $\boldsymbol{z}$ by definition of $\boldsymbol{\Gamma}$; see Equation (3). Verbally, when we regress $\boldsymbol{x}$ on $\boldsymbol{z}$, the resulting coefficient matrix $\boldsymbol{\Gamma}$ is the coefficient matrix that makes the residuals, $\boldsymbol{x} - \boldsymbol{\Gamma}\boldsymbol{z}$ uncorrelated with the regressors, $\boldsymbol{z} = (\boldsymbol{x}_o, \boldsymbol{z}_u)$. Thus, so long as the exclusion restriction is satisfied, the regression of $\boldsymbol{x}$ on $\boldsymbol{\Gamma}\boldsymbol{z}$ is consistent for β, reducing our three assumptions to one.[9]

2SLS exploits this observation by regressing $\boldsymbol{x}$ on $\boldsymbol{z}$ in the validation sample as the first stage. The coefficients from this first stage are an estimate of Γ, $\hat{\Gamma}$. The second stage regresses y on $\hat{\Gamma}\boldsymbol{x}$ in the validation and primary samples, and the coefficients from this regression are a consistent estimator for β.[10]

At first glance, the fact that this second-stage regression consistently estimates β may be puzzling. $\boldsymbol{\Gamma}\boldsymbol{z}$ is a prediction of $\boldsymbol{x}$, but $\boldsymbol{z}$ itself was already a prediction of $\boldsymbol{x}$. Why does multiplying $\boldsymbol{z}$ by $\boldsymbol{\Gamma}$ free us from Assumptions 2 and 3 from Section 2?

It is helpful to think of the first stage linear regression as a form of post-processing for the predictions. Linear regression can be understood as an algorithm for finding a Γ such that residuals are uncorrelated with the regressors. These are precisely the conditions required by Assumptions 2 and 3 of the plug-in estimator. No matter what classifier is used, running its outputs through this first stage assures the resulting linear predictions satisfy Assumptions 2 and 3.

Finally, it is essential to use only the validation data, and not the training data, to estimate $\boldsymbol{\Gamma}$. The purpose of $\boldsymbol{\Gamma}$ is to estimate the linear projection of $\boldsymbol{x}$ on $\boldsymbol{z}$ in the primary sample. Due to the inevitability of at least some overfitting, the coefficient in that projection will generally be closer to 1 in the training sample than it is in the primary sample, but it is the same in the validation sample as in the primary sample.

3.4 Combining Estimators with GMM

Rather than choose between these two estimators, we combine them via the GMM for greater efficiency. GMM enumerates a vector of functions that are in expectation equal to 0 at the true

9 This is guaranteed so long as the z_u explains enough variation in x_u after partialing out x_o—the usual "relevance" condition for instruments in 2SLS. This is likely, because algorithmic predictions z_u are optimized for explaining variation in x_u and typically use supplemental information (e.g., text, pixels) not fully captured by x_o.

10 More precisely, they are consistent as $n_v \to \infty$. We discuss this in further detail in Online Supplementary Material, Appendix A.

value of the parameter to be estimated. For example, OLS within the training and validations samples combined can be expressed as the following set of moment conditions:

$$g_1(\boldsymbol{b}) = \frac{1}{n_t + n_v} \sum_{i=1}^{n} (t_i + v_i)\boldsymbol{x}_i(y_i - \boldsymbol{x}_i^{\mathsf{T}}\boldsymbol{b}),$$

By solving for the $\boldsymbol{b}$ that makes this vector equal to $\mathbf{0}$, we get the familiar closed-form solution for the OLS estimator for the relevant subsample.

Likewise, the second stage of the two-stage least squares estimator can be written as

$$g_2(\boldsymbol{b}) = \frac{1}{n_p + n_v} \sum_{i=1}^{n} (p_i + v_i)\boldsymbol{z}_i(y_i - (\hat{\boldsymbol{\Gamma}}\boldsymbol{z}_i)^{\mathsf{T}}\boldsymbol{b}) \tag{4}$$

where $\hat{\boldsymbol{\Gamma}}$ is an estimate of $\boldsymbol{\Gamma}$ obtained by OLS of $\boldsymbol{x}$ on $\boldsymbol{z}$ in the validation sample.

The advantage of GMM is that multiple estimators can be combined by concatenating their moment conditions. Let $g(\boldsymbol{b}) = (g_1(\boldsymbol{b}), g_2(\boldsymbol{b}))$. This is what is known as an overidentified GMM, because there are $2d_x$ moment conditions (since g_1 and g_2 are both vectors of length d_x) but only d_x parameters to be estimated. Since there are more equations than parameters, there will in general be no vector of parameters that make all of the moment conditions exactly equal to $\mathbf{0}$. Rather, we find the $\boldsymbol{b}$ that makes g as close to $\mathbf{0}$ as possible:

$$\hat{\beta}_{\text{GMM}} = \arg\min_{\boldsymbol{b}} g(\boldsymbol{b})^{\mathsf{T}} \boldsymbol{W} g(\boldsymbol{b}) \tag{5}$$

where $\boldsymbol{W}$ is some positive definite weighting matrix $\boldsymbol{W} \in \mathbb{R}^{2d_x \times 2d_x}$ which we fix ex ante. This matrix governs how much the GMM prioritizes the OLS versus 2SLS moment conditions in the almost-certain event that they cannot all be satisfied exactly.

Thus, our proposed GMM tries to find an estimate of β that fits both the OLS and 2SLS moment conditions well. In Online Appendix A, we show that this estimator strictly dominates using either OLS in the labeled sample or 2SLS on their own. In that same Online Appendix, we delve into the technical details of our GMM estimator: how to derive optimal value of $\boldsymbol{W}$ that minimizes the variance, how to derive the asymptotic variance of this estimator, and how to account for the fact that $\boldsymbol{\Gamma}$ is estimated rather than known ex-ante. We show that, keeping n_v fixed, there is always an improvement in asymptotic efficiency from increasing the amount of unlabeled data n_p. Additionally, because we never required a specific distribution for the residuals $\boldsymbol{\epsilon}$, our standard errors and confidence intervals are robust to heteroscedasticity.

3.5 The Exclusion Restriction

In Section 2, we discussed how Assumptions 2 and 3 (prediction errors uncorrelated with observed covariates and predicted covariates) are implausibly restrictive for the naive estimator. They are often necessarily false under common circumstances (Aigner, 1973). Our proposed GMM estimator uses a first-stage regression to automatically satisfy these assumptions in the 2SLS component, but Assumption 1, the exclusion restriction, cannot be bypassed. The exclusion restriction is violated when predictions $\boldsymbol{z}_u$ explain the outcome $\boldsymbol{y}$ after the original covariates $\boldsymbol{x}$ have been linearly controlled for, which creates an omitted variable problem for 2SLS.

While it is not a theorem, the exclusion restriction is more plausible for ML predictions than for conventional instrumental variables analyses. Unlike traditional instruments, which are often taken as a matter of convenience and may be arbitrarily related to a variety of variables in the residual $\boldsymbol{\epsilon}$, ML predictions come from a mechanical process designed to approximate $\boldsymbol{x}_u$ as well as

possible. If the predictions $\boldsymbol{z}_u$ are explaining the residual ϵ, then they are doing so at the expense of explaining $\boldsymbol{x}_u$, which is, by the definition of β, uncorrelated with ϵ. Thus, accurate ML predictions are already designed to satisfy the exclusion restriction to some extent.

To make the exclusion restriction even more plausible, we recommend avoiding the use of variables known to correlate highly with ϵ, such as y, to produce the ML predictions $\boldsymbol{z}_u$. For example, in Online Appendix K we hide party identification from our classifier because it is highly correlated with the outcome of vote choice.

After taking appropriate precautions, we may still be concerned about violations of the exclusion restriction. In Online Appendix B, we derive a test of the exclusion restriction. When we treat the exclusion restriction as the null hypothesis, we can use an over-identification test to evaluate its suitability.

It is important to keep in mind that the test should not be treated as the final word on the exclusion restriction. A failure to reject the null hypothesis that the exclusion restriction is true (Assumption 1 is true) is the best outcome we can get, but a failure to reject the null hypothesis does not imply that the exclusion restriction is satisfied. A failure to reject could also occur because the test is not sufficiently powerful. Only the validation sample can be used for the test, so the only way to make the test more powerful is to increase the size of the validation sample. The validation sample will influence the power of the test at the usual root-n_v rate. Analysts should treat the exclusion restriction test as if it is testing for $\boldsymbol{z}_u$ as an omitted variable in the original regression of y on $\boldsymbol{x}$. Thus, the usual methods of power analysis (with n_v the relevant sample size) apply here. Simulations in Online Appendix J give more context on the ER test power at different validation sample sizes and different magnitudes of exclusion restriction violations. Note that increasing the size of the validation sample either requires more hand-coding (which is costly) or taking units from the training sample, which leads to less accurate predictions. As we show in Section 4, less accurate predictions tend to degrade performance. Therefore, the researcher faces an unavoidable tradeoff between maximizing the accuracy of their classifier and increasing the power of their test of the exclusion restriction.

The exclusion restriction test simulation in Online Appendix J show that small violations lead to small biases in the GMM estimator. However, these simulations emphasize that failure to reject the exclusion restriction must not be confused with evidence that the exclusion restriction is true. We include the exclusion restriction test as an automatic feature in our `R` package, but caution that passing the test we provide should be seen as a necessary condition for the GMM estimator to succeed but not as a sufficient condition.

4 Simulation Studies

In order to verify that the GMM preforms well for realistic configurations of the parameters, we provide simulations that compare the performance of four estimators: (1) the naive estimator that substitutes the predictions, z_u, for the missing covariates, x_u, in the primary sample, (2) the labeled-only estimator that performs OLS within the training and validation samples in which no covariates are ever missing, (3) the proposed GMM estimator from Section 3, and (4) an oracle estimator that regresses y on (x_o, x_u) for all observations, including the primary sample where x_u is actually missing. If GMM (or any proposed correction) is worse than the labeled-only estimator, then the analyst would be better off not using any unlabeled data at all. The oracle estimator is inaccessible in practice because the missing covariate, x_u, is not actually observed for all observations, but it sets an upper performance bound for any correction for prediction error.

Data for each iteration of each simulation is generated by the following process:

1. Let x_u be a vector of length $n_t + n_v + n_p$, where half of the observations are 1 and the other half are 0. x_o, the never-missing covariates, consists only of an intercept.

2. Generate $y \sim f(x)$, where f depends on the simulation setting.
3. Generate $z_u \sim \text{Bernoulli}(\pi x_u + (1-\pi)(1-x_u))$. π is the accuracy of the simulated classifier.
4. Split each draw into a labeled sample of size $n_t + n_v$ (where x_u is observed for all four estimators) and an unlabeled sample of size n_p (where x_u is hidden from all but the oracle estimator).

We study how the four estimators perform as three dimensions of problem difficulty vary: the size of the primary sample, the accuracy of the classifier (π), and the signal-to-noise ratio.

The returns for using unlabeled data depend on the amount of unlabeled and labeled data available. Below, we specify $n_t = 1{,}000$ training observations and $n_v = 1{,}000$ validation observations, and vary the number of unlabeled observations, n_p between 10,000 and 1,000,000. These ranges are drawn from prior applications. Grimmer, Messing, and Westwood (2012) hand-labeled 500 press releases for their classifier while Theocharis *et al.* (2016) hand-labeled 7,000 tweets. 2,000 labeled documents represents a substantial but feasible effort. As for n_p, Stewart and Zhukov (2009) consider a data set of 7,800 unlabeled documents and social media applications (such ours in Section 5) regularly run into the millions.

We parameterize the performance of the classifier by its accuracy, $\pi = P(z_u = 1 \mid x_u = 1) = P(z_u = 0 \mid x_u = 0)$. We consider two different classifier performance levels. In the first, the classifier is correct 72% of the time; this is roughly the accuracy achieved by Iyyer *et al.* (2014) for labeling the ideology of sentences and is typical of the performance a diligent social scientist could hope to achieve on a novel but well-defined problem. The second setup considers a best-case scenario where the classifier is able to achieve 90% accuracy. This is roughly the accuracy achieved by Socher *et al.* (2013) applying a then state-of-the-art classifier to the well-studied sentiment analysis problem with a massive training set.

Finally, the difficulty of the estimation problem depends on the effect size compared to the variance in the outcome, the signal-to-noise ratio. When the signal-to-noise ratio is high, the effect can be estimated with relatively few data points. In the high signal-to-noise setup, $y_i = x_{u,i} + \mathcal{N}(0,1)$. To demonstrate a low signal-to-noise setup and ensure that our results do not rely on Gaussian errors, we also consider a skewed, non-normal mixture distribution where $y_i = x_{u,i} + \mathcal{N}(0,8) + \text{Bernoulli}(0.15) \times |\mathcal{N}(0,20)|$, which generates a non-normal distribution with a standard deviation approximately 10 times as large as the high signal-to-noise ratio. This error distribution is designed to represent the many social science applications where the covariate of interest accounts for only a small proportion of the variation in the outcome.

Tables 1 and 2 present the bias, root-mean-squared-error (RMSE), and 95% confidence interval coverage with the realistic and best-case classifiers. Online Appendix I

Table 1. Realistic classifier.

		High signal-to-noise				Low signal-to-noise			
Metric	n_p	**NV**	**LAB**	**GMM**	**ORCL**	**NV**	**LAB**	**GMM**	**ORCL**
Bias	10^4	−0.47	−0.00	−0.00	0.00	−0.46	0.01	0.01	−0.00
	10^6	−0.56	−0.00	−0.00	−0.00	−0.56	0.01	−0.00	0.00
RMSE	10^4	0.47	0.04	0.04	0.02	0.50	0.46	0.36	0.19
	10^6	0.56	0.04	0.04	0.00	0.56	0.46	0.08	0.02
Coverage	10^4	0.00	0.96	0.96	0.96	0.35	0.96	0.95	0.96
of 95% CI	10^6	0.00	0.96	0.95	0.95	0.00	0.96	0.94	0.96

Table 2. Best-case classifier.

		High signal-to-noise				Low signal-to-noise			
Metric	n_p	**NV**	**LAB**	**GMM**	**ORCL**	**NV**	**LAB**	**GMM**	**ORCL**
Bias	10^4	−0.17	−0.00	−0.00	0.00	−0.17	0.01	−0.00	−0.00
	10^6	−0.20	−0.00	−0.00	−0.00	−0.20	0.01	0.00	0.00
RMSE	10^4	0.17	0.04	0.03	0.02	0.26	0.46	0.24	0.19
	10^6	0.20	0.04	0.02	0.00	0.20	0.46	0.04	0.02
Coverage	10^4	0.00	0.96	0.97	0.96	0.88	0.96	0.96	0.96
of 95% CI	10^6	0.00	0.96	0.95	0.95	0.00	0.96	0.96	0.96

The naive estimator performs poorly, even with a highly accurate classifier. Its bias is far larger than the other estimators, and the bias grows as the classifier becomes less accurate. As a result, its 95% confidence intervals do not achieve the proper coverage.

The GMM, by contrast, is essentially unbiased and its 95% confidence intervals achieve the correct coverage. It achieves a lower RMSE than the labeled-only estimator in all setups. Online Appendix I shows that its performance converges to that of the labeled-only estimator as the classifier becomes so inaccurate that it is completely uninformative, and the tables show that its performance approaches that of the oracle as the classifier becomes more informative.

The labeled-only estimator is preferable to the GMM if there is good reason to suspect the exclusion restriction is violated, such as when the test we provide in Online Appendix B rejects the null hypothesis that the exclusion restriction is satisfied. Even if the exclusion restriction cannot be rejected statistically, the labeled-only estimator's robustness to violation of the exclusion restriction make it attractive when the efficiency gains of the GMM are sm. Three parameters influence the gains of the GMM over the labeled-only estimator. First, the gains are large when the classifier is accurate, because a more accurate classifier allows more information to be extracted from the unlabeled observations in the primary sample. Second, the gains are large when there are more unlabeled observation in the primary sample, because more information can be extracted from the primary sample when it contains more observations. Finally, the gains are large when the signal-to-noise ratio is low, because linear regression estimates converge more slowly to the true parameter when the error term has a larger variance. This slower convergence increases the marginal return of additional data, which the GMM provides. If none of these conditions obtain, the researcher sacrifices little by using the labeled-only estimator. However, in these simulations, in which the exclusion restriction is satisfied, the proposed GMM estimator always does at least as well as the labeled-only estimator, and sometimes does substantially better.

A final point we consider is the payoff to increasing the number of labeled observations. Increasing the size of the labeled sample is a surefire way to decrease the variance and RMSE of the GMM estimator, but collecting more labeled data is often expensive (e.g., requires hand-coding), so there is a trade-off between getting a GMM estimator with the best performance and reducing the coding burden on the researcher. We can offer some guidance on optimizing this trade-off. The simulations in Online Appendix I vary the size of the labeled sample, and we can use this to infer the performance of the GMM estimator relative to a hypothetical labeled-only estimator that has access to more labeled data.[11] The simulations reveal a useful heuristic: when the primary sample is at least ten times the size of the labeled sample, the GMM estimator usually performs as well as or better than the labeled-only estimator that has access to twice as many

11 For more theoretical insight, the approximate relationship between the size of the labeled sample and the asymptotic variance of the GMM estimator can be found in Online Appendix A.

labeled observations (LABx2).[12] This appears to hold approximately across simulation settings, with the GMM performing much better than the LABx2 estimator under best-case accuracy and a low signal-to-noise ratio (GMM RMSE of 0.26 vs. LABx2 RMSE of 0.50) but only slightly worse than the LABx2 when the classifier is more realistic and the signal-to-noise ratio is high (GMM RMSE of 0.05 vs LABx2 RMSE of 0.04). These figures assume balanced classes, but the performance of GMM relative to LABx2 seems to improve somewhat with imbalanced classes.

This heuristic suggests a rough algorithm for determining the approximate amount of labeled data the analysis requires. Using theoretical knowledge (e.g., a power analysis) and/or parameter estimates obtained from labeled data that has already been collected, determine how many labeled samples would be required for estimation using labeled-only OLS. Call this number M. On average, the analyst will do just as well with the GMM estimator—provided there are at least ten times as many primary samples as labeled samples—if they only collect half as many, or $M/2$, labeled samples. If their predictions are particularly accurate or the signal-to-noise ratio is low, they can get away with even less, perhaps safely collecting as few as $M/4$ labeled samples.

In Online Appendix K, we provide a semi-synthetic application in whcih we study the relationship between homeownership and voting for Donald Trump in the 2016 American presidential election. Although we possess a complete data set, in that application, we artificially make the homeownership covariate missing for 90% of the observations. In that application, we find that the GMM's estimate is by far the closest to the oracle estimate, and the GMM yields a far more precise estimate than the labeled-only estimator. That extra precision allows the analyst to infer that the coefficient for homeownership is positive, while the labeled-only estimator cannot reject the null hypothesis that the coefficient is equal to 0.

5 Application: Hostile Political Dialogue

In this section, we offer an example application with actual missing data in which the consistency and efficiency of the GMM affects the substantive conclusion. This example studies how people respond to incivility in political discourse. Scholars of political communication have noted the rise of incivility in American political discourse and have set out to understand its causes, its consequences, and how it might be prevented (Mutz and Reeves, 2005; Theocharis *et al.*, 2016).

Munger (2017) shows that ordinary people respond to messages telling them that uncivil discourse is unacceptable by making fewer uncivil statements. Do third parties sanction uncivil messages in practice? Citizens might enjoy watching heated exchanges between their peers, as they do between elites (Mutz and Reeves, 2005), or they might be angered by violations of a norm of interpersonal communication. This question has important policy implications for how to reduce political incivility, and hence dampen affective polarization and improve political trust. If third parties rarely punish uncivil behavior, then reducing incivility may require interventions that incentivize doing so. If third parties frequently punish uncivil behavior and that behavior nevertheless persists at high levels, then an intervention that provides citizens with more effective messages for combating incivility may be more appropriate.

Until recently, it was difficult to observe political conversations among ordinary Americans in a natural setting. Fortunately, researchers can now access and study millions of online conversations on social media (Fong *et al.*, 2019). We draw our data from `reddit.com`, the world's largest online news aggregator, where users post threads and other users can comment on them. Users can also comment on other users' comments, and these exchanges sometimes turn into vitriolic debates. Users can upvote or downvote one another's comments, and each comment has a publicly visible score equal to one plus the number of upvotes minus the number of downvotes.

12 The simulations fix $n_p = 10^4$ and double the both the equal validation and training sample sizes from 500 to 10^3.

Our analysis investigates whether uncivil replies to comments on Reddit receive higher scores than civil replies.

Our data set consists of the 1,210,166 comments on the politics subreddit that are replies to other comments in 2014. Although Reddit's API provides both the text of the comments and their associated scores, we must ourselves determine whether each comment is uncivil. We classify a comment as uncivil if it insults another user ("Well hello, Captain Pedantic."), addresses another user in a condescending or insolent manner ("Do you understand that? I can't dumb it down any further."), accuses another user of being ignorant or ill-informed ("What part about simple math eludes you?"), or flatly and tersely contradicts another user ("lol wow you're delusional"). This criterion is similar to the "personal attack or harassment" criterion adopted by Munger (2017). We hand-labeled 3,026 randomly selected observations according to these criteria. To deal with the long tails of the score distribution, we impose a floor of −10 and ceiling of 10 on the scores.

Dividing the Data

Our hand-labeling yields 3,026 observations where the true incivility labels are known and 1,207,140 observations where the true incivility labels are unknown. We divide the labeled observations into $n_t = 2{,}413$ training observations to train the classifier and $n_v = 613$ validation observations to estimate $\boldsymbol{\Gamma}$. Given the difficulty of predicting the label (it occurs in less than a quarter of all observations), we allocate the majority of the labeled observations to training the classifier.

Predicting the Labels

We use the training sample to train a support vector machine as implemented in the *e1071* R package (Dimitriadou *et al.*, 2009). The features for this SVM are binary indicators for whether each word that appears in at least 5 training documents appear in the document. Because the uncivil label appears in only 21.0% of training observations, we weight each uncivil observation by $\frac{1}{0.210}$ and each civil observation by $\frac{1}{0.790}$ during training. This SVM achieves a precision for the incivility label of 0.36, a recall of 0.53, and an overall accuracy of 0.75. We use this fitted SVM to predict whether each document in the validation and primary samples is uncivil.

This example highlights the importance of keeping the data used to training the ML classifier (the training sample) separate from the data used to estimate the linear projection of the true label on the classifier outputs (the validation sample). In the validation sample, the estimated linear projection of *incivility* on the intercept and the SVM's prediction is $\hat{\boldsymbol{\Gamma}}_{\text{val},2} = (0.113, 0.242)$.[13] In the training sample, the estimated linear projection is $\hat{\boldsymbol{\Gamma}}_{\text{train},2} = (0.108, 0.382)$, deceptively suggesting a much stronger relationship between the SVM's prediction and incivility than will actually exist in the unlabeled sample. Even though the SVM is somewhat overfit (and classifiers are generally guilty of at least some overfitting), using a separate validation sample for the first-stage of the two-stage least squares ensures we accurately estimate the relationship between incivility and the SVM's prediction in the unlabeled primary sample.

The example also highlights the stringency of the assumptions required to support the naive estimator and the utility of our test for the exclusion restriction proposed in Section 3. Applying our test for the exclusion restriction yields a p-value of 0.302, so we fail to reject the null-hypothesis that the exclusion restriction holds. The exclusion restriction is at least plausible in this case, but because the null hypothesis is that the exclusion restriction is satisfied, failing to reject that null hypothesis is not the same as evidence that it is true. The naive estimator also requires Assumptions 2 and 3, which we can also test in the validation sample. Assumption 2 is satisfied in this case if the prediction error, $z_u - x_u$, is mean zero. Using the validation sample, We estimate

13 $\hat{\boldsymbol{\Gamma}}_1 = (1, 0)$, because it is the intercept.

a mean of −0.09 with standard error 0.02, which indicates that Assumption 2 is violated. Regressing the prediction error, $z_u - x_u$, on the predictions, z_u, in the validation sample to test Assumption 3, we estimate a coefficient of −0.65 with a standard error of 0.05. The violation of Assumption 3 is severe: the estimated correlation between the error $z_u - x_u$ and z_u is −0.67. We therefore expect the naive estimator to be biased and inconsistent.

The features of this application are similar to those in the simulations where GMM's performance gains relative to the naive and labeled-only estimators were largest: the classifier is not especially accurate, the number of unlabeled observations dwarfs the number of labeled observations, and the signal-to-noise ratio is low (as we will see, the estimated effects are roughly between −0.6 and −1.5, and the standard deviation of the outcome is 4.02). Simulations at the parameters from this Reddit data, detailed in Online Appendix L, confirm that the GMM outperforms the naive and labeled-only estimators in simulations at these parameter values.

Obtaining Estimates

We use the implementation of the GMM estimator in our R package. Even though this data set has over a million observations, the estimation conveniently runs in less than a minute using a single core on a personal computer. Our method and software should be accessible to researchers who can only dedicate a modest amount of computing resources to their regression estimates.

Comparison to Alternative Estimators

The substantive question is whether third parties punish uncivil posts. Targets of the uncivil behavior themselves may downvote the uncivil post. If so, then even in the absence of third-party punishment, the estimate of the coefficient for incivility on score could be as low as −1 (which would happen if the target of the incivility downvoted and everybody else was indifferent to the incivility). However, if the effect is below −1, then there must be third-party punishment on balance. Thus, to be certain we are finding third-party punishment, the null hypothesis must be that the coefficient is greater than or equal to −1. The coefficient on incivility merely being negative is not sufficient to conclude there is third-party punishment.

Figure 2 compares the results of the GMM estimator in this application to the other feasible estimators from Section 4. The naive estimator (using the predictions, z_u, for incivility, x_u) estimates a negative effect of incivility on post score, but the entire 95% confidence interval is greater than −1,

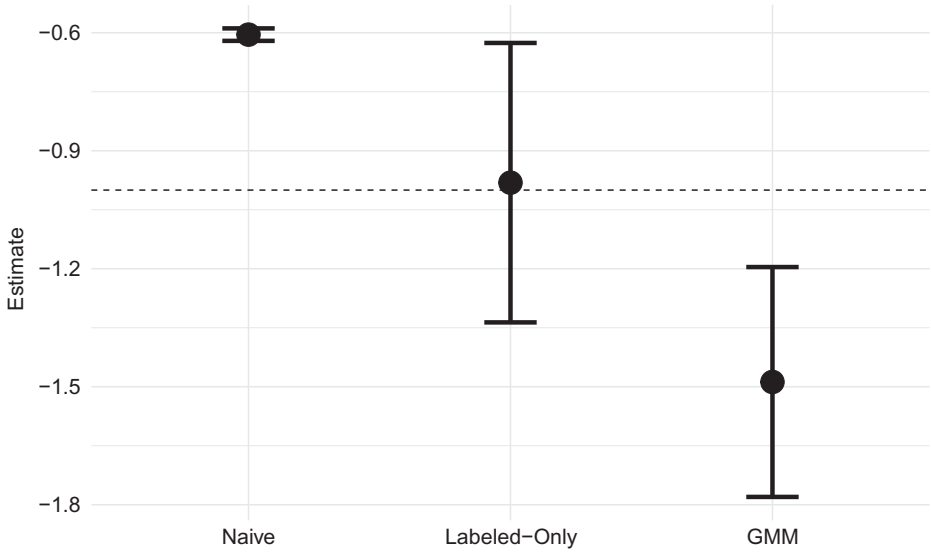

Figure 2. The effect of incivility on post score.

which does not allow us to reject the null hypothesis of no third-party punishment. But we have good reason to doubt the naive estimator, because we have shown that Assumptions 2 and 3 are both violated. The labeled-only estimator (ordinary least squares on the labeled data) produces a confidence interval which is consistent with the null hypothesis of a that incivility is only met with a downvote from the target themselves, but not any third parties.[14] In other words, the labeled-only estimate cannot reject the null hypothesis, although it is not a precise null either. Uniquely, the GMM's confidence interval is entirely below −1, rejecting the null hypothesis and finding clear evidence of third-party punishment.[15]

6 Conclusion

While using the outputs of ML algorithms as regression covariates offers a promising way of analyzing new data sources at a large scale, we show that researchers must account for error in the predicted covariates, unless they are willing to commit to demanding assumptions. We find that the bias and inconsistency posed by misclassification error can be large enough to alter the substantive conclusion of the analysis. Moreover, the solutions that many researchers employ to correct for misclassfication error do not actually address the problem. But we go further than identifying the problem and characterizing its severity; we also offer two solutions.

The simplest solution is to avoid ML by performing the analysis entirely in the subset of the data that has been labeled. This approach is best when the gains from using the GMM are small, such as if the classifier is inaccurate, the signal-to-noise ratio is high, or there are plenty of labeled observations. It is also attractive if the exclusion restriction is likely violated. Alternatively, the proposed GMM estimator offers greater efficiency if the exclusion restriction is satisfied. It performs well in simulations even under harsh conditions that are likely to arise in social scientific analysis: if classifier is inaccurate, the effect size is small relative to the variation in the outcome, or the ratio of unlabeled to labeled observations is large.

We have largely adopted the ML vocabulary common in text, image, and video applications, but this method can make it cheaper to collect data more generally. Surveys, voter files, administrative records, and other data sources are often missing important covariates, perhaps because the questions change from iteration to iteration of the same data set or because particular covariates are too expensive and time-consuming to measure. Our method offers a more efficient, reliable way of merging or completing data sets than obtaining the true values of all covariates for the full sample or restricting analysis to a complete subset of the data.

In the case of multiple surveys, if each survey is a simple random sample from the same population, researchers can use a ML algorithm to predict the missing covariates in the other surveys. Future work could extend the estimator to cases that are particularly likely to arise in these settings—for example, merging several surveys with partially overlapping covariates or with different but known sampling schema. These future extensions raise the exciting possibility of cheaply conducting statistical analyses with many observations and many covariates.

Our exposition focused on a missing binary covariate, but both our analysis and our software accommodates cases with multiple missing covariates and nonbinary covariates. Additionally, it can be extended to cases where the missing covariate is a function of variables that must be hand-coded. For example, Anastasopoulos *et al.* (2016) regress the percentage of a legislator's consistency that is black on the percentage of constituents in the legislator's photos who are black. The percentage of constituents in a given legislator's photos who are black can only be observed

14 The 95% confidence interval for a multiple imputation estimator drawn from Amelia with 20 imputed data sets is (−3.77, 1.83). This is unaccountably wider than the labeled-only estimator, undermining the motivation to use unlabeled data in the first place.

15 The robustness of this finding to controlling for the visibility of the comment is assessed in Online Appendix M, which shows that this score-incivility pattern persists across a variety of subsamples that restrict the data to posts that are more or less likely to be seen (and hence voted upon) by others.

by coding each of the legislator's photos by hand, so it is helpful to use ML to reduce the number of photos that must be hand-coded. Online Appendix E shows how this paper's GMM can be applied to these settings, subject to careful planning as to which photos are hand-coded.

If one of the labels is much rarer than the others, researchers may want to reduce the cost of training their classifier by using active learning or oversampling observations that are more likely to have the rare label. We caution against these popular practices for two reasons. First, by collecting a non-representative labeled sample, the researcher deprives themselves of the opportunity to use the labeled-only estimator, which we argue is the best estimator if the classifier is inaccurate, the signal-to-noise ratio is high, or the exclusion restriction is suspect. Second, training the classifier on a non-representative sample raises the possibility that prediction error will be systematically related to the characteristics on which the researcher oversampled. If those characteristics are correlated with the outcome, this induces a violation of the exclusion restriction. We explore these issues in greater detail in Online Appendix F, and describe a modification for the GMM to allow researchers to use oversampling or active learning in cases where they are confident it does not induce a violation of the exclusion restriction.

In many applications, ML is instead used to predict a missing outcome instead of a missing covariate. Applied researchers must resist the temptation to dismiss this prediction error as merely increasing the variance of the residuals. If the prediction error for the missing outcome is correlated with the covariates of the model, then the analogous naive estimator will be inconsistent, just like the missing covariate case discussed in this paper. Unfortunately, our preliminary efforts to solve this problem (not reported in detail here) indicate that the GMM framework developed in this paper is not as powerful in the missing outcome setting as it is in the missing covariate setting, since there is no natural analog of the exclusion restriction to exploit. Future research will need to be more creative in finding ways to extract information from ML predictions as regression outcomes.

Throughout, we have supposed all measurement error arises from the classifier's predictions. In practice, the human-coders themselves may be fallible. If there is only one human coder and no way to audit their decisions for mistakes, the GMM can be seen as a method for analyzing the data set as if the fallible human coder had coded every observation. Alternatively, if there are multiple human coders, the researcher can treat all of the coders agreeing that the label is 1, all coders agreeing the label is 0, and coder disagreement as three different values for the label. The classifier can then be trained to predict whether the coders would agree and, if so, what their decision would be. Since the GMM accomodates nonbinary covariates and multiple missing covariates, the GMM could then be applied to regress the outcome on the three-valued label. This would allow the observations that generate coder disagreement to have different coefficients, and this may be the most desirable strategy if we think there is unavoidable ambiguity in the correct label for some observations.

Alternatively, the goal is to estimate a regression as if all observations had been coded according to some gold standard that is in practice observed for only a subset of the observations. An example gold standard would be having all of the coders discuss the case and arrive at a consensus. To use our proposed GMM method for this purpose, the analyst would simply substitute the gold standard observations for the labeled observations in our setup, and the error-prone labels for the classifier's predictions. Future research could extend our framework to settings where there are some observations coded by the gold standard, some observations coded by fallible human coders, and some observations that have only a classifier-generated prediction. This extension would be especially valuable in cases where it is difficult but not impossible for humans to label the missing covariates, such as, arguably, our Reddit application.

Finally, our analysis also suggests a new desideratum for ML algorithms. Our simulations show that the returns from incorporating ML predictions is increasing in the accuracy of those

predictions, but satisfying the exclusion restriction required by our GMM is vital. If the goal of a classifier is to incorporate its outputs into a regression as a covariate, the classifier should seek to maximize its accuracy *subject to the constraint* that its predictions satisfy the exclusion restriction. Modifying existing classifiers to generate predictions that satisfy the exclusion restriction by construction, rather than focusing exclusively on accuracy, would be a valuable avenue for future methodological development.

Acknolwedgements

For helpful comments and suggestions, we thank Jason Anastasopoulos, Pablo Barbera, Justin Grimmer, Kosuke Imai, Walter Mebane, Brandon Stewart, and participants of the Society of Political Methodology's 2018 Poster Session and the International Methods Colloquium. All errors are entirely our own.

Data Availability Statement

Replication code for this article has been published in Code Ocean, a computational reproducibility platform that enables users to run the code and can be viewed interactively at Fong and Tyler (2020a) or https://doi.org/10.24433/CO.3552504.v1. A preservation copy of the same code and data can also be accessed via Dataverse at Fong and Tyler (2020b) or https://doi.org/10.7910/DVN/QQHBHY.

Supplementary Material

For supplementary material accompanying this paper, please visit https://doi.org/10.1017/pan.2020.38.

Bibliography

Aigner, D. J. (1973). "Regression with a Binary Independent Variable Subject to Errors of Observation." *Journal of Econometrics* 1(1):49–59.

Anastasopoulos, J., D. Badani, C. Lee, S. Ginosar, and J. Ryland Williams (2016). "Photographic Home Styles in Congress: A Computer Vision Approach." https://arxiv.org/pdf/1611.09942.pdf.

Cameron, A. C. and P. K. Trivedi (2005). *Microeconometrics: Methods and Applications*. Cambridge: Cambridge University Press.

Dimitriadou, E., K. Hornik, F. Leisch, D. Meyer, A. Weingessel, and M. F. Leisch (2009). "Package 'e1071'." R Software package, http://cran.rproject.org/web/packages/e1071/index.html.

Fong, C., N. Malhotra, and Y. Margalit (2019). "Political legacies: Understanding their singificance to contemporary political debates." *PS: Political Science & Politics* 52(3):451–456.

Fong, C. and M. Tyler (2020a). "Replication Data for: Machine Learning Predictions as Regression Covariates." Code Ocean, V1. https://doi.org/10.24433/CO.3552504.v1.

Fong, C. and M. Tyler (2020b). "Replication Data for: Machine Learning Predictions as Regression Covariates." https://doi.org/10.7910/DVN/QQHBHY, Harvard Dataverse, V1, UNF:6:vgF7Ffh39tB+eQJxHpax7A== [fileUNF].

Grimmer, J., S. Messing, and S. J. Westwood (2012). "How Words and Money Cultivate a Personal Vote: The Effect of Legislator Credit Claiming on Constituent Credit Allocation." *American Political Science Review* 106(4):1–17.

Grumbach, J. M. and A. Sahn (2020). "Race and representation in campaign finance." *American Political Science Review* 114(1):206–221.

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

Ibrahim, J. G., M.-H. Chen, S. R. Lipsitz, and A. H. Herring (2005). "Missing-data methods for generalized linear models: A comparative review." *Journal of the American Statistical Association* 100(469):332–346.

Imai, K. and K. Khanna (2016). "Improving Ecological Inference by Predicting Individual Ethnicity from Voter Registration Records." *Political Analysis* 24:263–272.

Iyyer, M., P. Enns, J. Boyd-Graber, and P. Resnik (2014). "Political ideology detection using recursive neural networks." In *Proceedings of the Association for Computational Linguistics*, pp. 1–11.

Jamal, A. A., R. O. Keohane, D. Romney, and D. Tingley (2015). "Anti-Americanism and Anti-Interventionism in Arabic Twitter Discourses." *Perspectives on Politics* 13(1):55–73.

Jerzak, C. T., G. King, and A. Strezhnev (2018). "An Improved Method of Automated Nonparametric Content Analysis for Social Science." https://gking.harvard.edu/files/gking/files/word.pdf.

Kane, T. J., C. E. Rouse, and D. Staiger (1999). *Estimating Returns to Schooling When Schooling Is Misreported.* National Bureau of Economic Research.
King, G., J. Pan, and M. E. Roberts (2013). "How Censorship in China Allows Government Criticism but Silences Collective Expression." *American Political Science Review* 107(2):326–343.
Munger, K. (2017). "Experimentally Reducing Partisan Incivility on Twitter." http://kmunger.github.io/pdfs/jmp.pdf.
Mutz, D. C. and B. Reeves (2005). "The new videomalaise: Effects of televised incivility on political trust." *American Political Science Review* 99(1):1–15.
Rubin, D. B. (2004). *Multiple Imputation for Nonresponse in Surveys*. Hoboken, NJ:John Wiley & Sons.
Socher, R., A. Perelygin, and J. Wu (2013). "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank." In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, pp. 1631–1642. Seattle, Washington.
Stewart, B. M. and Y. M. Zhukov (2009). "Use of Force and Civil–Military Relations in Russia: An Automated Content Analysis." *Small Wars & Insurgencies* 20(2):319–343.
Theocharis, Y., P. Barberá, Z. Fazekas, S. A. Popa, and O. Parnet (2016). "A Bad Workman Blames His Tweets: The Consequences of Citizens' Uncivil Twitter Use when Interacting with Party Candidates." *Journal of communication* 66(6):1007–1031.